A Broadcom Company

technologies

Service Operations Insight 4.2

Table of Contents

Release Notes **20**

CA SOI r4.2 Release Notes **20**

New Features 20

Compatibility Matrix 22

Connectors Compatibility Matrix 23

System Requirements 23

Operating System Support 23

Hardware Requirements 23

Software Requirements 25

Database Requirements 26

Software Support 26

Web Browser Support 27

Mobile Device Support 28

Database Enrichment Support 28

VMware Support 28

Special Character Support 29

Known Issues 30

Defect Fixed 31

International Support 33

Product Accessibility Features 33

Acknowledgments for CA SOI 4.2 35

CA Catalyst Acknowledgments 39

Release Comparison 41

Performance Results 42

Getting Started **44**

CA Service Operations Insight **44**

Understanding Your Role in CA SOI **44**

Where to Begin in CA SOI **45**

CA SOI Terminology and Concepts **45**

CA SOI Administration Process **52**

Access the CA SOI Dashboard and Operations Console **53**

Access the Mobile Dashboard, Reports, and USM Web View **54**

Installing **57**

Installation Planning **57**

Components 57

Installation Prerequisites 61

Installation Best Practices....63
Common Deployment Scenarios....64
Specialized Deployment Scenarios....70
Communication Ports....76
Installation....78
How to Perform a Full CA SOI Deployment....78
Obtain the Installation Worksheet....80
Install CA EEM....80
How to Perform a CA SOI Installation....80
Connector Installation....87
Install CABI (JasperReports Server)....92
Deploy SA Manager in Tiered Environment....94
CA SOI Reports with Unified Dashboards and Reporting for Infrastructure Management....95
Post-Installation Configuration and Customization....105
Help Desk Integrations....106
How to Configure a BMC Remedy Integration....106
BMC Remedy Integration Preparation....108
Install CA Process Automation for BMC Remedy....112
How to Configure BMC Remedy Integration Components....114
Using the BMC Remedy Integration....128
How to Customize BMC Remedy Ticket Fields with CA SOI Alert Parameters....129
How to Configure a CA Service Desk Integration....140
Review CA Service Desk Version Support....142
Configure CA Service Desk Connection....142
Export CA Service Desk SSL Certificate....142
Enable Automatic Links to Tickets....143
Set CA Service Desk Ticket Properties....144
How to Create Notification Triggers in CA Service Desk....145
Bypass CA SOI Polling for Ticket Assignees....155
Review CA Spectrum and CA Service Desk Ticket Synchronization....155
Review CA Service Desk Integration Troubleshooting....156
How to Configure a ServiceNow Integration....156
How to Configure Other Help Desk Product Integrations....157
How to Work with Configured Help Desk Integrations....162
High Availability Implementation....167
Implementation in a Microsoft Cluster Server Environment....167
How to Implement CA SOI in an MSCS Environment....167
How to Implement a Tiered CA SOI Deployment in a MSCS Environment....189
Install Connectors with HA Domain Manager....190
Update High Availability Environment....190

How to Update a High Availability Implementation with Multiple NICs ... 191
Upgrade Issue in Microsoft Cluster Server Environment ... 192
Configure CA SOI with EEM High Availability ... 192
Upgrades and Migrations ... 193
Upgrade CA SOI ... 193
Migration Requirements ... 196
How to Upgrade from a Previous High Availability Installation ... 197
SSL Implementation ... 197
Access the CA SOI Interfaces through an SSL Connection ... 197
Force SSL Connection for All Interface Access ... 198
Backing Up CA SOI Components ... 199
CA SOI Uninstallation ... 201
How to Perform a CA SOI Uninstallation ... 201
Uninstall Connectors ... 205
Connector Removal ... 205
Uninstall CA SOI HA Manager ... 207
Uninstall CA SOI HA Connector ... 207
Administrating ... 208
CA SOI Architecture ... 208
General Administration ... 214
SA Manager Details ... 214
View UI Server Connection Details ... 214
Configure Single Sign-On ... 215
Manage CA SOI Reports User Group in CABI JasperReports Server ... 217
Configure CA SOI Reports for CABI JasperReports Server ... 218
Configure Help Desk Integration ... 221
Configure Email and Failure Notifications ... 225
Configure Global Settings ... 227
Configure Auditing Levels ... 229
CA Process Automation Integration ... 230
Configure Google Maps Integration ... 231
Configure JNLP ... 232
Configure Metric Definition ... 232
Configure Mobile Dashboard Integration ... 233
Configure Synchronization ... 233
Configure USM Web View Integration ... 234
View Client Details ... 234
Access the SOI Console ... 236
Connector Administration ... 236
Connector Configuration Tasks ... 236

Managing CA Catalyst r3.x Connectors.... 242
Security Administration.... 244
Security Policy Statement.... 244
How to Configure Role-Based Security.... 245
Super User (samuser).... 247
Predefined User Groups and Access Privileges.... 247
Populate CA EEM with Users.... 248
Create User Groups.... 249
Manage User Groups.... 250
Manage User Group Access to Services.... 252
Manage User Group Access to Alert Queues.... 254
Manage User Group Access to Customers.... 255
Enable Guest User Account.... 258
Support for Common Access Card and Smartcard Authentication Using Client Certificates.... 259
Enable FIPS Mode in CA EEM.... 260
Generate Certificates and Keystore.... 261
Configure UI Server to Enable CAC Authentication.... 264
Configure CA SOI for New Smart Card.... 266
Enable SSL for SA Manager.... 269
Enable SSL for Enterprise Domain Connector.... 271
Configure Windows Client with CA SOI Server Certificates.... 271
Limitations.... 272
CAC Troubleshooting.... 272
Service Modeling.... 274
Service Modeling Introduction.... 274
Service Models.... 274
Service Model Concepts.... 274
Service Model Types.... 281
Federated Modeling.... 283
Planning Service Models.... 284
Navigate the Topology View.... 286
How to Build Service Models.... 290
Create and Configure the Service Model.... 292
Create Groups.... 294
Set the Granularity Level.... 295
Create Propagation Policy and Assign Types.... 296
Create and Assign a Service-Level Agreement.... 301
Create and Assign an Escalation Policy.... 301
Validate and Save the Service.... 302
Service Modeling Examples and Scenarios.... 304

How to Customize Service Model Display ... 321
Editing and Managing Services ... 324
Importing Services ... 327
Service Discovery ... 329
How to Create Dynamic Service Policies ... 330
How to Create Automatic Relationship Policies ... 334
How to Create Unmanaged Relationship Policies ... 338
Topology Warnings ... 343
How to Manage Service Discovery Policies ... 344
Service Discovery Connector Configuration ... 348
How to Use Command Line Service Discovery Operations ... 349
How to Create Generic Service Relationship Policy ... 351
How to Create and Work with Service-Level Agreements ... 355
How to Create and Manage Customers ... 362
Identify Customers, Sub-Customers, and Priorities ... 364
Identify Services ... 364
Create Customers ... 365
Create Sub-Customers ... 366
Configure Customer Priorities and Labels ... 367
Create Escalation Policies and Alert Queues ... 367
View Customer and Sub-Customer Details ... 367
View Customer Metrics ... 368
View Customers Associated with a Service ... 369
Example: Creating and Working with Customers ... 369
WSSSAServiceCmdV2 Command Usage ... 379
Performance Results for WSSSAServiceCmdV2 ... 381
Alert Management ... 381
Introduction to Alert Management ... 382
Alert Lifecycle ... 384
Alert and Event Visualization ... 385
Alert Management Administration ... 386
Configure Alert Escalation Integrations ... 386
Configure Alert Management Global Settings ... 386
Set Root Cause Analysis Mode ... 388
Exempt Alerts from Impact Analysis ... 392
Hide Alerts in Maintenance Mode ... 395
Rename Custom User Attributes ... 399
Set CI User Attributes ... 400
Enable Pending Clear State for Alerts ... 401
How to Create Escalation Policy ... 404

Escalation Policy Considerations....406
Define Escalation Policy....407
Create Escalation Actions....411
Configure Policy Assignments by Alert Queue....425
Configure Policy Assignment by Service....425
How to Create and Manage Alert Queues....426
Regular Expressions....431
Event Management....432
Introduction to Event Management....432
Event Management Architecture....433
Event Management Examples....439
Event Searches....440
Event Properties and Event Information....440
Normalized and Raw Event Types....443
Event Data Sources....443
Run an Event Search....444
Run a Raw Event Search....447
Event Search Syntax Guidelines and Best Practices....448
Configure Event Search Settings....452
Event Search Examples....452
Event Search Examples: Time-Based Correlation....452
Event Search Examples: Occurrence Frequency....453
Event Search Examples: Advanced Search Techniques....453
Event Search Examples: Raw Events....454
Event Search Examples: Moving from Simple to Complex....454
Working with Event Policies and Actions....455
Event Policy with Actions....455
Event Policy Deployment....456
Create an Event Policy with a Filter Action....457
Create an Event Policy with a Create Event Action....462
Create an Event Policy with an Enrichment Action....466
Create an Event Policy with a Normalization Action....481
Event Action Functions....488
Managing Event Policies....490
How to Manually Refine Event Policy....493
How to Add External Extensions to CI Types....501
Event Management Example Scenarios....503
Event Management Example 1: Filter Duplicate Events from Integrated Domain Managers....503
Event Management Example 2: Enrich Events with Related CI Information from the Persistence Store....504
Event Management Example 3: Create a New Event to Indicate a Crashing Service....506

Event Management Example 4: Combine a Create Event Action with an Enrichment Using Reevaluation....508
Event Management Example 5: Normalize Monitoring Traps....510
How to Schedule Maintenance for Services and Resources....513
Customization and Maintenance....519
Product Customization....519
PC Dashboard Customization....519
Mobile Dashboard Customization....525
Operations Console Customization....529
How to Customize the Operations Console Menu....533
Launch Web Server Scripts....545
Include TenantID in Correlation....546
Map CA Spectrum Global Collections to Alert Queues....546
Installation Maintenance....547
Password Maintenance....547
Communication Port Maintenance....554
Change CA EEM Connection Information....558
Update Relationship Significance....559
Update a CI Property....560
WSS Command Usage....561
CA Catalyst Operations....562
How to Perform CA Catalyst Reconciliation....563
Reconciliation Concepts....564
Configure the Reconciler....566
Enter or Modify Default Formula Input....567
Change Individual Property Formulas....568
Configure Existence Policy....568
Synchronization....570
How to Enable Alert Synchronization....570
How to Enable CA SOI Service Synchronization....574
How to Enable Maintenance Mode Synchronization....578
Database Maintenance....584
Rebuild the SA Store Database....584
Move the SA Store to a Remote Database Server....584
CA SOI Toolbox Utility....585
How to Clean Up Data with the CA SOI Toolbox....591
Task Scheduler for SOIToolbox Utility....596
Using....598
Operations Console Basics....598
Start the Operations Console....598
Operations Console Panes....598

How to Create and Manage Object Searches ... 602
How to View Object Audit Trails ... 606
View the Service Membership of a CI ... 610
Alert Management for Operators ... 610
View Alerts, Alert Details, and Extended Information ... 610
How to Assign and Update Alerts ... 619
How to Escalate Alerts ... 623
Work with Alert Tables ... 625
How to View Alert Queues ... 625
How to View Customers and Customer Details ... 628
CA SOI Dashboard ... 631
Access the Dashboard on a PC ... 631
Dashboard Metrics ... 631
View Service Status and Details ... 632
Run Reports from the Dashboard ... 638
View Services in Google Map ... 639
Display Service Detail Charts in Carousel Mode ... 641
CA SOI Mobile Dashboard ... 641
Access the Dashboard on a Mobile Device ... 641
Navigate the Mobile Dashboard ... 642
Perform Actions on Alerts on the Mobile Dashboard ... 647
Generating Reports in CABI JasperReports Server ... 649
USM Web View for PC ... 651
Access the USM Web View Starting Page on a PC ... 651
Perform a Search with USM Web View ... 651
Browse the USM Data with USM Web View ... 653
Work with USM Properties in USM Web View ... 654
Create and Manage Relationships with USM Web View ... 655
Manage CIs with USM Web View ... 656
Delete Manual Overrides with USM Web View ... 657
Favorite Views in USM Web View ... 658
Work with Results in USM Web View ... 658
USM Web View for Mobile Devices ... 659
Access the USM Web View Mobile Starting Page ... 659
Perform a USM Web View Mobile Search ... 659
Browse the USM Data with USM Web View Mobile ... 660
Results and USM Properties in USM Web View Mobile ... 661
Manage Relationships with USM Web View Mobile ... 663
Manage CIs with USM Web View Mobile ... 664
Delete Manual Overrides with USM Web View Mobile ... 666

Subscribe to RSS Feeds in USM Web View Mobile....666
Reference.... 668
CA SOI 4.2 Data Dictionary....668
dbo.ActionHistory....668
dbo.ActionPropertySets....669
dbo.AdminConfiguration....669
dbo.AlertActions....670
dbo.AlertAnnotation....671
dbo.AlertEscalationActions....672
dbo.AlertEscalationPolicy....672
dbo.AlertHistory....674
dbo.AlertImpact....675
dbo.AlertQueueAssignments....676
dbo.AlertQueues....677
dbo.AlertRelated....678
dbo.AlertRelationship....679
dbo.Alerts....679
dbo.AssociationType....682
dbo.AuditRecordActions....683
dbo.AuditRecords....684
dbo.AuditRecordTypes....684
dbo.ca_reportstrings....685
dbo.ca_ssa_alert....685
dbo.ca_ssa_application....687
dbo.ca_ssa_applicationserver....689
dbo.ca_ssa_applicationsystem....691
dbo.ca_ssa_asset....692
dbo.ca_ssa_backgroundprocess....694
dbo.ca_ssa_binaryrelationship....696
dbo.ca_ssa_bootsoftware....698
dbo.ca_ssa_businessprocessserver....700
dbo.ca_ssa_businesstransaction....702
dbo.ca_ssa_changeorder....703
dbo.ca_ssa_changepackage....705
dbo.ca_ssa_ci_detail....706
dbo.ca_ssa_ci_timestamp....712
dbo.ca_ssa_cluster....713
dbo.ca_ssa_comment....714
dbo.ca_ssa_communicationserver....715
dbo.ca_ssa_compliancestatus....717

dbo.ca_ssa_computersystem....718
dbo.ca_ssa_database....720
dbo.ca_ssa_databaseinstance....722
dbo.ca_ssa_directoryserver....724
dbo.ca_ssa_entity....726
dbo.ca_ssa_environmentsensor....727
dbo.ca_ssa_events....729
dbo.ca_ssa_file....729
dbo.ca_ssa_genericipdevice....731
dbo.ca_ssa_group....732
dbo.ca_ssa_hypervisormanager....734
dbo.ca_ssa_incident....736
dbo.ca_ssa_interfacecard....737
dbo.ca_ssa_itactivity....739
dbo.ca_ssa_itactivityprofile....741
dbo.ca_ssa_itactivitytemplate....743
dbo.ca_ssa_location....744
dbo.ca_ssa_mailserver....745
dbo.ca_ssa_managedaccess....747
dbo.ca_ssa_managementagent....749
dbo.ca_ssa_mediadrive....751
dbo.ca_ssa_memory....753
dbo.ca_ssa_multifunctionentity....755
dbo.ca_ssa_network....756
dbo.ca_ssa_networkserver....757
dbo.ca_ssa_notebooks....759
dbo.ca_ssa_notebooks_timestamp....760
dbo.ca_ssa_operatingsystem....760
dbo.ca_ssa_organizationalentity....762
dbo.ca_ssa_person....764
dbo.ca_ssa_port....765
dbo.ca_ssa_powersupply....768
dbo.ca_ssa_printer....769
dbo.ca_ssa_printserver....772
dbo.ca_ssa_problem....774
dbo.ca_ssa_processor....775
dbo.ca_ssa_project....777
dbo.ca_ssa_provisionedsoftware....779
dbo.ca_ssa_request....781
dbo.ca_ssa_resourceserver....783

dbo.ca_ssa_router....785
dbo.ca_ssa_runninghardware....787
dbo.ca_ssa_runningsoftware....788
dbo.ca_ssa_securityserver....790
dbo.ca_ssa_service....792
dbo.ca_ssa_servicespecification....794
dbo.ca_ssa_snmpv1access....795
dbo.ca_ssa_snmpv3access....797
dbo.ca_ssa_softwarecomponent....798
dbo.ca_ssa_storagearray....800
dbo.ca_ssa_switch....802
dbo.ca_ssa_tablespace....803
dbo.ca_ssa_tags....805
dbo.ca_ssa_transactioncontext....806
dbo.ca_ssa_transactionsegment....807
dbo.ca_ssa_transactionserver....808
dbo.ca_ssa_virtualizationmanager....810
dbo.ca_ssa_virtualsystem....812
dbo.ca_ssa_vmdatastore....814
dbo.CI....815
dbo.CIChangeHistory....818
dbo.CIEscalationPolicy....820
dbo.CIEvent....820
dbo.CIHealth....821
dbo.CIQuality....822
dbo.CIRelationship....823
dbo.CIRisk....824
dbo.CISLO....825
dbo.CIStaging....826
dbo.CIStagingTemp....829
dbo.Class....830
dbo.ClassProperty....832
dbo.ClassRelationship....832
dbo.ConnectorConfiguration....833
dbo.ConnectorPopupLauncher....834
dbo.Customer....835
dbo.CustomerImpact....836
dbo.CustomerRelationship....836
dbo.DbAvailHistory....837
dbo.DbQualityHistory....838

dbo.DbRiskHistory....839
dbo.DbSchemaVersion....840
dbo.DbUserPreference....840
dbo.EscalationPolicyRelationship....840
dbo.EscalationScheduleRelationship....841
dbo.Family....841
dbo.GEIntegration....842
dbo.GlobalDefaults....842
dbo.HelpDeskConfiguration....843
dbo.InactiveAlerts....843
dbo.KeyDefinition....843
dbo.OutageAlerts....844
dbo.OutageQualityAlerts....845
dbo.OutageRiskAlerts....845
dbo.PolicyGroup....846
dbo.PolicyGroupCI....847
dbo.PolicyType....847
dbo.ProductInfo....849
dbo.RootCauseRules....849
dbo.ScheduleRelationship....850
dbo.service_discovery_rules....851
dbo.ServiceLevelObjective....851
dbo.SLARecord....852
dbo.SLOSchedule....853
dbo.TopoLayout....854
dbo.UM_CLAIM....855
dbo.UM_CLAIM_BEHAVIOR....856
dbo.UM_CUSTOM_USERSTORE....857
dbo.UM_DIALECT....858
dbo.UM_HYBRID_REMEMBER_ME....859
dbo.UM_HYBRID_ROLE....859
dbo.UM_HYBRID_USER_ROLE....861
dbo.UM_PERMISSION....862
dbo.UM_PROFILE_CONFIG....863
dbo.UM_ROLE....864
dbo.UM_ROLE_PERMISSION....864
dbo.UM_TENANT....865
dbo.UM_USER....866
dbo.UM_USER_ATTRIBUTE....867
dbo.UM_USER_PERMISSION....868

dbo.UM_USER_ROLE ... 869
dbo.UserGroupSecurity ... 870
dbo.UserGroupSecurityAssignment ... 871
dbo.UserSecurityAssignment ... 872
dbo.UserSecurityGroups ... 873
dbo.UserSecurityTypes ... 873
Frequently Asked Questions - CA SOI ... 874
Troubleshooting ... 875
Diagnostic Tools and Methods ... 875
Debug Consoles ... 875
Manage the Debug Level for Specific Modules ... 879
Log Files ... 880
Using Log Files for Diagnosis ... 880
Logging in the CA SOI IFW ... 884
Connector-specific Logging ... 885
Configure Event Management Logging ... 885
View and Manage the Client Log (client.log) ... 886
View and Manage UI Server Connection Log ... 886
Control the Rolling Behavior of the Client Log File ... 887
Isolate CA Catalyst Logging Information from soimgr.log ... 887
Using Services for Diagnosis ... 888
Using the Status Bar for Diagnosis ... 890
Using the Administration Tab for Diagnosis ... 891
Set Notifications for the OutOfMemory Conditions on the SA Manager ... 891
CI Flow in CA SOI and Log File Outputs ... 892
Alert Flow in CA SOI and Log File Outputs ... 895
How to Track Alerts and CIs from CA Spectrum to CA SOI Using Debug Logs ... 901
How to Track Alerts from CA Spectrum to CA SOI Using Debug Logs ... 902
How to Track CIs from CA Spectrum to CA SOI Using Debug Logs ... 907
Trace a CI Using USM Web View for Diagnosing Synchronization Errors ... 913
Use the CA Catalyst Trace UI for Diagnosing Synchronization Errors ... 914
Product Troubleshooting ... 915
Important! Before Troubleshooting a Problem ... 916
CA Catalyst Troubleshooting ... 916
Synchronization - Priming Utility Runs for Hours ... 916
Installation or Initialization Errors ... 916
ActiveMQ Server Errors ... 917
Registry Errors ... 917
Correlation and Persistence Errors ... 918
Notification Manager Errors ... 918

Reconciliation Errors....919
Synchronization Errors....919
XML of CI Attributes Before and After EI transformation not Available....920
Exception Occurs when the Container Service is Stopped....921
Connectors Troubleshooting....921
Alert Synchronization Not Working for a Connector....921
Anti-virus Programs Affecting the Connector Performance....922
CA Spectrum Connector Binding in a Dual NIC Environment....922
CA Spectrum Connector Firewall Limitations....923
CA Spectrum Connector Keeps Reinitializing....923
Change Connector Credentials....923
Connector Data Not Imported....924
Connector Not Online....924
Connector Unable to Connect to the SSL-Enabled Domain Manager....925
How Do I Troubleshoot the Connector Policy?....925
How Do I Troubleshoot Connector Connection Problems?....926
How to Troubleshoot Connector Shutdown Behavior....926
Improved Connector Lifecycle Management....930
No Connector Data in the Operations Console....931
How to Convert Unmanaged CIs to Managed CIs....931
Sample Connector Changes Are Not Reflected in CA SOI....935
Unable to View CA Catalyst r3.2 Connectors in CA SOI r3.x....935
Dashboard Troubleshooting....936
Administration Tab or Dashboard Links Not Working with Firefox....936
Administration Tab Values Not Saving....936
Alerts Not Created....936
CIs Not Created....937
Failure Notification Emails Not Sending....937
Google Earth Does Not Display New or Updated Locations on the Map....937
Administration Tab Display Issues....937
Event Management Troubleshooting....938
Event Management Connection Problems....938
Event Search Returns no Results or Unexpected Results....938
Expected Alerts Not Appearing on Operations Console After Processing....938
Event Policies Not Producing Expected Actions....939
How Do I Control the Event Management Data Flow to the Operations Console?....939
Searches Taking Too Long....940
Error Messages on an Event Result Error Dialog....940
How to Search Archived Event Store Files?....942
Not Enough Event Groups in Search Results....942

Event Processing Performance Due to Mid-Tier Connector.... 943
Help Desk Integrations Troubleshooting....943
CA Service Desk Integration Troubleshooting....943
CA Service Desk Ticket Not Created by Escalation Policy Action....944
Cannot Create a BMC Remedy or HP Service Manager Ticket in CA SOI.... 944
HP Service Manager Integration Troubleshooting....944
Ticket Creation, Closure, or Update is Failing.... 945
Ticket Status Changed when Connector Shut Down or Removed.... 945
Integration Framework Troubleshooting.... 945
Java Heap Space Out Of Memory Error....945
Unable to Start the IFW Services....946
Mobile Dashboard Troubleshooting.... 946
Service Names do not Appear on the Mobile Dashboard.... 946
Operations Console Troubleshooting....947
Access - Proxy Server Prompt Opens When Accessing the Operations Console....947
Access - Operations Console Link Disabled.... 947
Access - Unable to Start the Operations Console.... 947
Alerts - How Do I Find an Alert?....948
CIs - Domain Manager Administrative CI States not Reflected in CA SOI.... 948
CIs - Duplicates Appear on the Operations Console.... 948
CIs - How Do I Find a CI?.... 948
CIs - Missing from CA SOI....948
CIs - Property Values are Incorrect....949
Escalation Actions - CA Process Automation Forms Not Available....949
Escalation Actions - Email Actions Not Sending....949
Escalation Actions - Tickets Not Created....950
Escalation Actions - Resolve Escalation Action Failures.... 950
Self Monitoring - Application CI Disappears.... 951
Service Discovery - Default Significance does not Match CI Significance....951
Service Models - Resolve Looping Problems.... 951
SLA Recurrence Not Triggering.... 952
Alerts Update are Delayed in the Alert Queues Tab....952
SA Manager and UI Server Troubleshooting.... 952
Disk Full.... 952
Error with Browser-Based UIs.... 953
Resolve an SA Store Database Connection Failure.... 953
Resolve a Third-Party Server Connection Failure....954
SA Manager Crashing on Startup or Dumping Memory and Performing Slowly....954
WrapperStartStopAppMain Message....954
Customize the Default Session Timeout for a User in CA SOI....955

CA SOI Manager takes Long Time to Restart....955
No Alerts in Operation Console....956
SOI Toolbox Troubleshooting....956
Toolbox Fails to Run....956
Toolkit Fails to Update Configuration Files....956
USM Web View Troubleshooting....957
USM Web View Does Not Display All CIs....957
USM Web View Search Returns Incorrect Results....957
Cannot log in USM Web View after Password Change....958
Service Discovery Troubleshooting....959
Reporting Troubleshooting....959
Connectors....962
Introduction to Connectors....962
Connectors Overview....962
Connector Infrastructure....963
Connector Integration Types....965
Connector Integration Example Scenarios....968
Basic Connector Information....969
Unified Service Model....970
How to Find USM Properties for a CI....976
How to Access the USM Schema Documentation....976
USM Parts....976
Advanced USM Features (Additional Information)....979
Connector Identification Numbers....982
Generic Connector Documentation....984
Domain Connector....984
Install the CA SOI Domain Connector....985
Import Services into the Enterprise SA Manager....987
Domain Connector Properties....988
Enable Southbound Synchronization....989
Convert Unmanged CI to Managed CI....990
Universal Connector....991
Universal Connector Components....991
Configure the Universal Connector....992
Universal Connector Programming Interface....993
Universal Connector Command Line Interface....995
Sample Universal Connector XML Files....1003
How to Map Old Schema Properties to USM Properties with Universal Connector....1008
Product Connector Documentation....1009
Connector Development....1009

Sample Connector....1009
How to Build a Custom Connector....1015
How to Set Up the Sample Connector in Eclipse IDE....1017
How You Implement a Custom Connector....1020
Connector Configuration Files....1033
Connector Operations....1037
How to Test a Custom Connector....1037
Custom Connector Logging....1040
Custom Connector Deployment....1042
Creating Connector Policy....1043
Policy Operations....1048
Connector Policy Examples....1064
Connector Policy Customization Best Practices....1077
CA Catalyst r3.4.1 Documentation....1078
CA Catalyst r3.4.1 Introduction....1078
CA Catalyst r3.4.1 Installation Planning....1078
How to Install CA Catalyst....1080
CA Catalyst Log Files....1084
Uninstall the CA Catalyst Container....1084
Upgrade the CA Catalyst Container....1085
How to Perform a CA Catalyst High Availability Implementation....1086
CA Catalyst r3.4.2 Documentation....1092
CA Catalyst r3.4.3 Documentation....1096
CA Catalyst r3.4.4 Documentation....1098
Web Services....1100
CA SOI REST Web Services....1100
REST Web Services Authentication....1103
REST Web Services Ordering Metric....1104
Available CA SOI REST Web Services....1107
Calling CA SOI REST Web Services from Perl Scripts....1116
Work with Maintenance Settings Using CA SOI REST Web Services....1120
View and Modify User Filters Using REST Web Services....1123
WS-MAN Web Services....1124
USM Entity Web Services....1129
USM Binary Relationship Web Services....1134
Notification Web Services....1143
Queue Web Services....1148
Customer Web Services....1153
Alert Web Services....1156
Propagation Policy Web Services....1161

Escalation Policy Web Services..... 1166
Escalation Action Web Services..... 1172
Additional Resources..... 1179
Green Book..... 1180
Documentation Legal Notice..... 1181

Release Notes

Domain management solutions monitor various aspects of a service, including support for IT infrastructure components or the end-user experience. None of these individual solutions give you a complete, end-to-end view of service health and availability across all management domains. Operations personnel often guess how the fault or performance issues reported across the network, systems, database, or application monitoring tools actually affect key IT services, degrade service quality, or increase the risk of an outage. Similarly, service stakeholders may not understand whether IT enables them to fulfill their business objectives.

CA Service Operations Insight (CA SOI) helps overcome these challenges by unifying the health and availability information from your domain management tools and aligning with your IT services. CA SOI conatins a service management layer to your management infrastructure and through an open and extensible integration platform (CA Catalyst), leverages and adds value to your investment in existing management technology. CA SOI provides integrations with several CA Technologies products and third-party applications, and the CA Catalyst integration platform lets you reconcile and synchronize data in CA SOI and across all domain managers. CA SOI uses several graphical interfaces to display the service operations data that supports the required business functions for all parties in the appropriate format. Operations staff uses these graphical interfaces to focus efforts correctly and business and IT objectives are properly aligned.

CA SOI also serves as a comprehensive level one operations console for managing the full stream of events and alerts from all integrated products. Operations staff can use CA SOI for a consolidated view of all alerts, enabling automatic escalation of important alerts that require quick action and problem resolution across domains from one interface. CA SOI provides alert queues for grouping logical categories of alerts. CA SOI also provides an event management layer that supports detailed event searches. CA SOI has several graphical interfaces for defining simple and complex event policies for event filtering, correlation, and enrichment.

CA SOI supports layered service and alert security through the use of user groups, customers, and alert queues. This security allows for a flexible user-specific view of the services and alerts company-wide.

CA SOI r4.2 Release Notes

The Release Notes for CA Service Operations Insight r4.2 contains information about new features, enhancements, product requirements, and any known issues.

Read this document before installing or upgrading CA SOI.

New Features

Support for AdoptOpenJDK Java

From the current release, we support only AdoptOpenJDK Java1.8.0.212. The Oracle JRE/ JDK is not supported. After you install the current patch, the jre, jre-32, jre-64 directories under the <SOI_HOME> is overwritten with the new JRE.

Important:

In case you delete these directories or uninstall the patch the Service Operations Insight application services stop.

Support for Night View in Google Maps

You can view services in Google Maps in the Night View mode. This helps the user to swap from a bright view and dark view. For more information, see View Services in Google Maps.

Decouple of MQ Server

You can install the MQ Server as a separate CA SOI component on a Standalone system and on High Availability. The disk storage of MQ Server is increased to 100 GB. For more information, see How to Perform CA SOI Installation and High Availability Implementation.

NOTE

Installation of MQ Server as a separate component is available only on CA SOI r4.2 version and on a new installation. During an upgrade, MQ Server is installed on SA Manager.

Database Support

This release supports Microsoft SQL Server 2014 (64-bit) Standard or Enterprise with the latest service packs. For more information, see Database Requirements.

Operating System Support

This release supports Microsoft Windows Server 2016 (64-bit) Standard with the latest service packs. For more information, see Operating System Support.

Cluster High Availability Support

This release supports High Availability implementation on Microsoft Cluster Server 2016. For more information, see Software Support.

Expanded CA EEM Support

This release supports the following updates for CA EEM:

- **Expanded CA EEM Version Support**
  Support for CA EEM versions r12.6 is certified in this release. CA SOI 4.2 is fully compatible with CA EEM r12.6.

  NOTE

  CA EEM r12.6 is supported only on the Windows Operating system.
- **CA EEM r12.6 in Cluster**
  Support for configuring CA EEM r12.6 in a cluster has been added and certified.

Support for CA Helpdesk Connector 1.0.0

The CA Helpdesk Connector integrates CA SOI and Service desk solutions (for example, HP Service Manager) that lets you resolve issues, incident, problem, change, and request. The following new features have been introduced in this release:

- Create an Incident in Service Desk solution for an alert in CA SOI
- Update a ticket in Service Desk solution for identical alarms in CA SOI
- CA SOI reflects Severity, Assignee changes and alert closure in Service Desk solution
- Service Desk solution reflects Severity, Assignee changes and alert closure in CA SOI

For more information, see CA Help Desk Connector documentation.

Support for Catalyst Container r3.4.2

The Catalyst Container supports the decouple of MQ Server. To connect to MQ Server, provide the MQ Server details in CA Catalyst installer. For more information, see CA Catalyst r3.4.3 Documentation.

NOTE

Upgrade of CA Catalyst r3.4.3 from previous CA Catalyst versions is not supported.

Enhancement of MQ Server **Disk Storage**

The disk storage of MQ Server is increased from 5 GB to 100 GB.

Enhancement of **WSSSAServiceCmd Command**

Improved the performance of the WSSSAServiceCmd command. The enhanced command is known as **WSSSAServiceCmdV2.** This command exports the services in a timely manner.

To export the service, use the following command:

```
WSSSAServiceCmdV2 -h<wsHostName:wsPort> -u <wsUsername> -p <encrypted wsPassword> -
a<Export> -s <Service Instance ID|*> -f <fileName>
```

To import the service, use the following command:

```
WSSSAServiceCmdV2 -h<wsHostName:wsPort> -u <wsUsername> -p <encrypted wsPassword> -
a<Import> -s <Service Instance ID|*> -f <fileName>
```

For more information, see Import Services Using WSSSAServiceCmd Command.

Enhancement of Relationship Visibility in Topology View

In Topology View Mode, you can now hide or show relationships for a specific relationship type (aggregates, bound, custom, and operative) in the main Contents pane. To select a specific relationship, click **Change relationship visibility** icon and select the required type.

For more information, see Navigate to Topology View.

Enhancement in Cleared Alert History tab

A **ClearedBy** field is added in the **Cleared Alert History tab** that allows you to view who cleared the alert. For more information, see How to Assign and Update Alerts.

Enhancement in Access Privileges to Alert Queues

You can view all the alerts in an Alert Queue, even for the CIs that are modeled in Services regardless of the access privileges. For more information, see How to Create and Manage Alerts Queues and How to View Alert Queues.

Compatibility Matrix

The supported operating systems and databases for this release of CA SOI are as follows:

Operation Systems

The certified edition of the Microsoft Windows Server platform is Standard, Enterprise. For the supported operating system, see Operating System Support.

Database

The databases are certified on 64-bit. For the supported database, see Database Requirements.

The following table provides the compatibility matrix for CA Service Operations Insight:

Version	Release Date	Integration Services	Catalyst Container	Connector
CA Service Operations Insight 4.0	October 2015	4.0.0.134	3.4.1.167	To know the compatibility matrix between the Connectors on Integration Services and Catalyst Container, click here.
CA Service Operations Insight 4.0 CU1	April 2016	4.0.0.134	3.4.1.167	
CA Service Operations Insight 4.0 SP2	November 2016	4.0.0.134	3.4.1.167	
CA Service Operations Insight 4.2	September 2017	4.2.0	3.4.3	

Connectors Compatibility Matrix

CA Service Operations Insight Release Information

The following table provides the release dates of each supported version of CA SOI:

CA Service Operations Insight	3.3	3.3 CU1	3.3 CU2	4.0	4.0 CU1	4.0 SP2	4.2
Release Date	December 2014	March 2015	June 2015	October 2015	April 2016	November 2016	September 2017

Compatibility Matrix

For more information about the integration of CA SOI with other CA products, and connectors, see CA SOI Connectors Compatibility Matrix.

System Requirements

Operating System Support

The following operating systems are supported for installing manager components and connectors:

- Microsoft Windows Server 2019 (64-bit) Standard with the latest service packs
- Microsoft Windows Server 2016 (64-bit) Standard with the latest service packs
- Microsoft Windows Server 2012 (64-bit) Standard or Datacenter with the latest service packs
- Microsoft Windows Server 2012 (64-bit) R2 Standard or Datacenter with the latest service packs

Some connectors must be installed on the system where the integrated domain manager exists.

For more information about operating system support for a specific connector, please see the product-specific *Connector Guide* that is provided with each connector package.

Hardware Requirements

The following table represents the recommended memory, disk space, and CPU requirements for running a CA SOI implementation of approximately the following size:

- Total Configuration Items (CIs): 200,000 - 400,000
- Managed CIs: 100,000
- Services: 5,000
- Open managed Alerts: 10,000
- Un-managed Alerts (not service impacting): 50,000
- Alert rate: Up to 30 alerts per second

These hardware requirements are not the minimums but instead are the recommended best practice settings for sustaining the optimal product performance for this size of implementation. Adjust these as necessary to fit your implementation.

The System column lists the individual system configurations that can exist, depending on how you choose to deploy the product components. See the descriptions below the table for additional information about each system configuration and its recommendations.

NOTE
Due to the inherent memory limitations on 32-bit operating systems, many of the best-practice memory recommendations require a 64-bit operating system. However, the product is certified to work within the memory limits of a 32-bit operating system with a reduced capacity.

System	Memory (GB)	Hard Disk Space (GB)	CPU
MQ Server	8	60	Two 3.0-GHz CPUs
SA Manager with SA Store	8	60	Two 3.0-GHz CPUs
SA Manager without SA Store	6	40	Two 3.0-GHz CPUs
UI Server	6	10	Two 3.0-GHz CPUs
Connectors	4	40	Two 3.0-GHz CPUs
CA Business Intelligence JasperReports Server	16	100	2.80-GHz CPU
Standalone deployment	12	60	Two 3.0-GHz CPUs

NOTE

The hardware requirements are same for both MQ Server and SA Manager.

All systems

For the optimal performance, run CA SOI on systems with at least two separate disk drives:

- Disk1: OS with swap space or virtual memory
- Disk2: All CA SOI components

SA Manager

The SA Manager system includes the CA Catalyst Logic Server and Registry, Mid-tier connector, Service Discovery, Event Management, and the Universal connector. This system can optionally include CA EEM and Microsoft SQL Server, hosting the SA Store database.

Consider the following items when planning your SA Manager installation:

- You can install CA EEM remotely
  - This can be a requirement where CA EEM is to support security requirements outside of CA SOI.

- You can install Microsoft SQL Server remotely.
  - This is recommended where Microsoft SQL Server is part of mid- to large-scale CA SOI deployments, it is a shared database server, or it is included in a SQL cluster to meet High Availability requirements.
  - A local installation of Microsoft SQL Server is preferred for a small-scale installation of CA SOI (for example, managing less than 60,000 CIs).
  - Follow the hardware and software requirements that Microsoft recommends for your SQL Server version.

- If Microsoft SQL Server is installed on a separate system, the SA Store database requires a minimum of 25-GB hard disk space.
  - The disk space requirements for the database grow over time due to the number of managed services, CIs, and alerts increasing over time. The database also contains tables that keep a record of historical data.
  - To control the database growth, perform periodic maintenance.

MQ Server

The MQ Server controls all messaging and communication from external sources. The server also receives alerts and CI information from connectors through the IFW and sends this information to various components for storage and analysis.

UI Server

The UI Server system includes the UI Server component.

Connector

A connector system has one or more connectors that are installed on the associated domain manager system or a remote system.

Consider the following items when planning connector installations:

- When you install more than one connector on a connector host, add approximately 200-300 MB of memory and hard disk space to this requirement for each connector.
- As a best practice, connectors that are installed on the same computer must not exceed 200,000 total CIs.

Standalone deployment

You can install all CA SOI components on the same system for demonstration and proof-of-concept (PoC) requirements.

Consider the following items when planning a standalone deployment:

- A standalone installation cannot include CA Business Intelligence, which must always be installed on a separate system.
- The required hard disk space for a standalone installation varies depending on:
  - The number of connectors that are installed on the standalone system
  - The size and location of the SA Store database
  - The size of your implementation
- Some connectors require installation on the same system as their domain manager.

Software Requirements

This section discussed the software requirements.

CA Embedded Entitlements Manager (EEM) Requirements

CA SOI requires one of the following versions of CA Embedded Entitlements Manager (EEM) to be present before installing CA SOI:

- CA EEM r12.6

  NOTE

  CA EEM r12.6 is supported only on the Windows Operating system.
- CA EEM r12.51 and above
- CA EEM r12.5 and above
- CA EEM r12 and above
- CA EEM r8.4 SP4CR15 (r8.4.415) and above

For convenience, CA EEM r12.51CR02 is packaged on the CA SOI installation media.

NOTE

CA SOI does not yet support CA EEM configured using Multiple Domains/Active Directory forests.

JRE Requirements

CA Technologies, a Broadcom Company, is moving towards adopting more open source technologies in its products. As a part of this strategy, various products have started using open-source implementations of Java. To align with this corporate direction, CA SOI has adopted AdoptOpenJDK (1.8.0.212), replacing Oracle JDK.

NOTE

- For information about setting the minimum JRE requirement for the Operations Console, see Configure JNLP.
- To avoid performance issues, if the connector server you monitor receives more than 100,000 CIs daily, enable the 64-bit IFW. For more information, see How to Install CA Catalyst Connectors (Pre-r3.2).

Database Requirements

CA SOI requires that you must have one of the following databases installed:

- Microsoft SQL Server 2016 (64-bit) Standard or Enterprise with the latest service packs
- Microsoft SQL Server 2014 (64-bit) Standard or Enterprise with the latest service packs
- Microsoft SQL Server 2012 (64-bit) Standard or Enterprise with the latest service packs
- Microsoft SQL Server 2008 R2 (64 bit), Standard, or Enterprise Editions with the latest service packs
- Microsoft SQL Server 2008 (64 bit) Standard or Enterprise with the latest service packs

NOTE

- CA SOI only supports clustering with Microsoft Cluster Services with SQL Server.
- Microsoft SQL Server **Always-On** Cluster is not supported. Do not select **Always-On** check box in SQL Server on Microsoft Cluster with CA SOI.
- Microsoft SQL Server **Always-On** Availability Groups for Active-Active failover is not supported.

The database must be case-insensitive.

Wherever you install the database, verify that you follow the hardware and software requirements that Microsoft recommends for your SQL Server version. Remember that the disk space requirements for the database grow over time due to the increases in the number of managed services, CIs, alerts, and tables that keep historical data. To control the database growth, perform periodic maintenance. For database sizing and maintenance, see CA SOI Toolbox Utility

Software Support

CA SOI features are supported with the following versions of CA Technologies and third-party products:

- Apache Tomcat 7.0.72
- Apache ActiveMQ 5.14.4.
- Review current information about supported connectors on the CA Service Operations Insight Connectors page on the CA Technologies Support site. From there, you can also see the product-specific *Connector Guide* that is provided each connector package. For more information about connectors, see Connectors.

Google Maps Integration

From 4.0 SP2 onwards, CA SOI supports Google Maps to view services. For information about configuring the Google Maps, see View Services in Google Map.

Cluster High Availability

CA SOI supports high availability implementations on:

- Microsoft Cluster Server 2016
- Microsoft Cluster Server 2012 Release 2
- Microsoft Cluster Server 2012
- Microsoft Cluster Server 2008 Release 2

Some of the SA Manager data files are moved to a shared disk when you install the SA Manager on a cluster node. The files do not require a large amount of space on the disk. The shared disk must meet the following requirements:

- The shared disk is defined in the cluster group that is selected for the CA SOI fail-over.
- The shared disk cannot be the QUORUM disk.
- The shared disk must be an MBR partitioned disk with a 32-bit ID value. GPT partitioned disks are not supported.

> **NOTE**
> For more information about implementing CA SOI in a Microsoft Cluster Server environment, see How to Implement CA SOI in an MSCS Environment.

Help Desk Integration

CA SOI requires the following software for Help Desk Integrations:

- CA Process Automation r3.1, r4.0, r4.1, 4.2.1, 4.3.0, or 4.3.1. CA Process Automation r4.x integrations are supported using either of the following configurations:
  - **CA Process Automation Player** - Provides the CA Process Automation product with predefined workflows.
  - **Full CA Process Automation product** - Provides full product functionality, including the ability to define custom workflows. A full product license is required to integrate with CA Process Automation for escalation action workflows.

> **NOTE**
>
> All help desk integrations except for CA Service Desk require CA Process Automation.

CA SOI supports the following help desk integrations:

- BMC Remedy AR System with ITIL or non-ITIL based HelpDesk
- CA Service Desk

> **NOTE**
> The Service Desk Web Services component must be installed for all versions for the CA Service Desk integration to work. If you have upgraded to r12.5 from a previous release, clear the CA Service Desk browser cache.

- HP Service Manager
- ServiceNow

Reporting Integration

CA SOI requires the following software for reporting integration:

- CA Business Intelligence (JasperReports Server) 6.3

SOAP

CA SOI supports SOAP 1.2 only.

Web Browser Support

Enable the following for all supported browsers:

- Java
- ActiveX
- Cookies
- Security settings:
  - Add the CA SOI URL to trusted sites.
  - Add the domain name of the CA SOI UI Server and SA Manager (for example, company.com) to the privacy-managed websites and set to Allow.

CA SOI supports the following web browsers:

- Microsoft Edge with Adobe Flash v11 and above.
- Microsoft Internet Explorer 8, 9, 10, and 11 with Adobe Flash v11.

  NOTE

  When using Internet Explorer 11, to avoid issues with the display of the pages on the Administration tab of the Dashboard, either disable IE enhanced security, or add the domain of your CA SOI Dashboard to the browser Compatibility View settings.

  If you are using Internet Explorer to view this Wiki, Explorer 8 (not in Compatibility Mode) or higher is required.

  Few Adobe Flash based features may not work due to data rendering issues by the IE engine.
- Mozilla Firefox 12.x and above with Adobe Flash v11 and above.

  NOTE

  Due to conflict with Firefox versions before and including version 12, we recommend upgrading to versions later than 12. The conflicting versions can result in the Dashboard Administration tab and other Dashboard links not working properly. For more information, see Administration Tab or Dashboard Links Not Working with Firefox.
- Google Chrome 23.x or higher.
- Apple Safari 5.x or higher.

Mobile Device Support

The Mobile Dashboard and USM Web View support the following mobile devices:

- Mobile devices running iOS 5 and later
- Mobile devices running Android 2.1 and later
- Mobile devices running BlackBerry OS 6 or later

  NOTE
  For more information about configuring the Mobile Dashboard, see Configure Mobile Dashboard Integration. For information about configuring USM Web View, see Configure USM Web View Integration.

Database Enrichment Support

CA SOI supports the database enrichments in Event Management policies using the following database types:

- Microsoft SQL Server 2008, 2012, 2014
- Oracle 10g R2, 11g R1, and 11g R2
- MySQL 5.0, 5.1, and 5.5

Enter connection information specific to your database type to enable the enrichment.

Although it may be functional, CA Technologies has only tested the listed database types to work with the database enrichments in this CA SOI release and cannot ensure compatibility with other databases.

For more information about database enrichment actions, see Create a Database Enrichment Action.

VMware Support

CA SOI supports installation in VMware environments as long as the virtualized environment meets the same underlying operating system and hardware allocation requirements specified for CA SOI and its supporting components.

For more information about CA Technologies support for running products in VMware environments, see the CA Support article.

Special Character Support

NOTE
ASCII alphanumeric refers to the following characters: A-Z, a-z, and 0-9.

This product does not support localized characters or double dollar sign characters ($$) in the following items:

- Computer name
- Microsoft SQL Server instance name
- Installation source media, response files, or destination paths
- User names
- Passwords

NOTE
The password limitation does not apply to Active Directory or LDAP.

This product supports the following set of characters with any frequency, pattern, or ordering of those characters for the fields that are specified here:

- **Computer name:** ASCII alphanumeric and hyphen (-)
  Underscores and other special characters are commonly used by Microsoft Windows operating environments but are not allowed in RFC-952 even with the looser restrictions of RFC-1123. Their use can cause problems in systems that communicate over the network, or even locally when an application uses network-related functions. This condition is true even if your network is using the Microsoft DNS Server or no DNS server at all.
  Some versions of Microsoft Windows block the use of non-standard characters into the computer name in their administrative UIs, or they present warnings about the ramifications of their use. Still, the Win32 SetComputerName API lets you get around some of those blocks.
  Although adding IP address or computer name aliases into the \etc\hosts file of each computer may correct some problems you may encounter while using non-standard characters or names not allowed by the DNS standard, the deployment of this product in such an environment is not supported.

  NOTE
  For more information, see Microsoft Article 909264 and the documentation for the SetComputerName Win32 API.
- **Microsoft SQL Server Instance Name:** ASCII alphanumeric
- **Installation source media and response file path:** ASCII alphanumeric, hyphen (-), underscore (_), period (.), open bracket ([), close bracket (]), space ( ), tilde (~), open paren ((), close paren ()), and dollar sign ($).
- **Installation destination path:** ASCII alphanumeric, hyphen (-), underscore (_), period (.), space ( ), tilde (~), open paren ((), and close paren ()).
- **System %TMP% and %TEMP% path:** ASCII alphanumeric, hyphen (-), underscore (_), period (.), open bracket ([), close bracket (]), tilde (~), space ( ), and dollar sign ($).
  Unsupported characters can exist in these variables if your logged on user name contains such characters or because you have otherwise manually defined them this way. In this case, change these system environment variables to contain characters meeting these restrictions.

  NOTE
  The installation source media and response file path, installation destination path, and system %TMP% and %TEMP% paths cannot start with a space ( ). The colon (:) character is supported only when it immediately

follows the name of an existing disk drive as part of a path specification such as C:\. The backslash (\) character is supported only as a path separator of directory levels.

- **User name:** ASCII alphanumeric, hyphen (-), and underscore (_).
- **Passwords:** ASCII alphanumeric, hyphen (-), and underscore (_).

Known Issues

This release contains the following known issues:

Wsssaservicecmdv2 Command

- The wsssaservicecmdv2 command does not give an error when you import an already existing service. The command overwrites the existing service.
- The wsssaservicecmdv2 command does not retain the relationship type when you import and export a service. For example, A relationship type in category **Aggregate** appears as **Aggregates**, category **Custom** appears as **DependsOn**, category **Bound** appears as **BoundTo**, and category **Operative** appears as **Requires**.
- The CIUserAttibute 6 through 10 values are missing for a CI in a service. This issue occurs when you import the CIs through WSSSAServiceCmdV2.

CA SOI Installer

- The CA SOI installer fails with the error *SQL Server not found* when the reverse and forward DNS lookup is not synchronized.

Sample Connector

- The Launch in Context displays **Page Not Found** error in a distributed environment for an alert.

MQ Server

- Ensure that you clear the MQ Server option on the component selection window before installing SA Manager and User Interface on the same system where MQ Server is installed.
- The MQ server installation fails on Windows Server 2008 environment.

IFW

- The IFW service status fails to update when you stop IFW service while the SA Manager service is not completely started.
- The configuration files that are located in **\SOI\jsw\conf** folder overwrite when you upgrade CA SOI to 4.2 version. As a workaround, take backup of the configuration files located in **\SOI\jsw\conf** .
- The Current Integration Framework Status appears **closing** when the CA Catalyst Container service is stopped.

Domain Connector

- All the CA SOI services are restarted when you install Domain Connector on CA SOI system.

High Availability

- The following message appears when you execute SSAHA script on CA SOI system with User Interface installed. You can ignore the message and execute SSAHA sript.

```
CA Service Operations Insight is not at the correct release level. You must be on
 release 4.2 or above
```

Connectors Issue

- The UIM hostname validation fails in the UIM Connector installer. This issue occurs after successful login to the host.
- CA Spectrum connector 2.0.0.244 does not support Windows 2016 Operating System.

Google Maps

- An error message does not appear for an invalid service location on Google Maps.

Event Policy

- The event policy fails to execute when you define the search pattern in a policy with four backslashes. During pattern parsing, the four backslashes become as two backslashes (escaped character).
  As a workaround, Add more two backslashes to the policy in a Policy Editor and save the policy.

Alert Management

- The conditions set for **Service Impact, Severity**, **Family** and **Class** tabs in the Alert Filter are not saved for user preferences when you select all the conditions in the tabs.

Catalyst Container

- The following error appears when you select **Remember Me** check box on the Catalyst Registry login page.

```
Login failed! Please recheck Server URL and try again
```

- The Current Container Status appears **closing** when the CA Catalyst Container service is stopped.

CA SOI Reports with Unified Dashboards and Reporting for Infrastructure Management

- Report export fails when auto refresh and exporting reports occur simultaneously.
  As a workaround, Export the reports again.
- The data in detailed dashboard may vary with the data of aggregated dashlets when the data is obtained from CA SOI between the query intervals changes with real-time data.
- The Filter Group window does not close when you click apply in the Filter Group window.
  As a workaround, manually close the Filter Group window.
- The Alert Queue KPI and Service Name sections fail to remain in the same context after auto refresh and the dashlet navigates to the first page.
  As a workaround, refresh the dashboards.
- The data fails to appear for a user in dashboards when you move a user from one user group to another user group in Operation Console.
  As a workaround, Delete the user and add the user to the user required group.
- The down arrow scrollbar fails to appear in the dashlets when you select an entry where there is more than one page of data on the dashboards.
- CA SOI Reports do not support the **SQLServer** named instance.

CABI Server Dashboard

- The filter option does not work for *any* of the columns (**Health**, **Quality,** and **Risk)** on **Library**, **Business Service View** page.

Defect Fixed

The following issues are fixed as a part of CA SOI 4.2:

Alert Management

- The existing Alert User Attribute values for the updated alert are removed when an alert is updated. This issue occurs when the escalation policy is not executed again but update alert is triggered from the connector.
- The data for user attribute was not updated in the Operation Console when an enrichment policy was created for an alert using the type JDBC.
- The updated severity of an alert fails to reflect on the Operation Console. This issue occurs when you change an alert from normal severity to minor, major, or critical severity in the nimsoftconnector_policy.xml file.
- The exceptions are not seen in the CA SOI Manager log file when an alert or CI validation fails.
- The TicketID fails to appear in the Component Details pane in the Operation Console after you restart the CA SOI Manager.
- Irrespective of the initial view preferences set for the Services tab, the Alerts tab appeared in the default view when the Services tab was selected.

Vulnerabilities

- Fixed the security vulnerability which is caused by Apache Commons Collections 3.1. Apache Commons Collections 3.1 is replaced with Apache Commons Collections 3.2 in the instances where the security issues occurred.
- Cross-site Scripting and Unencrypted Login Request vulnerability issues on CA SOI Application Server.

License

- Fixed the CA SOI Jasper Report Server license issue. The CA SOI Jasper Report Server license is renewed.

Connectors

- Few Alerts that are cleared in Spectrum are not cleared in CA SOI intermittently. However, it shows a clear alert that is processed in the CA SOI manager logs.
- The following error appears when an escalation action was imported for creating a service desk ticket using the wssamservicecmd.bat command.

  ```
  java.lang.NullPointerException
  ```

- During the connector startup, sometimes, the SA manager receives no data from the MDR. Therefore, all pre-existing CIs and alerts in SA Manager were deleted.
- The following error appears when Available statuses property in helpdesk configuration for Universal Helpdesk configuration was not saved.

  ```
  Error Saving the Configuration
  ```

Service Modeling

- In the Service Discovery log file, the duplicate entries were inserted which caused database deadlock.
- The value for the Configurable Item property does not appear as ModelName format in SDM ticket. This issue occurs when you provide the value ModelName as the Configurable Item property to SDM while creating a ticket.

USM Web View

- When SSL is enabled on CA SA Manager Server, the USM Web View property was not updated and an error occurred.

TenantID Correlation

- The configuration item obtains a value in which the tenantid appears as a prefix to **$[Model Name]**. This issue occurs when you enable correlation with tenantid and when you create an escalation action using create ticket or update ticket type.

SOI Toolbox Utility

- The --cleanHistoryData command fails when the CA SOI services are stopped and the following error appears:

```
SOI Instance has no attribute 'servicestatus'
```

Escalation Action and Escalation Policy

- The Escalation Policies trigger after the policy action delay is expired. This issue occurs when you define the time criteria in Alert Escalation Policy Editor, Time tab.
- The policy fails to run within the specified time that you define in Alert Escalation Policy Editor, Time tab. This issue occurs for the policies that are assigned to a particular Service in the Service Assignment tab.

Operation Console

- The Operation Console is not accessible whenever the SOI services are restarted and the following error appears in soimgr-debug log file.

```
Error!!! Loop detected!
```

Google Maps

- The color of CI does not update in Google Maps when the health of the CI is changed in the Operation Console.
- The service details are not displayed in Google Maps due to many services.

International Support

The CA SOI r4.2 software and documentation are not localized.

An *internationalized* product is an English product that runs correctly on local language versions of the required operating system and required third-party products. An internationalized product supports local language data for input and output. The internationalized products also support the ability to specify local language conventions for date, time, currency, and number formats.

CA SOI r4.2 currently supports the ability to specify local language conventions for date, time, currency, and number formats.

For updates regarding the availability of fully localized versions of CA SOI, see http://ca.com/support.

Product Accessibility Features

CA Technologies is committed to helping all customers, regardless of their ability, to use its products and supporting documentation to accomplish vital business tasks. This section outlines the accessibility features that are part of CA Service Operations Insight.

Product Enhancements

CA Service Operation Insight offers accessibility enhancements in the following areas:

- Display
- Sound
- Keyboard
- Mouse

Note: *The following information applies to Windows-based and Macintosh-based applications. Java applications processed on many host operating systems, some of which already have assistive technologies available to them. For these existing assistive technologies to provide access to programs written in JPL, they need a bridge between themselves in their native environments and the Java Accessibility support that is available from within the Java virtual machine (or Java VM). This bridge has one end in the Java VM and the other on the native operating environment, so it will be slightly different for each operating environment it* bridges to. *Sun is currently developing both the JPL and the Win32 sides of this bridge.*

Display

To increase visibility on your computer display, you can adjust the following options:

- **Font style, color, and size of items**
  Lets you select font color, size, and other visual combinations.
- **Screen resolution**
  Lets you change the pixel count to enlarge objects on the screen.
- **Cursor width and blink rate**
  Lets you make the cursor easier to find or minimize blinking.
- **Icon size**
  Lets you make icons larger for visibility or smaller for increased screen space.
- **High contrast schemes**
  Lets you select color combinations that are easier to see.

Sound

Use sound as a visual alternative or to make computer sounds easier to hear or distinguish by adjusting the following options:

- **Volume**
  Lets you turn the computer sound up or down.
- **Text-to-Speech**
  Lets you hear command options and text read aloud.
- **Warnings**
  Lets you display visual warnings.
- **Notices**
  Gives you aural or visual cues when accessibility features are turned on or off.
- **Schemes**
  Lets you associate computer sounds with the specific system events.
- **Captions**
  Lets you display captions for speech and sounds.

Keyboard

You can make the following keyboard adjustments:

- **Repeat Rate**
  Lets you set how quickly a character repeats when a key is struck.
- **Tones**
  Lets you hear tones when pressing certain keys.
- **Sticky Keys**
  Lets those who type with one hand or finger select alternative keyboard layouts.

Mouse

You can use the following options to make your mouse faster and easier to use:

- **Click Speed**
  Lets you select how fast to click the mouse button to make a selection.
- **Click Lock**
  Lets you highlight or drag without holding down the mouse button.
- **Reverse Action**
  Lets you reverse the functions controlled by the left and right mouse keys.
- **Blink Rate**
  Lets you select how fast the cursor blinks or if it blinks at all.
- **Pointer Options**
  Lets you complete the following tasks:
- – Hide the pointer while typing
  – Show the location of the pointer
  – Set the speed that the pointer moves on the screen
  – Select the size of the pointer and color for increased visibility
  – Move the pointer to a default location in a dialog

Keyboard Shortcuts

The following table lists the supported keyboard shortcuts:

Keyboard	Description
Ctrl+P	Print the current window
Alt+Left	Back
Alt+Right	Forward
Alt+Up	Up
Ctrl+R	Refresh a window

CA Service Operation Insights Keyboard Shortcuts

Keyboard	Description
Ctrl+N	Create a service
Delete	Delete a service or user
Ctrl+A	Create auditor
Ctrl+L	Locate the objects
Ctrl+T	Create an even policy
Delete	Delete a policy in the Event Policy window
E	Export a policy
S	Summary of the policy

Acknowledgments for CA SOI 4.2

The third-party software was used in the creation of CA SOI. All third-party software has been used in accordance with the terms and conditions for use, reproduction, and distribution as defined by the applicable license agreements. This section

contains all third-party software license agreements for applications that are included as part of the current release of CA SOI.

The license agreements of the following products are available as an attachment. Click here to download the license agreements.

1. activation 1.1
2. ActiveMQ 5.14.4
3. ActiveMQ 5.4.3
4. ActiveMQ Protobuf 1.1
5. Adobe Flex SDK 3.2
6. AdoptOpenJDK 1.8.0_212
7. Antlr 2.7.6
8. Antlr 2.7.7
9. Antlr 3.2
10. Antlr 3.3
11. AOP Alliance 1.0
12. Apache JCS 1.3
13. Apache Tomcat 7.0.77
14. Apache Tomcat 7.0.82
15. ASM 1.5.3
16. ASM 5.0.4
17. AspectJ 1.8.9
18. Axiom 1.2.7
19. Axis 1.4
20. Axis2 1.4.1
21. Axis2 1.5
22. backport-util-concurrent 3.0
23. Batik 1.6
24. bcprov 16-140
25. Black White Pearls Icons Social Media Logos 12-May-2009
26. BSAFE Crypto-J 3.6
27. BSAFE Crypto-J 5.0
28. CA Normalized Integration Management for Service Management (CA NIM SM) 3.2.0.0
29. Camel 2.4.0
30. Commons beanutils 1.6
31. Commons Cli 1.2
32. Commons Codec 1.10
33. Commons Codec 1.3
34. Commons Codec 1.4
35. Commons Collections 2.1.1
36. Commons Collections 3.1
37. Commons Collections 3.2.1
38. Commons Collections 3.2.2
39. Commons Collections VFS 2.0
40. commons dbcp 1.2.2
41. commons discovery 0.2
42. commons discovery 0.5
43. Commons FileUpload 1.2

44. Commons httpclient 3.1
45. Commons httpclient 4.1.1
46. Commons IO 1.4
47. Commons Lang 2.4
48. Commons Lang 2.5
49. Commons Lang 2.6
50. Commons Logging 1.1
51. Commons Logging 1.1.1
52. Commons Pool 1.3
53. concurrent utilities 1.0
54. concurrent utilities 1.3.4
55. Derby 10.3.1.4
56. DJNativeSwing .9.6
57. dom4j 1.6.1
58. drools 5.5
59. Eclipse EMF 2.2.3
60. Embedded Entitlements Manager (EEM - previously eIAM) r12.51CR02
61. End User License Agreement (EULA) r5.1.3-GA
62. Google Maps API 3.0
63. GWT 2.4.0
64. gwt-log 3.1.8
65. Hazelcast 1.8
66. Hibernate 3.2.6
67. httpclient 4.0.1
68. httpclient 4.1.1
69. httpcore 4.1
70. InstallAnywhere 2017
71. InstallAnywhere 2017 SP1
72. iText 1.4
73. jackson 1.9.2
74. jackson-annotations 2.5.0
75. jackson-annotations 2.6.3
76. jackson-core 2.6.3
77. jackson-databind 2.6.3
78. Jaspersoft JasperReports Server 6.3
79. Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 8
80. Java Mail 1.4
81. javax servlet api 2.3
82. jaxb-api 2.2
83. JAX-RPC 1.1
84. JDOM 1.0
85. Jersey 1.1.5
86. Jersey 1.12
87. Jettison 1.1
88. Jetty 7.6.18.v20150929
89. jmockit 0.991
90. JNA 3.0.0

91. jquery 1.4.2
92. jquery 1.6.4
93. jquery 3.0.0
94. jquery mobile 1.1.0
95. jquery mobile 1.1.0 RC1
96. jQuery UI 1.11.4
97. jQuery UI 1.8.1
98. JQuery.query 2.1.7
99. jquery-notify 1.5
100. JSON 20070829
101. jsr 250 1.2
102. JSTL (The JSP Standard Tag Library) 1.1.2
103. JSW (Java Service Wrapper) Professional 3.5.13
104. jTDS 1.2.2
105. Log4j 1.2.13
106. Log4j 1.2.15
107. Log4j 1.2.17
108. mimepull 1.3
109. mvel 2.1.3
110. mysql-connector-java 5.1.46
111. not-yet-commons-ssl 0.3.11
112. Oracle Java Runtime Environment (JRE) 1.8.0_45
113. OverlappingMarkerSpiderfier 1.6.2
114. protobuf-java 2.4.1
115. Quartz 1.6.2
116. rampart 1.4
117. rome 1.0
118. Saxon HE-9.6
119. scala-library 2.11.0
120. servlet-api 2.4
121. slf4j 1.6.4
122. Solr 1.4.1
123. spring framework 3.0.3.RELEASE
124. spring framework 3.0.4
125. SQL Server JDBC Driver 4.0
126. Standard Widget Toolkit (SWT) 3.4
127. StringTemplate 3.2
128. StringTemplate 3.2.1
129. Struts 1.0.2
130. Swing Wizard Builder 1.0
131. Tomcat 7.0.68
132. Tomcat 7.0.82
133. Tomcat 7.0.90
134. TomSawyer 8.0
135. Tuscany SCA 1.6
136. Tuscany SDO 1.1.1
137. Velocity 1.7

138whirlycache 0.7.1
139whirlycache 1.0.1
140Woodstox (wstx-asl) 3.2.9
141Woodstox 4.0.8
142ws02 registry 4.1.1
143ws-addressing xsd 2006/03
144wss4j 1.5.8
145xalan 2.7.1
146xercesImpl 2.12.0
147xercesImpl 2.8.1
148xercesImpl 2.9.1
149xml-apis 1.3.03
150xml-apis 1.3.04
151xml-apis 1.4.01
152xmlpull 1.1.3.4d_b4_min
153xmlsec 1.4.3
154XStream 1.4.1

CA Catalyst Acknowledgments

The attached document contains licensing agreement information for third-party software used in CA Catalyst. The following license agreements are available as an attachment. Click here to download the license agreements.

1. activation 1.1
2. ActiveMQ 5.7.0
3. AdoptOpenJDK 1.8.0_212
4. Apache CXF 2.7.1
5. Apache JCS 1.3
6. api-bridge 1.0.1.ca-3
7. Aries Transaction 0.2-incubating
8. Avalon Framework 4.1.3
9. Bouncy Castle 1.47
10. BSAFE Crypto-J 5.0
11. CA Directory r12 SP10
12. cglib-nodep 2.1.3
13. Commons beanutils 1.7
14. Commons Cli 1.0
15. Commons Codec 1.3
16. Commons Collections 3.2.2
17. Commons Collections VFS 1.0
18. commons dbcp 1.2.2
19. Commons Digester 1.8
20. commons el 1.0
21. Commons FileUpload 1.2
22. Commons httpclient 3.1
23. Commons IO 1.4
24. Commons Jexl 2.0.1
25. Commons Lang 2.4

26. Commons Logging 1.1.1
27. Commons Pool 1.5.4
28. concurrent utilities 1.3.4
29. Derby 10.9.1.0
30. drools 5.0
31. Eclipse equinox OSGi 3.6.0
32. Embedded Entitlements Manager (EEM - previously eIAM) r8.4SP4CR12
33. geronimo 1.1.1
34. H2 1.2.147
35. Hazelcast 1.8.3
36. httpcore 4.0-alpha2
37. InstallAnywhere 2017 SP1
38. IPv6 Cookbook 1
39. JAF 1.1.1
40. Java Mail 1.4
41. JDOM 1.0
42. Jersey 1.1.5
43. Jettison 1.1
44. Jetty 7.2.2
45. jline 0.9.9
46. Joda-time 1.6
47. JSTL (The JSP Standard Tag Library) 1.1.2
48. JSW (Java Service Wrapper) Professional 3.5.13
49. JTA 1.0.1B
50. JWSDP 1.3
51. Karaf 2.2.5
52. Log4j 1.2.17
53. OData4J 0.6
54. openjpa 1.2.0
55. OpenSAML 1.1b
56. ORO 2.0.8
57. Pluto 2.0.2
58. Quartz 1.6.2
59. rampart 1.4
60. rome 1.0
61. ServiceMix 3.3.1
62. servlet-api 2.5
63. Simple Logging Facade for Java (SLF4J) 1.5.10
64. Solr 1.4
65. spring framework 3.0.4
66. Standard Widget Toolkit (SWT) 3.3.2
67. StAX 1.2
68. Tomcat 7.0.62
69. Tuscany SDO 1.1.1
70. whirlycache 1.0.1
71. Woodstox (wstx-asl) 3.2.0
72. ws-addressing xsd 2006/03

73. WSDL4J 1.6.2
74. Xalan-J 2.7.1
75. xbean-blueprint 3.7
76. xbean-classloader 3.5
77. xercesImpl 2.8.1
78. Xerces-J 2.9.1
79. XMLBeans 2.2.0
80. XStream 1.3.1

Release Comparison

This table compares the key features in all the active releases for CA Service Operations Insight:

Key Features	Release 4.2	Release 4.0 SP2	Release 4.0 CU1	Release 4.0
Import Services Using WSSSAServiceCmd Command	**yes**	no	no	no
Enhancement of Relationship Visibility in Topology View	**yes**	no	no	no
Enhancement in Cleared Alert History tab	**yes**	no	no	no
Decouple of MQ Server for Standalone CA SOI Components and High Availability	**yes**	no	no	no
Support for CA Catalyst r3.4.2	**yes**	no	no	no
Support for Microsoft SQL Server 2014 Enterprise	**yes**	no	no	no
Support for Night View Option for Google Maps	**yes**	no	no	no
Enhancement of Relationship Visibility in Topology View	**yes**	no	no	no
SOI Reports Dashboards	**yes**	**yes**	no	no
Introduced Link for Google Map,?replaces Google Earth link	**yes**	**yes**	no	no
Added Four Commands to SOIToolbox	**yes**	**yes**	no	no
Execute SOIToolbox Commands without Stopping CA SOI services	**yes**	**yes**	no	no

Set custom ticket properties to Request Area and create ticket in CA Service Desk Manager	**yes**	**yes**	no	no
CA SOI Service Alarms Contain TenantID	**yes**	**yes**	no	no
Support for Common Access Card (CAC) Support	**yes**	**yes**	no	no
Support for Latest Version of CA Business Intelligence	**yes**	**yes**	**yes**	no
Performace Improvements • System startup time • CA SOI Operation Console • Startup time of SA Manager, shutdown time, and alarm processing	**yes**	**yes**	no	no
Support for Java 8 and Tomcat 7.0.62 to handle POODLE vulnerability	**yes**	**yes**	**yes**	**yes**
Display or Hide the Progress Bar when you Take Action on Alerts	**yes**	**yes**	**yes**	**yes**
Infrastructure Alerts added to the Alert Queue that is sent to SOI Enterprise Manager by Domain Connector	**yes**	**yes**	**yes**	**yes**

Performance Results

This article provides the performance results of CA SOI.

The environment details are as follows:

Infrastructure	
SystemType	Virtual Machines
Processor on All Virtual Machines	Intel® Xeon® CPU E5-2680 v2 @ 2.8 -GHz
Operating System	Windows Server 2012 R2 Standard

System Usage	CPU	Memory	Disk Space
Database	8	16 GB	100 GB
UI Server	4	8 GB	100 GB
SA Manager	8	16 GB	100 GB

Catalyst Container	4	8 GB	100 GB

Total Number of Created Escalation Policy and Escalation Action	
Alerts Escalation Policy	18
Alerts Escalation Action	33

For the mentioned environment details the results are as follows:

Alert Processing

Alert Queue Size	Time to Process
25000	14.2 seconds
50000	14.33 seconds
75000	45.46 seconds

SA Manager Startup Time

CIs	**SA Manager Startup Time With Cache**	**SA Manager Startup Time Without Cache**
894486	4.48 minutes	32.25 minutes
919020	2.48 minutes	30.59 minutes
944020	4.42 minutes	33.28 minutes
969020	2.3 minutes	34.42 minutes
994020	3.23 minutes	36.02 minutes
1019020	4.13 minutes	37.4 minutes
1044020	3.34 minutes	39.5 minutes
1069020	7.12 minutes	42.11 minutes
1094020	4.31 minutes	50.13 minutes

Getting Started

This section provides information about what CA SOI does, the basic concepts, and how to get started with the product. Read this material before installing and working with CA SOI.

CA Service Operations Insight

When degradation or downtime affects a key service, customers quickly become frustrated. Whether they are external customers or your own employees, poor service has a negative impact.

Domain management solutions monitor various aspects of a service, including support for IT infrastructure components or the end-user experience. None of these individual solutions give you a complete, end-to-end view of service health and availability across all management domains. Operations personnel often guess how the fault or performance issues reported across the network, systems, database, or application monitoring tools actually affect key IT services, degrade service quality, or increase the risk of an outage. Similarly, service stakeholders may not understand whether IT enables them to fulfill their business objectives.

CA Service Operations Insight (CA SOI) helps overcome these challenges by unifying the health and availability information from your domain management tools and aligning with your IT services. CA SOI introduces a new service management layer to your management infrastructure and through an open and extensible integration platform (CA Catalyst), leverages and adds value to your investment in existing management technology. CA SOI provides integrations with several CA Technologies products and third-party applications, and the CA Catalyst integration platform lets you reconcile and synchronize data in CA SOI and across all domain managers. CA SOI uses several graphical interfaces to display the service operations data that supports the required business functions for all parties in the appropriate format. Operations staff uses these graphical interfaces to focus efforts correctly and business and IT objectives are properly aligned.

CA SOI also serves as a comprehensive level one operations console for managing the full stream of events and alerts from all integrated products. Operations staff can use CA SOI for a consolidated view of all alerts, enabling automatic escalation of important alerts that require quick action and problem resolution across domains from one interface. CA SOI provides alert queues for grouping logical categories of alerts. CA SOI also provides an event management layer that supports detailed event searches. CA SOI has several graphical interfaces for defining simple and complex event policies for event filtering, correlation, and enrichment.

CA SOI supports layered service and alert security through the use of user groups, customers, and alert queues. This security allows for a flexible user-specific view of the services and alerts company-wide.

Understanding Your Role in CA SOI

As a CA SOI user, you fulfill one or more of the following roles:

- **Administrator**
  The CA SOI administrator is responsible for installing, populating, and maintaining CA SOI. The CA SOI administrator responsibilities include configuration, user management, service modeling, and maintenance.
- **Integration Developer**
  The CA SOI integration developer is responsible for getting domain manager data into CA SOI and integrating CA SOI with external sources.
- **Operator**
  The CA SOI operator is a non-administrator CA SOI user who is responsible for viewing and responding to alerts.

Where to Begin in CA SOI

Once you understand your role, become acquainted with the information suited to your role.

Follow these steps:

1. All roles should review the main CA SOI Terminology and Concepts.
2. Depending on your role, perform one or more of the following tasks:
   – As an administrator, you install SOI and its components. Review the CA SOI Administration Process to understand the administrator role in configuring and administering CA SOI.
   – As an administrator or operator, you can log into CA SOI and become familiar with the Dashboard and the Operations Console. You can also access the Mobile Dashboard, reports, and USM Web View.
   – As an integration developer, you want to learn about building and integrating domain manager connectors with CA SOI.

CA SOI Terminology and Concepts

CA SOI introduces the following terms:

Service

The concept of a service model, or service, is central to CA SOI. A *service* typically consists of several CIs, which you can group to represent, for example, web server farms or clusters. Services can also contain subservices, which are subordinate service models. Subservices are previously created services that are reused as building blocks of another service. Service models typically represent high-level abstract entities like a web-based retail transaction service, an application server service, a printing service, or a routing service. You can define any type of service with CA SOI as long as one of the integrated domain managers monitors the service components.

CA SOI provides a comprehensive understanding of how a fault condition, which CA SOI represents as an infrastructure alert, impacts the business. Consider a managed resource such as a router. You can accurately, but narrowly, define it as a device that forwards data from one network to another. From a service perspective, however, a router is an indispensable component among other cooperating components that support interconnected business activities.

When router performance is compromised, the activities that depend on that router are likely compromised also. A router can be associated with other network devices such as switches or servers, which are associated with the applications or databases that they host. These relationships and dependencies comprise the logical and physical topology of the service. CA SOI lets you incorporate these relationships in the service model. These relationships help you capture how one CI relates to another and how they collectively deliver the service logic.

Service models contain policy that determines how alert conditions on one CI can impact related items and the service itself. You can modify and extend this policy to refine the model and capture the collective behavior of all associated entities.

You can reuse a service model any number of times. You can also combine it with other configuration items (CIs) and services to build higher-level service models. For example, the DNS service can be critical to several higher-level services such as Microsoft Exchange and SAP. Similarly, Exchange can itself form part of still higher-level services such as email or Blackberry.

You can define new service models, import them from domain managers, or define policies that automatically discover and create services according to specified criteria. For example, an operator working with a service owner can select configuration items that are discovered through integration with the domain managers. The operator can then create relationships among those configuration items. Similarly, if a service model is defined in a domain manager, you can import that service model and all its topographic information directly into CA SOI. You can extend or combine imported service models in the same manner as the service models defined in CA SOI. This ability provides a powerful mechanism to leverage your existing investment.

Configuration Item (CI)

A configuration item (CI) is a collection of information about a managed resource such as a printer, software application, or database. CA Catalyst uses an instance of a USM type for each CI. Connectors transform CIs between a domain manager format and USM.

CA SOI provides a view of CIs across all management domains in a single place, and provides a unified view from all the perspectives in which a CI is managed. CA Catalyst correlates and reconciles CIs managed by multiple domain managers so that CA SOI maintains one CI with a unified set of properties.

Impact

Impact indicates how much a CI affects a service and related CIs.

The following factors determine impact:

- CI severity
- CI significance
- Propagation type and policy

Impact provides IT personnel with a good understanding of what fault conditions really mean to the services that CIs support.

CA SOI calculates the impact by multiplying severity by significance. Significance is a number from 1 (lowest) to 10 (highest), and severity ranges from 0 (Normal) to 4 (Down). Therefore, the highest possible impact is 40. Consider an application with a severity of 4 and a significance of 4. The resulting impact is 16.

Note: When a custom propagation policy connects CIs, you can define complex rules for changing the severity value. For more information, see Define Custom Propagation Policy.

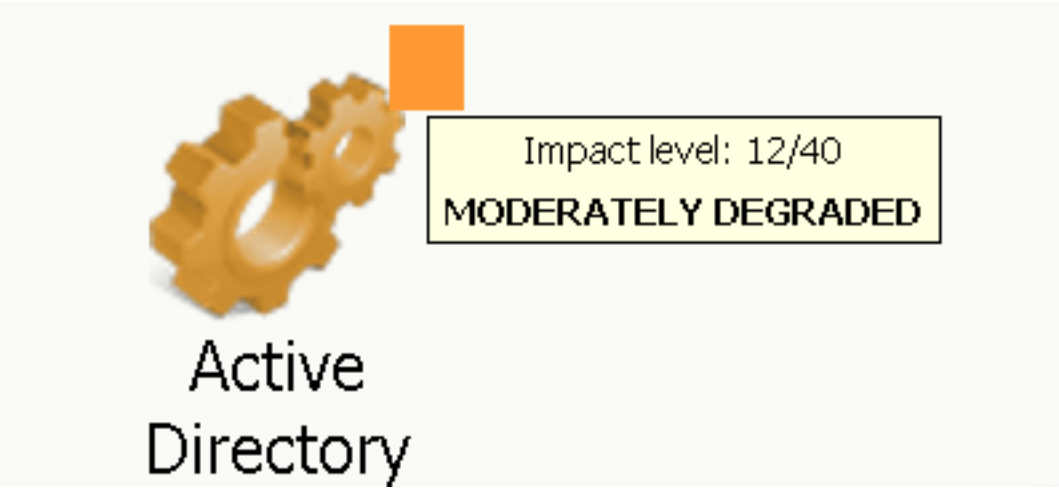
Consider the following items:

- If you change the significance of an item, the impact changes after a new alert is received.
- Priority is used in the calculation of impact instead of significance in the following situations:
  - The service is a top-level service
  - The significance of the parent relationship is zero

The following table defines the impact ranges:

Impact	Color	Description
0	Green	Operational
1-10	Yellow	Slightly Degraded
11-20	Orange	Moderately Degraded
21-30	Red	Severely Degraded
31-40	Burgundy	Down

The Topology tab of the Contents pane shows the impact color as a small box above and to the right of a service, CI, or group. Mouse over the box to display the impact value.



The impact number shows on the small box when you click Chart Display Complexity Level

and switch to the Advanced view.

The color of the service icons in the Operations Console indicates the quality impact of the service. *Quality impact* is the inverse value of service health, which displays on the Dashboard and the Component Detail pane of the Operations Console.

Significance

Significance indicates the importance of a CI.

In a service, many CIs can affect another CI or service. Although each CI affects the health of the service, some CIs can be more important than others. For example, it is important that a print server is available for an online order service. However, the print server does not affect order processing as much as the inventory database does. In this case, the print server has a lower significance than the inventory database.

Significance is a property of a CI in the context of a service. Each CI type has a default significance, but the actual significance is stored in the relationship, thus providing significance with a service scope. When you change significance for an individual CI in a service, the CI's significance value only changes within that service. If the CI belongs to other services, it retains its previous significance value in those services. This behavior ensures that you can maintain different significance values for the same CI if it is more or less important to other services.

CIs can be in as many services as needed and can affect each service differently, based on the significance setting. Suppose that the Payroll department also uses the print server to print checks. Therefore, the print server has much higher significance.

Significance is a value from 1 through 10, where 1 is the least significant and 10 is the most significant. The numeric value denotes how important a child object (antecedent) is to the functioning of its parent object (dependent). Each CI is assigned a significance when it is imported from a domain manager.

CA SOI assigns a default global significance value to every CI type available in the product. For instance, all servers of a Computer System type added to a service model have a significance of 5, while switches and routers have a value of 9. The types Service, Network, and Operating System are considered to be the most important and have a significance of 10.

You can set significance globally or for individual CIs and relationships.

Infrastructure Alerts

A domain manager reports an *infrastructure alert*, which is a fault condition on a CI in CA SOI.

All infrastructure alerts begin their CA SOI lifecycle as events. Events that have a severity greater than normal and come out of Event Management event policy filtering become infrastructure alerts.

CA SOI automatically associates infrastructure alerts with their corresponding CI and assigns to each alert condition a severity that determines the CI color on the Operations Console. One CI can have several alert conditions simultaneously, and the alert with the highest severity determines the impact on the CI and its color. When the alerted CI belongs to a service, CA SOI calculates the impact value from the seriousness of the fault condition and the importance of the CI to the services it supports.

Infrastructure alerts typically include a URL so that an operator can navigate in context from the Operations Console to the originating domain manager and can view the alert in its original context.

Infrastructure alerts belong to one of the following categories:

- **Quality**
  Indicates the level of excellence that consumers of a resource experience. For example, performance degradation detected by CA APM takes the form of a quality alert.
- **Risk**
  Indicates the likelihood of delivering the service quality that is required to support business objectives. For example, an alert specifying that a computer system has low disk space is a risk alert. If no category is defined, risk is the default category.

These categorizations help determine the quality, risk, and health value for any associated service.

Depending on their service association, infrastructure alerts can appear as the following types:

- **Service impacting**
  Infrastructure alerts are service impacting when they affect a CI that is part of a managed service. You can view these alerts when viewing the service in the Operations Console or other interfaces.
- **Non-service impacting**
  Infrastructure alerts are non-service impacting when they affect CIs that are not part of a managed service. These alerts appear under associated alert queues on the Alert Queues tab of the Operations Console. If the alert does not meet the criteria of any defined alert queues, it appears as a part of the Default queue. You can perform the same operations on non-service impacting alerts (assignment, escalation, and so on) as you can on service impacting alerts.

Service Alerts

A *service alert* is an alert condition that CA SOI generates based on analysis of a modeled service that it is monitoring.

Service alerts result when the condition of one or more CIs combines to impact the overall service quality or risk level. The policy that you define for that service model determines how CI alert conditions impact other CIs and the overall service.

You can use the Alert and Topology Views of the Operations Console to view the root cause infrastructure alerts that caused the service alert. You can also view the root cause type: root cause, symptom, or unclassified.

Health, Quality, and Risk

Health, quality, and risk are the primary metrics exposed to Dashboard and external interfaces for monitoring service status. They categorize service impact values to reflect the type of outage or impact according to alert categories.

Alerts impacting a service belong to one of the following categories:

- **Quality**
  Indicates the level of excellence that consumers of an IT service experience, whether they are other IT services, customers, or end users. The quality levels are Operational, Slightly Degraded, Moderately Degraded, Severely

Degraded, Down, and Unknown. The highest propagated impact of an associated quality alert determines the service quality value.

- **Risk**
  Indicates the likelihood of delivering the quality of service required to support the overall business objectives. The risk levels are Down, Severe, Moderate, Slight, None, and Unknown. The highest propagated impact of an associated risk alert determines the service risk value. If an alert has no defined type, it is a risk alert by default.

Service health is the highest impact held by quality or risk. The following table shows the available Health, Quality, and Risk values:

Health	Quality	Risk
Normal	Operational	None
Minor	Slightly Degraded	Slight
Major	Moderately Degraded	Moderate
Critical	Severely Degraded	Severe
Down	Down	Down

For example, a slightly degraded service with a severe risk of degradation would have a service health of Critical.

Severity

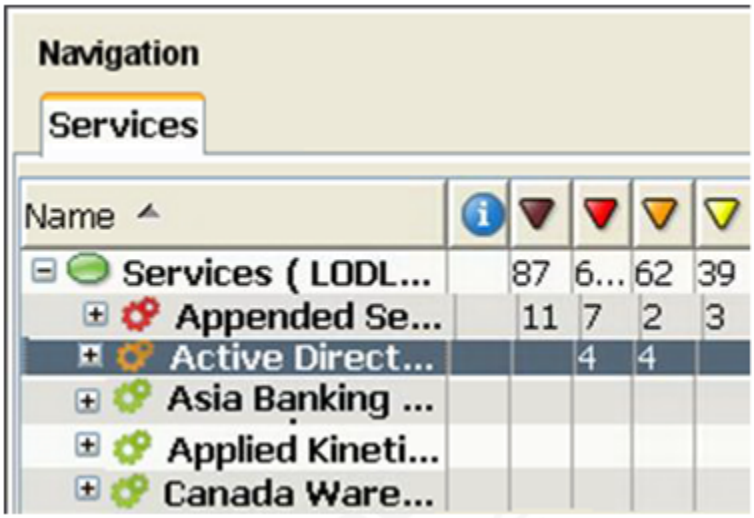
Severity indicates the condition of a CI as reported from the domain manager to CA SOI through alerts.

If multiple domain managers send alerts for the same CI, the highest severity is used. CI severity helps determine the service impact by propagating the impact of the condition to related CIs in the service model according to propagation settings.

The following table describes each severity:

Severity	Color	Description
Normal	Green	Operational
Minor	Yellow	A nominal displacement of CI function that can require an inspection
Major	Orange	A serious causal change typically leading to degradation of function
Critical	Red	High probability of imminent failure and severe degradation of service
Down	Burgundy	The CI is incapable of providing function or service

Color-coded icons on the Operations Console indicate CI severity (the color-coded icons for services indicate the service impact). Alerts in the Contents pane have the color corresponding to their severity. The Navigation pane also represents severity in columns next to services and CIs. The following graphic shows that each column lists the number of items with the corresponding severity (represented by the colors in the previous table).



NOTE
If no alerts are raised for a CI, its severity is green even if the device contains child CIs with different severities. Also, groups are simply containers and would not usually have alerts. You can expand the tree and can follow the numbers to the row that lists the item whose severity you are looking for.

Granularity

CA SOI supports granularity at two levels: Normal and Low.

Normal granularity mode represents an explicit modeling principle where alerts are presented in CA SOI only if all the impacted CIs are included in the model. For example, consider a computer system as CI, which is running many service-supporting resources. When the service granularity is Normal, only the resources that are directly modeled with the CI are included to show their alerts as impacting the service.

Low granularity mode represents a mixed modeling principle where you manage the service granularity as Normal and Advanced:

Normal

Only the parent CIs are included and no associated child CIs are included in the service. In this case, parent CIs act as aggregators for all alerts that affect them directly or indirectly through their commonly related resources. For example, consider the scenario of a computer system as CI, which is running some service-supporting resources. In this case, you only include the computer system, and any alerts affecting any of the resources hosted on the computer system are aggregated to the computer system.

A parent CI consists of a specific child CI (not all child CIs) in the service. In this case, any alerts that are directly associated with the included child CI is aggregated to those CIs. The alerts that are associated with the nonmodeled child CIs are aggregated to the corresponding parent CIs. For example, consider a scenario in which a computer system is a CI, which is running some service-supporting resources and one of the service-supporting services – CPU - is included in the service model as child CI of the computer. If any CPU-related alert comes in, it is associated directly to the CPU. If any other alert from the other service-supporting resources comes in, they get attached to the parent computer system CI. Therefore, including granular child CIs does not change the low granularity of the service model for excluded child CIs, enabling mixed-mode low granularity modeling.

Advanced

Only the parent CIs are included and no associated child CIs are included in the service In this case, parent CIs act as aggregators for alerts that affect them directly and class significance of child CIs act as aggregators for alerts that affect them indirectly through their commonly related resources. For example, consider the scenario of a computer system

running three service-supporting resources [Application, Memory, and Processor]. In this case, you include the class significance of the service-supporting resources and alerts affecting any of the resources hosted on the computer system.

A parent CI consists of a specific child CI (not all child CIs) in the service. In this case, any alert that is directly associated with the included child CI is aggregated to that child CI. The alerts that are associated with the nonmodeled child CIs are aggregated on the basis of the class significance of the child CIs and get attached to the parent computer system CI. For example, consider a scenario in which a computer system is a CI, which is running some service-supporting resources and one of the service-supporting services-CPU-is included in the service model as child CI of the computer. If any CPU-related alert comes in, it is associated directly to the CPU. Any other alert not directly associated with the CPU comes in, they are calculated based on the class significance of the child CIs and get attached to the parent computer system CI.

NOTE

To configure the settings for low granularity, check the Modeler settings in the Configure Global Settings page.

Events

An *event* is a message that indicates an occurrence or change in your enterprise. Events can indicate a negative occurrence or object state change. CA SOI Event Management lets you view and manage events that are received from all connectors.

CA SOI collects events from various types of event sources:

- Domain managers that manage alerts indicating problems with their domain. Domain managers can include CA Spectrum for network faults, CA eHealth for network performance, and CA Application Performance Management for application performance
- High-volume raw event sources, such as SNMP traps and IBM Tivoli Netcool

All collected events and alerts initially become events in CA SOI and are maintained in Event Stores that are distributed across the environment. The CA SOI Event Management component lets you manage a large event stream by exception using event policy to correlate, filter, and enrich events from any or all event sources. Event Management lets you control the types of information from the event stream that are displayed as actionable CA SOI alerts.

Alert Queues

Alert queues are user-defined alert groups. CA SOI auto-assigns alerts to a particular alert queue based on user-defined policy, which can include alert content and associated CIs.

Alert queues let you group alerts as they come in based on specific criteria to monitor the status of your infrastructure more efficiently. You can add global and non-global escalation policies to alert queues to take a specified action automatically on alerts that come into a queue.

For example, consider a company with engineers responsible for different aspects of the infrastructure, such as networks, systems, and databases. Without defined queues, alerts from all integrated domain managers appear in one consolidated view on the Alert Queues tab. The administrator can define queues that are based on a domain (Network Alerts, Database Alerts, and so on). Engineers can then find and resolve their alerts quickly. The administrator can define additional queues that are based on other alert categories, such as severity, assignment status, or description for an optimized unified alert management system.

Services provide a similar organizational function as alert queues at a higher level with the additional benefit of resource topology and impact analysis. Defining alert queues is less intensive than modeling services, and they can simplify alert management as you make the transition to a service-oriented management paradigm. Alert queues are also useful in an environment with services defined to provide a supplemental management perspective outside of services. For example, you can define a queue for alerts that are not acknowledged or a queue for alerts from the same source domain manager.

Alert Escalation Policy

Escalation policy specifies the automated actions to take in response to fault conditions.

Escalation policy consists of the following items:

- **Policy type and assignment**
  Defines the alerts that the policy evaluates. Escalation policy can be global, or you can assign the policy to evaluate only alerts in specified services or alert queues.
- **Policy criteria**
  Defines criteria that an alert must match for the specified action to occur. Criteria can be type-based, time-based, and attribute-based, and the policy can also have an associated schedule.
- **Escalation action**
  Defines the action to perform when an alert meets the policy criteria.

Use escalation policies to automate the response to common alert conditions and therefore decrease the time that is required to resolve problems.

Customers

A *customer* in CA SOI is any consumer of a managed service. The CA SOI administrator creates customers and associates them with service models to see the impact of service degradation on the customer.

Customer management provides an extra layer of security and insight into how end users dependent on provided services are affected when the services experience downtime or degraded performance.

For example, you can define a customer as a particular region of your company such as Europe, so that operators in that Europe see only the services and alerts particular to Europe. Similarly, you can define a customer that represents a division of your company. You can further define sub-customers, that represent entities within that division such as human resources and accounting.

For more information about customers, see How to Create and Manage Customers.

CA SOI Administration Process

CA SOI adds a service and operations management layer above the existing domain managers in your enterprise. CA SOI uses connectors and the overall CA Catalyst infrastructure to connect to the domain managers. Each domain manager is unique in the type of resources it manages, how it represents those resources, the alerts it raises, and so on. CA SOI translates this disparate data to standardized information to simplify the data visualization and issue resolution process.

The typical CA SOI service creation and administration workflow, from initial configuration to modeling to detailed reporting, is as follows:

1. Configure security: After installation, an administrator sets up security that defines the users and roles that will interact with the system.
2. Define event policies: Create the event policies optimize the quality of resultant alerts. For example, enriching alerts with contact information or creating an alert that is based on correlated conditions.
3. Define the alert queues: Establish the queues and policies that determine how operators visualize and manage alerts.
4. Define the service: Build service models in the Service Modeler that you open from the Operations Console.
5. Manage user group access: After a service definition, the administrator defines user groups and each group's access privileges.The access privileges determine the features user groups can access: CA SOI features, services, customers, and alert queues. For example, a person responsible for monitoring the Payroll Service may need to view or access only HR-related services. Likewise, if CA SOI is monitoring services for several internal or external customers, each customer should have access to their own information only.

6. Validate the service: The administrator publishes a fully configured service so that CA SOI can begin to manage or instrument the service:
   a. The instrumentation process begins by determining the current active infrastructure alerts for each CI associated with the service model across all integrated domain managers. This process determines the overall state of the CI based on the highest impacting infrastructure alert condition.
   b. Next, the propagation type and policy determine how the infrastructure alerts propagate across the model, and ultimately how they impact the service itself. If the service is impacted, one or more service alert conditions appear and the root cause information helps to diagnose and fix the problems. CA SOI also determines how an alert condition could impact a service by considering service quality, service risk, and overall service availability.
   c. Alerts can enable a launch-in-context to the application reporting a fault condition, letting operators to gather more information to help diagnose and resolve an issue.
7. Define the customers: The administrator defines the customers to determine the system degradation impact to a specific customer.
8. Manage alerts: As CA SOI detects infrastructure or service alert conditions, alerts appear on the Operations Console. The alerts are associated with services and alert queues and behave according to the associated escalation policies. A single infrastructure alert condition can affect multiple services (if the associated CI supports more than one service) or alert queues. Therefore, more than one alert escalation policy can be associated with the alert. Alert escalation policies automate the following escalation actions.
9. View the service information: Conditions that impact a service are reflected across all views that may be supported for that service. The conditions are in the Operations Console and the Dashboard.
10. Report the service details: You can run, configure, and schedule predefined reports on the service models to help managers make business decisions. The reports also show operators historical service and resource status. Reports can help you understand the impact of fault conditions and predict future issues that are based on past performance by spotting trends and chronic fault conditions.

Access the CA SOI Dashboard and Operations Console

Use a web browser to access the CA SOI Dashboard.

Follow these steps:

1. Open a web browser and enter the following URL:
   ```
   http://<UI server>:<port>/sam/ui
   ```
   – *UI server*
     Specifies the name of the system where the UI Server is installed.
   – *port*
     Specifies the port on which the UI Server listens.
     Default: Non-SSL connection – 7070; SSL connection - 7403

   If you entered an SSL port, a security certificate dialog opens.
2. (Optional) Click Yes to accept the certificate for an SSL connection.
   The login screen opens.
3. Enter a valid user name and password.
   Default: samuser, defined during installation.

   The CA SOI user interface opens to the Dashboard tab by default.
4. Click one of the following items:
   – **Dashboard tab**
     Provides graphical information about service status. If an administrator has not defined any services, the dashboard shows no data.
   – **Administration tab**
     Displays the Administration page, which lets administrators configure connectors and SA Manager settings.
   – **Reports link**

Displays the JasperReport Server Interface. You must configure reporting before the Reports link is available. For information about setting up reporting, see Configure CA SOI Reports for CABI JasperReports Server. For information about working with reports, see Generating Reports [4.0 CU1].

– **Console link**
  Displays the Operations Console, where you can model services, monitor service status, and view and manage alerts. A Java application runs briefly when you start the Operations Console.For information about configuring the Operations Console, see Operations Console Customization. For information about working in the Operations Console, see Operations Console Basics.
– **Google Maps link**
  You can view CA SOI services in Google Maps. The CA SOI services are displayed according to their location property.For information about configuring Google Maps, see Configure Google Maps Integration. For information about working with Google Maps, see View Services in Google Maps.
– **USM Web View link**
  Displays the USM Web View interface, which lets you search, browse, and interact with the store of USM data. For information about configuring USM Web View, see Configure USM Web View Integration. For information about working with USM Web View, see USM Web View for PC and USM Web View for Mobile Devices.

Create Operations Console Desktop Shortcut

You can create a shortcut to the Operations Console so that you can access the interface from an icon on your desktop and avoid accessing from the Dashboard. Start the Operations Console at least once from the Dashboard on your system before creating the desktop shortcut.

Follow these steps:

1. Open a command prompt and navigate to the location of Java on your system as follows (or navigating to the system32 directory on Windows systems):

   NOTE
   If you navigate to the location of a specific Java directory, it must be the same Java that CA SOI uses.

   ```
   cd Program Files\Java\jre6\bin
   ```

2. Enter the following command:

   ```
   javaws.exe -viewer
   ```

   The Java Cache Viewer page opens.
3. Right-click the CA Service Operations Insight Console entry and select Install Shortcuts.
   Your desktop has the CA Service Operations Insight Console shortcut icon. You double-click the icon and enter CA SOI credentials to access the Operations Console.

   NOTE
   If a proxy server login page opens when you start the Operations Console or during console operation, use Java 1.7.0_55 or below.

Access the Mobile Dashboard, Reports, and USM Web View

Contents

Once an administrator has configured the product, users can access the mobile dashboard, reports, USM Web View, and the CA Catalyst Registry.

Access the Mobile Dashboard

The Mobile Dashboard provides a mobile device view of all service metrics that are displayed on the Dashboard. You can also perform other actions such as viewing service contents, viewing alerts, and submitting help desk tickets. For information about configuring the Mobile Dashboard, see Configure Mobile Dashboard Integration.

Follow these steps:

1. Enter the following URL on the mobile device:
   ```
   http://<ui-server>:<port>/mobile
   ```
   Refer to your Installation Worksheet for the following values:
   – **ui-server**
     Specifies the UI Server host name.
   – **port**
     Specifies the Tomcat server port number specified during UI Server installation.
     Default: Non-SSL connection – 7070; SSL connection – 7403.
2. Enter valid credentials in the login dialog.
   The Mobile Dashboard home page opens. For information about working with the Mobile Dashboard, see CA SOI Mobile Dashboard.

Access CA SOI Reports

You access the JasperServer Reports interface to run CA SOI reports. The Reports link on the Dashboard is not available until you configure reporting. For information about configuring report integration, see Configure CA SOI Reports for CABI JasperReports Server.

Follow these steps:

1. Log in to the Dashboard.
2. Click the Reports link.
3. The Jasperreport Server login page opens.
4. Enter valid credentials and click Login.
5. Expand root, public, ca, Service Operations Insight, report in the Folders pane.
   Use the CA Service Operations Insight, Alert Management Reports, Detail Reports, drill down reports, and Top Ten Reports folders to view the CA SOI reports. For more information about managing reports, see Generating Reports [4.0 CU1].

Access USM Web View through the Dashboard

You access the USM Web View Starting Page through the CA SOI Dashboard so you can search or browse USM data. For information about configuring USM Web View, see Configure USM Web View Integration.

> **NOTE**
> The SSO (Single Sign-on) for USM Web View link does not work for the "samuser" user. However, you can use the "samuser" user to log into USM Web View by entering the credentials at the login prompt.

Follow these steps:

1. Log in to the Dashboard.
2. If the USM Web View link is active, skip to Step 9.
3. Click the Administration tab on the Dashboard.
   The administration options appear in the left pane.
4. Click the plus sign (+) next to CA Service Operations Insight UI Server Configuration.
   The available UI Servers appear.
5. Click the plus sign (+) next to the server you want to configure.
   The UI Server configuration options appear.
6. Click USM Web View Configuration.
7. Enter the server name in the USM Web View Server field, enter the port if you specified one other than the default in the USM Web View Port field, and click Save.
   The USM Web View connection settings are saved.

8. Click Refresh.
   The USM Web View link becomes active.
9. Click the USM Web View link.
   The Starting Page opens.
   Click the help link to access the online help or see USM Web View for PC.

Access USM Web View by URL

You access the USM Web View Starting Page by URL so you can search or browse USM data.

Follow these steps:

1. Open a web browser and enter the following URL:
   ```
   http://<ui-server>:<port>/ssaweb/m
   ```
   Refer to your Installation Worksheet for the following values:
   – **ui-server**
     Specifies the UI Server host name.
   – **port**
     Specifies the Tomcat server port number specified during UI Server installation.
     Default: Non-SSL connection – 7070; SSL connection – 7403.
2. Enter login credentials in the login dialog and click OK.
   The Starting page displays.
   Click the help link to access the online help or see USM Web View for PC.

Installing

This section describes how to install CA SOI and all required components.

Installation Planning

This section describes how to plan your CA SOI installation.

Components

Contents

CA SOI includes components that let you monitor services and resources, configure the product, add user groups, and perform other actions.

Integration Framework

The *integration framework (IFW)* is the mechanism that CA SOI uses to connect to domain managers and gather CI, service, topology, and state information.

It exists on any system with a connector or the SA Manager, and it interfaces with the connector framework to prepare connector data for transmission to the manager components. The IFW contains a transformation engine that uses a connector policy to transform connector data to the USM format. The IFW also includes the infrastructure of the Event Management component. The component provides the mechanism for storing events from connectors for exposure to event policy and eventual display as alerts after event processing completes.

The IFW uses the Apache ActiveMQ message broker, which fully implements the Java Message Service (JMS) as its protocol.

MQ Server

Apache ActiveMQ is an open source (Apache 2.0 licensed) message broker that fully implements the Java Message Service 1.1.

The MQ Server controls all messaging and communication from external sources. The server also receives alerts and CI information from connectors through the IFW and sends this information to various components for storage and analysis. Install MQ Server before any other CA SOI components.

Connectors

A connector is software that provides the interface for the data exchange between the CA Catalyst infrastructure and a domain manager. Connectors are the gateway through which data is retrieved from various domain managers for a consolidated management. Each integrated product has its own connector that supports one or both of the following operation types:

- **Outbound from connector**
  Outbound from connector operations obtain data (such as services, CIs, topology, alerts, and status) from the source domain manager. All connectors must implement outbound operations. Outbound data populates the CA Catalyst Persistence Store.
  Outbound data flows to one or more clients. Clients such as CA Catalyst consume the data to implement a unified view of data from multiple domain managers and their connectors.
- **Inbound to connector**

Inbound to connector operations (also referred to as "southbound") use records in the CA Catalyst Persistent Store and the CA Catalyst Synchronizer to create, update, or delete items in the source domain manager. The inbound operations enable domain manager synchronization with the changes that CI reconciliation, CI creation, and CI updates initiate in other domain managers.
Many provided connectors support inbound operations. Connectors that implement inbound operations sometimes limit the implementation to a subset of the types and properties their outbound operations support.
A *bidirectional connector* supports both inbound and outbound operations. Outbound-only connectors contain one connector policy file that transforms the gathered data to the standard USM format. Bidirectional connectors contain two connector policy files that transform outbound data to the USM format and transform inbound data to the source format of the domain manager.

You can configure connectors and start and stop them by accessing the CA SOI Administration UI.

CA SOI also provides the following tools for defining custom integrations:

- The Universal connector that can retrieve services, CIs, and status events from various CA Technologies and third-party products. The Universal connector provides a web services interface that products can use to publish new services, CIs, and events, which are normalized to a common format and made available to the SA Manager.
- A connector SDK for developing custom connectors. The SDK includes a Sample connector, which provides the framework for writing a connector to integrate with important applications in your enterprise.
- An Event connector that collects events from low-level event sources, transforms them into the CA SOI alert format, and displays them as infrastructure alerts in CA SOI associated with existing or created CIs.

NOTE
For more information about the connector architecture and how to build custom connectors, see Connectors Overview. Each connector also ships with a connector-specific *Connector Guide* that contains information about connector installation, configuration, how the connector interprets data from its domain manager, and whether it supports inbound operations.

CA Event Integration

CA SOI uses the CA Event Integration technology to enable integrations with low-level event sources through the Event connector. The Event connector provides integration with several raw event sources:

- Windows Event Log
- CA NSM agent messages from a CA NSM Event Manager or Agent
- Log files
- CA OPS/MVS EMA alarms and CA SYSVIEW PM alerts
- HP Business Availability Center alerts
- Web services events
- SNMP traps

The Event connector configures automatically when you install it. However, you can add or edit integrations with specific sources by launching the CA Event Integration administrative interface in context from CA SOI.

CA Catalyst Infrastructure

CA SOI fully adopts the CA Catalyst integration platform as its infrastructure. CA Catalyst is the CA Technologies common integration platform that provides the groundwork for unifying data from all CA Technologies products and many third-party products. CA Catalyst is fully embedded in the SA Manager installation. CA Catalyst provides the following functionality:

- A common semantic schema for data from all integrated products
- CI reconciliation to ensure that resources managed in multiple products have a unified set of property values
- CI synchronization that triggers bidirectional connector updates to source domain managers according to CI reconciliation and other operations
- The ability to enact specific use cases (including use cases that were available in previous releases of CA Catalyst), and the ability to manipulate the infrastructure to configure custom use cases, reconciliation formulas, and synchronization rules

Unified Service Model

The *Unified Service Model (USM)* is the semantic schema that is used as the CA Catalyst and CA SOI infrastructure.

Connectors transform all data that is collected from domain managers to the USM format before sending the data through CA Catalyst. The USM schema is stored in the CA Catalyst Registry.

USM is a high-level abstraction and generalization of IT management concepts that facilitate the semantic merging and interoperability of more specific domains. USM is developed to abstract and integrate information across many management products and domains. USM provides a single point for data federation, interoperability, and access to management data across an enterprise.

CA Catalyst provides the mechanisms to make all outbound from connector data adhere to the USM schema. CA SOI provides the interfaces to display the USM-compliant data.

Persistence Service

The Persistence Service enables other components to manipulate the Persistent Store. Operations such as reconciliation and synchronization require CA Catalyst components to modify the USM data. The Persistence Service enables interactions through a flexible interface that supports the following operations:

- Creating and storing USM data in response to incoming CIs from connectors
- Updating and deleting USM data in response to the reconciliation that the Logic Server performs
- Retrieving the USM details for display by the USM Web View
- Creating and updating USM data in response to operations initiated from the USM Web View

The Persistence Service provides an abstraction layer for working with data in the Persistent Store, which is the database record of USM data.

Logic Server

The CA Catalyst Logic Server provides the logic and the modules that carry out the following operations:

Reconciliation

Creates a unified set of properties and values from instances of a single entity that multiple connectors retrieve. The Logic Server reconciles CIs using formulas that define the property values to use. The policy rules that you can define include first non-null wins, majority wins, and data from a specific domain manager wins.

Synchronization

Detects the following CI changes within the Persistent Store:

- reconciliation
- CI creation in the USM Web View
- CI update or deletion in a source domain manager, or through some other method

Synchronization pushes the changes to applicable domain managers integrated through bidirectional connectors. Synchronization policies can create specific synchronization rules or use cases that keep source domain managers synchronized with the USM data.

The Logic Server lets CA Catalyst create and maintain a unified set of reconciled, correlated data in the Persistent Store. From the Operations Console, you can view the reconciled set of USM properties for any CI, named the reconciled sheet. You can also view the USM notebook for any CI. The notebook lists the reconciled sheet and the USM properties for each managed instance of the CI in source domain managers.

Registry

The CA Catalyst Registry is the repository for the USM schema and the policies that control the behavior of the Logic Server. You can access the Registry Administration UI from the CA SOI Administration UI to manipulate the Logic Server policies that control reconciliation and CI synchronization.

UCF Broker

The UCF Broker is a communication layer that controls access to the enabled bidirectional connectors, which can invoke inbound to connector operations on source domain managers. The Logic Server communicates synchronization changes to bidirectional connectors through the UCF Broker.

SA Store

The SA Store is the central repository for all CA SOI configuration and management data. It is a relational database from which the other CA SOI components retrieve their configuration policy and the read-write management data about the state of services and resources. The SA Store includes the following components:

- The CA Catalyst Persistent Store, which maintains a record of reconciled USM data (CIs, alerts, and relationships)
- Tables that contain data specific to CA SOI, such as escalation policies and service models

SA Manager

The SA Manager integrates the data that the connectors send:

- Correlates data so that CIs managed in multiple products are managed as one entity in CA SOI
  - Updates the Persistent Store with USM data
  - Provides correlation information to the Logic Server for reconciliation
- Manages CI and service status as follows:
  - Monitors the health and availability of managed CIs and services
  - Performs service impact and risk analysis
  - Monitors service-level agreements against defined thresholds
  - Updates the SA Store with analysis results and state changes
- Provides event and alert management functionality as follows:
  - Event policies to filter, correlate, and enrich events in the event store
  - Federated query of events across all integrated domain managers
  - Management of service impacting and all non-service impacting alerts
  - Alert queues to manage alerts by common properties
  - Escalation policies that can automate the response to alert occurrence, such as creating a help desk ticket, sending an email, and running a custom script or a CA Process Automation process

UI Server

The User Interface Server (UI Server) is the server that hosts the user interface applications. The UI Server is hosted within a web server, and you can deploy multiple UI Servers in a single CA SOI installation to support load balancing.

CA SOI has the following user interfaces:

- **Operations Console**
  Supports all administrative functions, including service modeling, defining alert queues and defining associated policy, and provides an operational view of the data for analysis purposes. Operators and other technicians use this interface to view and respond to alerts that report fault conditions. Administrators use this interface to define users and user groups, set role-based security, create and maintain service definitions, and more.
- **Dashboard**
  Displays service data that is tailored to the role of the user. Managers and others use this interface to analyze the overall health and availability of monitored services. They can also determine who is resolving problems and when those problems are fixed.
- **Mobile Dashboard**
  Provides content similar to the Dashboard in a format suitable for mobile devices. The Mobile Dashboard also lets you view service, customer, and alert queue details and take actions on alerts.
- **Report Console**
  Displays several types of scheduled and on-demand reports for service data in a portal-style interface. The reports provide service stakeholders with historical information that includes details about the availability and risk of a service.
- **Administration tab**
  Provides the tools to maintain connectors and SA Manager settings. The Administration tab also lets you configure single sign-on using CA EEM, email notifications, and other administrative functions.
- **USM Web View**
  Lets you browse and search all USM data in the Persistent Store. You can use the USM Web View to locate specific information, browse data based on many different criteria, and subscribe to RSS feeds to be notified of updates to specific CIs. This interface also lets you create new CIs and update existing CI information.
- **Debug Pages**
  The CA SOI Debug pages let you test and debug various CA SOI components. For more information, see the Debug Consoles.

CA EEM

CA Embedded Entitlements Manager (CA EEM) provides role-based authentication services for the CA SOI user interfaces and supports single sign-on across most interfaces. Single sign-on (SSO) requires all applications participating in SSO to use the same CA EEM server.

CA Business Intelligence

JasperReports Server is a third-party business intelligence platform that provides interactive reporting. CA Business Intelligence (CABI) is packaged with JasperReports Server. CA SOI uses CABI JasperReports Server to integrate, analyze, and present the information through various reporting options that are required for enterprise IT management.

Installation Prerequisites

Contents

Before you install CA SOI, perform the following tasks:

- Verify that your installation environment meets the requirements, such as hardware requirements, operating system support, database requirements, CA Technologies software support, and web browser support. See System Requirements.
- If you have a previous version of CA SOI installed, see Upgrades and Migration.
- Before installing any CA SOI component on a Windows server, verify that the server name satisfies these guidelines:
  - Conforms to the RFC specifications, particularly RFC-952 and RFC-1123.
  - Conforms to the Microsoft NetBiOS Computer naming conventions.
  - If the server is in the DNS, verify that the server name matches the NetBiOS name.
- On Windows 2008, disable Windows Firewall.
- Define a plan for the CA SOI deployment such as the installation location and how many systems you want CA SOI to monitor. For more information, see Installation Best Practices, Deployment Scenarios, and Plan the Implementation.
- Verify that all domain managers that you are going to monitor with CA SOI are installed and managing the infrastructure.
- (Optional) Install CA Business Intelligence JasperReports Server in a standalone system along with SOI Reports if you want to use CA SOI reporting. If you are upgrading to CA SOI 4.0 CU1 and using CABI BusinessObjects, you can continue using it.

  NOTE
  For more information about installing CA Business Intelligence, see Install Reporting Solutions.
- Install CA EEM if it is not already installed. Use the same CA EEM server for CA Catalyst and CA SOI installations. If CA EEM is already installed in the environment for use with other products, the CA Catalyst and CA SOI installations can leverage this installation. CA EEM requires that Java Runtime Environment (JRE) is installed first.
- Some connectors require you to complete tasks before you can install them. Before you install connectors, see the *Connector Guide* that is provided with each connector and verify that all prerequisites are met.
- Grant administrative privileges to the user who performs the installation.
- Ask the database administrator to create the SAMStore database and add a user with db_owner privileges on the database. For more information, see Database Considerations.
- Synchronize the time on all servers where CA SOI and CA Catalyst are installed. Configure the servers in the same Windows Server domain (so the domain controller synchronizes the time) or synchronize the servers over Network Time Protocol (NTP).

Database Considerations

Install and configure the database with the following parameters:

- For database version support, see Database Requirements.
- The database can be local to the SA Manager installation or on a remote server. As a best practice, install the database on the SA Manager system in smaller scale installations (less than 60,000 CIs). Install on a remote system in mid-scale to large-scale installations.
- Configure the database to use Mixed Mode authentication with the "sa" user name and password. You can then enter a SQL Server user name and password during CA SOI installation.
- As a best practice, the database administrator should create the SAMStore database before you install the product and create a user with db_owner privileges for the database. In this situation, the CA SOI installer requires only the user name defined for the database. Otherwise, the installer requires you to enter a database user with sysadmin privileges to create a new database.

  NOTE

  You must maintain a user with the db_owner role because the role can perform all configuration and maintenance activities on the database that CA SOI requires.
- Select the default instance or specify a named instance during the database installation. During CA SOI installation, specify a database instance and port with the option of using the default instance. The database connection behaves as follows with regards to the instance and port values:

  - If you change the port but not the default instance, the database tries to connect to that port using the instance name to find the port. For this lookup to work, the SQL Browser service must be running on the SQL Server node.
  - If you change the instance name but not the default port, the driver tries to resolve the instance name to the port supplied by the SQL Browser Service. If the resolution is successful, the database connects to that port.
  - If you change both default values, the database connects using only the port and the instance is ignored.
  - If you use a named instance and the SQL Browser service is not running or the database server is on a different network than the BusinessObjects server, BusinessObjects may have trouble connecting to the database when installing the CA SOI reports. To help ensure a successful connection in this situation, select Administrative Tools, Data Sources (ODBC) on the database server and add the instance name to the SAMStore DSN (for example, dbserver\myinstance).
- For collation settings during database installation, select SQL collations (used for compatibility with previous versions of SQL Server) and highlight Dictionary order, for use with 1252 Character Set.
- The database must be case-insensitive.
- The default name of the SA Store database is SAMStore, but you can change the database name if necessary.
- For more information about database disk space requirements, see Database Requirements.

Plan the Implementation

Before you implement CA SOI, consider the following questions:

- What services do you want to monitor?
- What event and alert sources do you want to monitor?
- What customers are dependent on those services and alert sources?
- What resources such as processes, software applications, and IT devices support those services and event and alert sources?
- How CA Spectrum domain managers are managing the service and alert resources?
- Approximately how many CIs and events does each domain manager manage?
- Who requires access to information about a modeled service or alert queue, and how should they be presented with that information?
- Who is notified if a service is impacted in some way?
- What processing operations on event sources are required to enable a quality alert set?
- What actions should occur automatically when an alert condition that impacts a service is identified?
- Does your enterprise contain logical tiers that require a multi-tiered CA SOI deployment, or can the enterprise be managed in one deployment with multiple services and alert queues?

The answers to these questions help you determine the following items:

- How to install CA SOI
- The connectors to implement
- The necessary user access permissions
- Details about the services to model
- The event policies and alert queues you require

Installation Best Practices

CA SOI has a flexible installation model with many different options for where to install components. The following list represents the recommended best practices for component installation:

- **CA EEM**

If you are using CA EEM with multiple products and CA SOI, install CA EEM on a separate server for best performance. If you are using CA EEM for only CA SOI, it can exist on the same system as the SA Manager without affecting performance.

- **SA Manager**
  Install the SA Manager component on one dedicated system. This component includes the CA Catalyst Logic Server and Registry which you must reference when installing other components on separate servers. You can install the SA Store database on this server or point to a remote database. If your enterprise requires a tiered implementation, install multiple SA Managers with CA SOI Domain connectors forwarding information to an enterprise SA Manager.
- **SA Store Database**
  Due to resource requirements, we recommend installing the SA Store Database on a dedicated server. Review the following scenarios when determining the best database location:
  – A remote database installation may be necessary depending on your Microsoft SQL Server configuration (for example, if Microsoft SQL Server exists on a SQL cluster).
  – Install the database on the SA Manager system for smaller scale installations (less than 60,000 CIs) to avoid network latencies for database communications.
  – Install the database on a remote system for mid or large-scale installations to avoid processing conflicts.
- **UI Server**
  Install the UI Server on a separate dedicated system. You can install multiple UI Servers on multiple systems, but each UI Server can connect to only one SA Manager.
- **CABI JasperReports Server and CA SOI Reports**
  Due to resource requirements, install CABI JasperReports Server and the CA SOI reports on a dedicated server with no other CA SOI components. See the system allocation in Install Reporting Solution.
- **Connectors**
  Install connectors based on connector requirements and system availability. We recommend distributing connector installations across multiple dedicated systems to distribute the processing work load. Most connectors can be installed remotely, but some require installation on the same system as the domain manager. Many connectors support running multiple instances of the same connector on the same system.
  Multiple connector installations are supported on one system. If you plan to install connectors on a system with other CA SOI components or multiple connectors on the same system, see the hardware requirements in the Release Notes to ensure that the system has the appropriate resources to support the installations.
  If you anticipate receiving more than 100,000 alerts daily, enable 64-bit on the IFW server. For more information about enabling 64-bit, see How to Install CA Catalyst Connectors (Pre-r3.2).
  As a best practice, the total number of CIs managed by connectors on one system should not exceed 200,000 to avoid memory problems. For example, if you have four CA eHealth installations that manage a total of 300,000 CIs, you should distribute the four remote CA eHealth connectors across at least two systems. Also consider the number of events managed by each connector if your overall deployment represents a significant event stream. In this case, try to distribute the event stream evenly across remote connector systems.
  For more information about specific connector installation support, see Connectors Overview.
- **Domain connectors**
  Install CA SOI Domain connectors in a tiered SA Manager deployment on the enterprise SA Manager, on a system with other connectors reporting to the enterprise SA Manager, or on a remote system with no CA SOI components. Do not install CA SOI Domain connectors on a system with any component of the source CA SOI deployment from which it is collecting data.

For an illustration of this best practice installation scenario, see Typical Multi-server Deployment.

Common Deployment Scenarios

Contents

You can deploy CA SOI in several different configurations. A typical deployment of CA SOI contains the following components:

- CA EEM
- MQ Server
- SA Manager (includes the CA SOI connector and Universal connector)
- UI Server
- SA Store database
- CA Business Intelligence

Consider the following deployment options:

- System quantity on which to install components
- Connectors to install
- Where to install the connectors, which can be local or remote to the domain manager depending on the connector
- SA Manager and UI Server components quantity, which depends on tiering and processing requirements

The following deployment restrictions apply for component installation:

- The UI Server can connect to one SA Manager only. However, multiple UI Servers can connect to the same SA Manager.
- The SA Store database and the CA EEM application instance have a 1:1 relationship.
- Due to resource requirements, we recommend installing the SA Store database and CA Business Intelligence on dedicated servers.

Typical Multi-Server Deployment

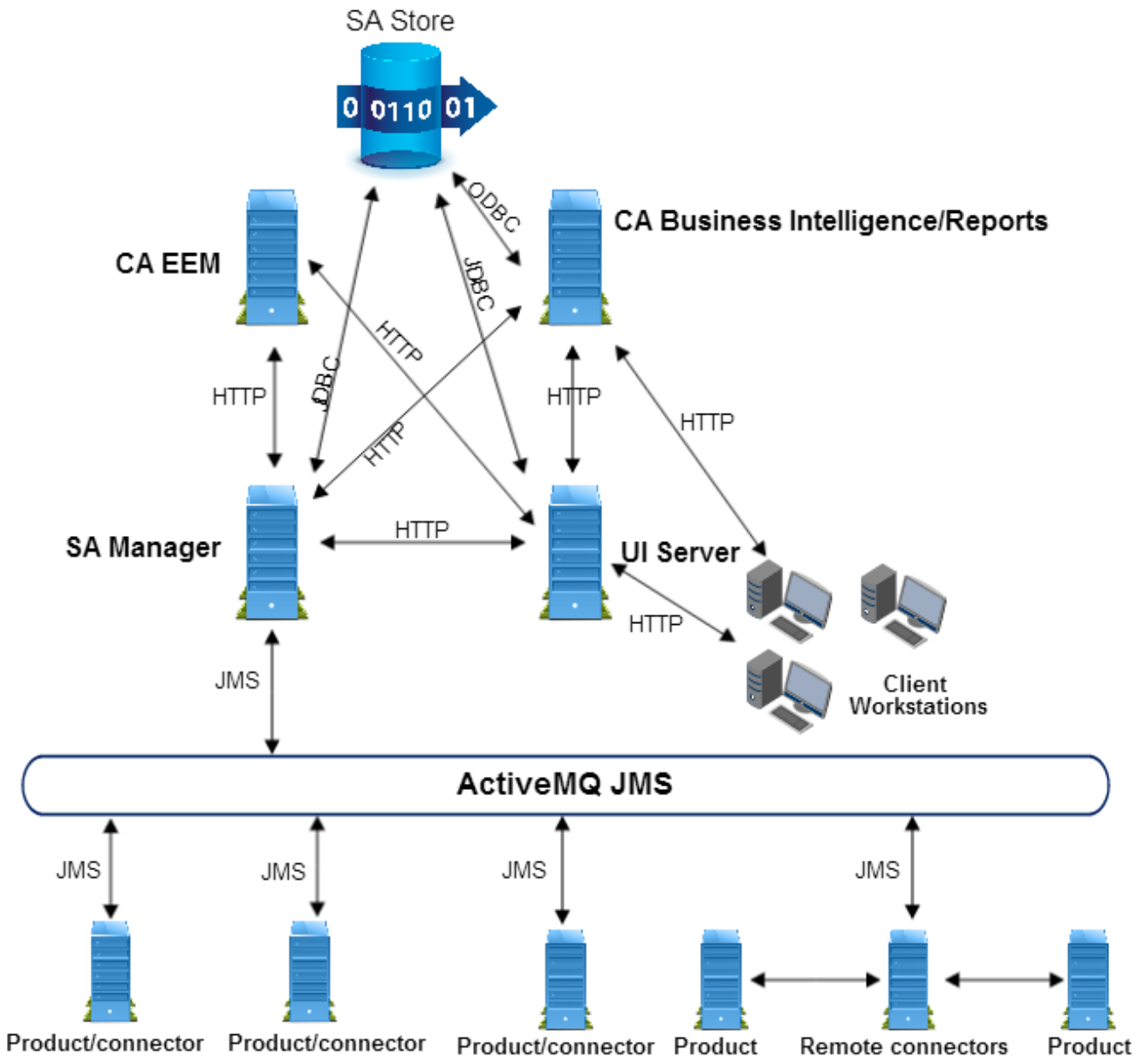
A typical CA SOI deployment occurs across multiple servers. Each server contains specific components, and the servers communicate using various protocols to provide all product functionality. A typical, multi-server deployment consists of the following components on the following servers:

- Server 1: CA Business Intelligence, CA SOI Reports
- Server 2: CA EEM
- Server 3: SA Manager, Universal connector, Mid-tier connector, Service Discovery, CA EEM
- Server 4: UI Server
- The Microsoft SQL Server database is installed on a database server that can be local or remote from the SA Manager or part of an existing SQL cluster.
- Extra connector servers may be required to optimize performance or if you install local connectors that require their domain manager on the same server.

The following graphic shows a typical multi-server deployment and how all components communicate:

> **NOTE**
> Although the list groups CA EEM with Server 2 or 3, the graphic depicts it separately to show how other components communicate with it. You can install CA EEM on a separate server if products other than CA SOI uses it.

Figure 1: multi-server deployment



Small Scale or Proof of Concept Deployment

You can deploy most CA SOI components on a single server. This deployment functions the same as a multi-server deployment, except that the components communicate on the same server. A small scale or proof of concept deployment includes the following servers:

- Server 1: CA Business Intelligence/Reports
- Server 2: SA Manager, UI Server, connectors

The components that you cannot install on a single server are as follows:

- Connectors that support only installations on the same server as their domain manager
- CA Business Intelligence, which must always be installed on a dedicated server

Most typical CA SOI environments cannot function optimally with this deployment. This configuration should be an option only in small scale or proof of concept deployments. Verify that the server with the most CA SOI components meets the requirements for a CA SOI stand-alone installation in Hardware Requirements.

High-Performance Deployment

For environments with many CIs, services, alerts, and connectors, consider a high performance deployment to optimize product functionality. Install CA SOI on a 64-bit operating system to use a 64-bit JRE for enhanced performance. A high performance deployment includes the following servers:

- Server 1: CA Business Intelligence, Reports
- Server 2: CA EEM
- Server 3: SA Manager (64-bit)
- Server 4: UI Server (64-bit)
- Server 5: Remote database server hosting the SA Store
- Server 6: Connector server, which can be multiple servers depending on the number of connectors.
- Server 7: MQ Server

64-Bit Implementation

When you install CA SOI components on a 64-bit operating system, they automatically install as 64-bit applications configured to use a 64-bit JRE. Any deployment type can take advantage of the performance benefits gained by installing with a 64-bit JRE as long as you install on 64-bit operating systems. The following components are installed as 64-bit applications on 64-bit operating systems:

- SA Manager
- UI Server

> **NOTE**
> Because some connectors do not work with a 64-bit IFW, the IFW does not install as a 64-bit application by default. However, you can change the IFW to 64-bit mode.

Tiered SA Manager Deployment

You can deploy a distributed, tiered SA Manager environment for the following conditions:

- You want to structure your enterprise by regions, organizations, or departments.
- You want to provide a multi-tenant environment with physical separation of resources.

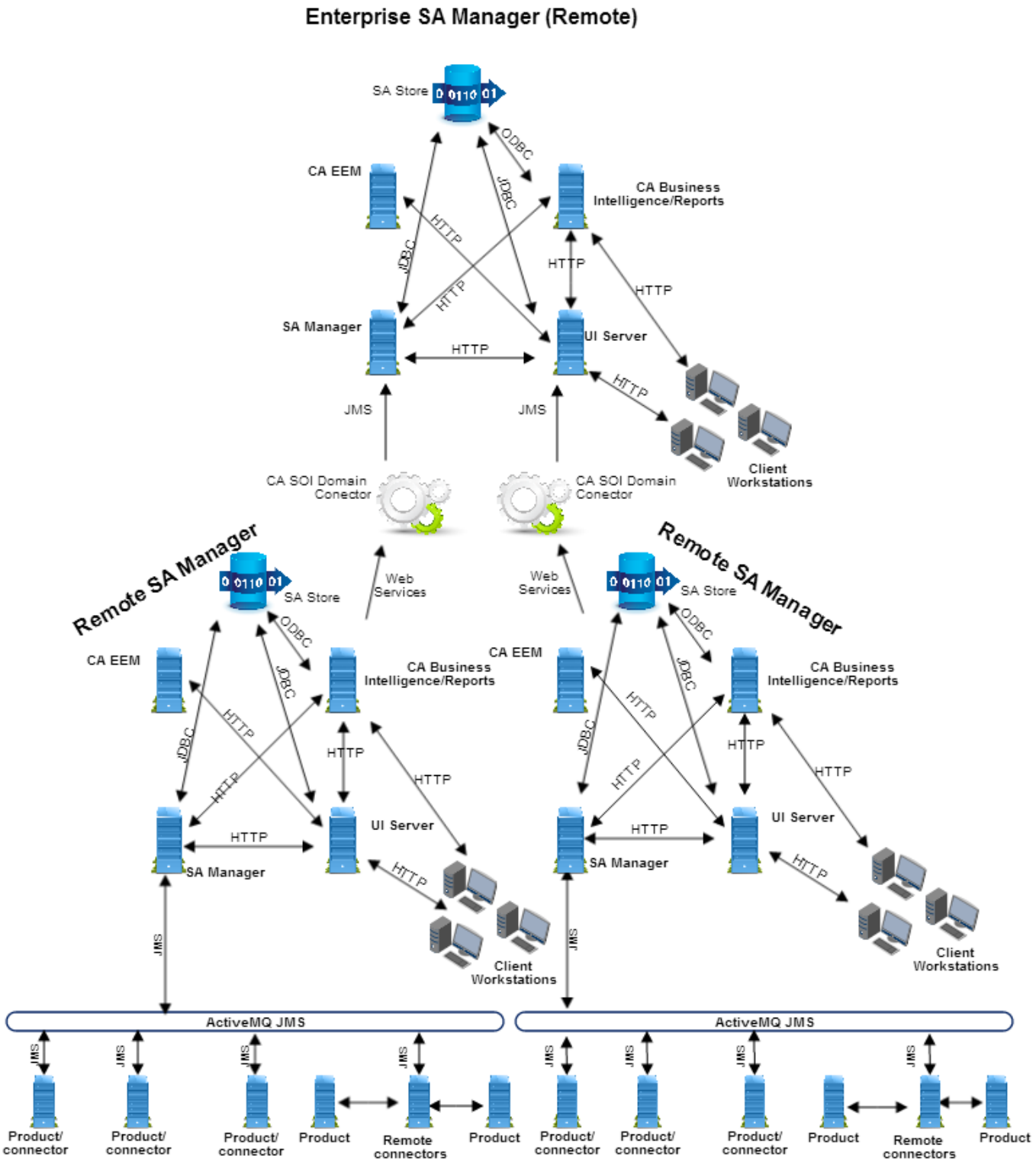
In a tiered SA Manager deployment, CA SOI Domain connectors collect services, service-impacting alerts, and customer information associated with services from source SA Managers (local). The CA SOI Domain connectors forward this information to an Enterprise SA Manager (remote) for analysis with other source SA Manager data.

A typical distributed SA Manager deployment consists of the following components:

- One Enterprise SA Manager with all other components of a typical CA SOI deployment. Connectors typically do not report directly into the Enterprise SA Manager.
- Multiple full CA SOI deployments, including connectors, that represent a logical area of your enterprise (region, location, data category, or other)
- CA SOI Domain connectors that integrate data from each local source SA Manager into the remote Enterprise SA Manager

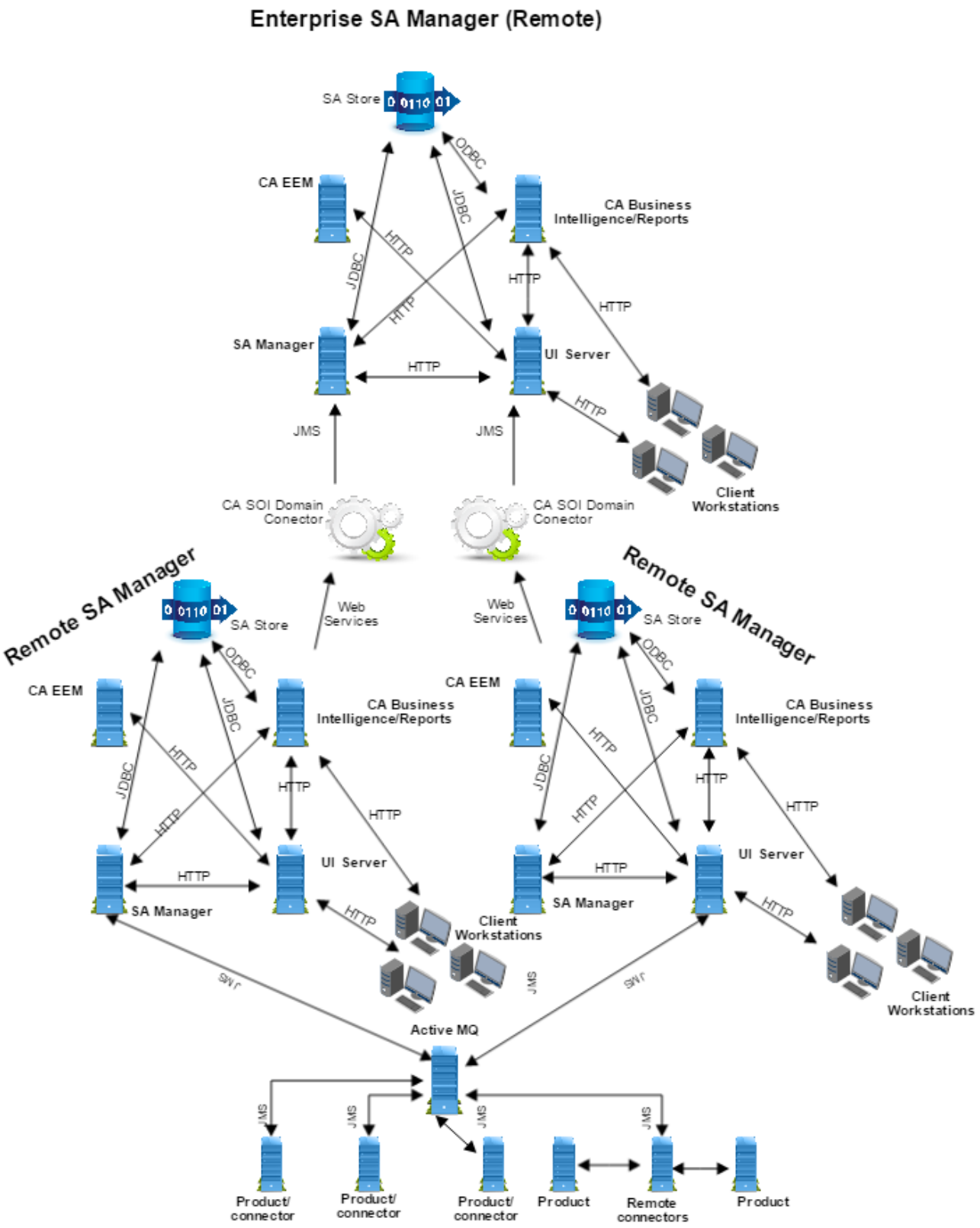
The following graphic shows a deployment with two full SA Manager deployments reporting to an Enterprise SA Manager through CA SOI Domain connectors:

Figure 2: enterprise sa manager



The following graphic shows a deployment with one MQ Server shared with two full SA Manager deployments, and reporting to an Enterprise SA Manager through CA SOI Domain connectors:

Figure 3: Single Avtice MQ



Specialized Deployment Scenarios

Contents

This section describes several supported specialized CA SOI deployments.

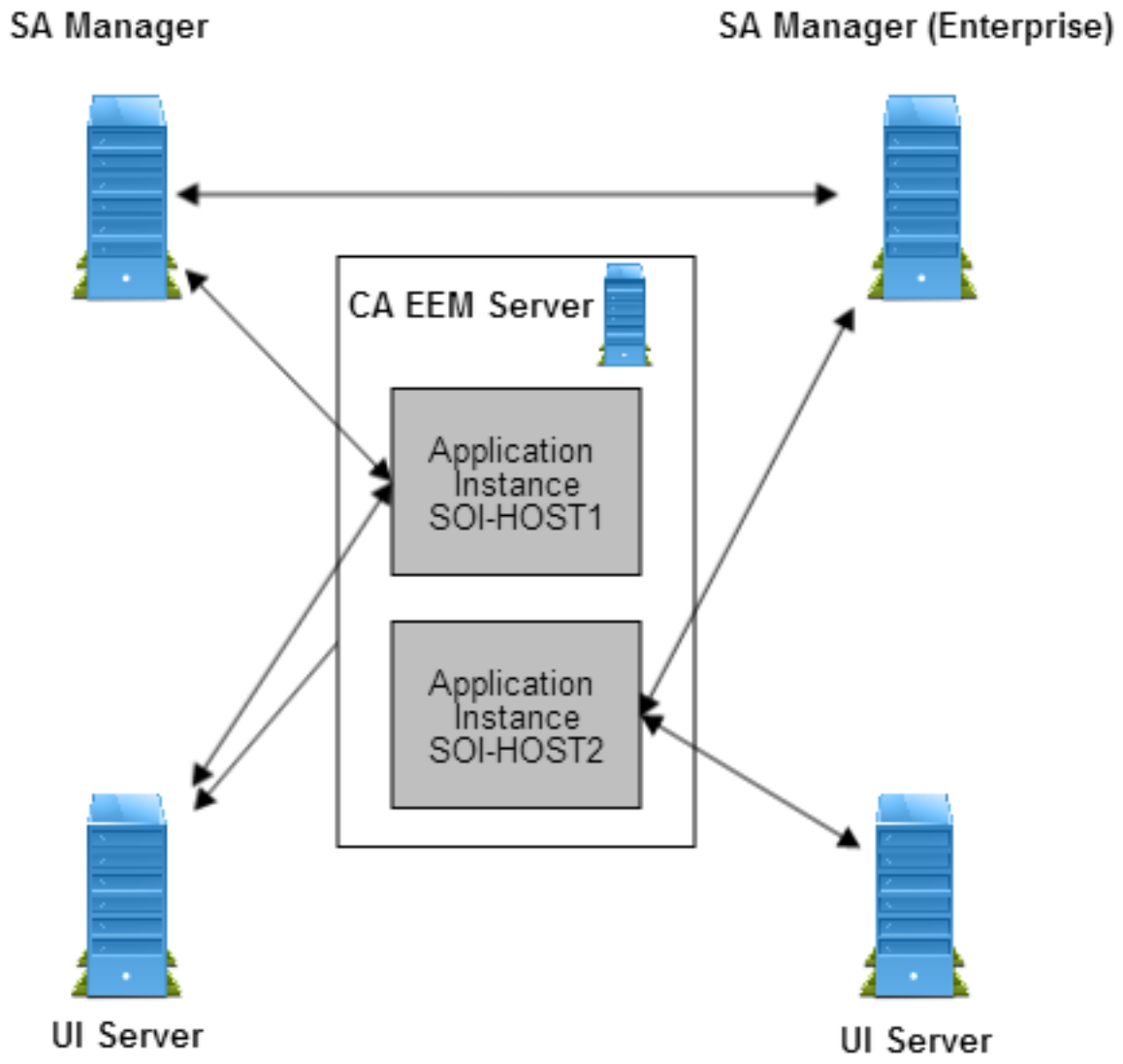
One CA EEM Server and Multiple SA Managers

You can connect CA SOI with CA EEM in several different ways. However, installing CA EEM on a separate server provides the most flexibility, especially if you want to use CA EEM to provide user authentication for other CA Technologies products.

All CA SOI components that are a part of a single CA SOI deployment must share the same CA EEM server and application instance. Also, SA Managers and UI Servers can point to the same CA EEM server. When using separate application instances for each CA SOI instance, a tiered SA Manager deployment can use one CA EEM server.

The following graphic shows two UI Servers and SA Managers using different application instances on the same CA EEM server:

Figure 4: ca eem

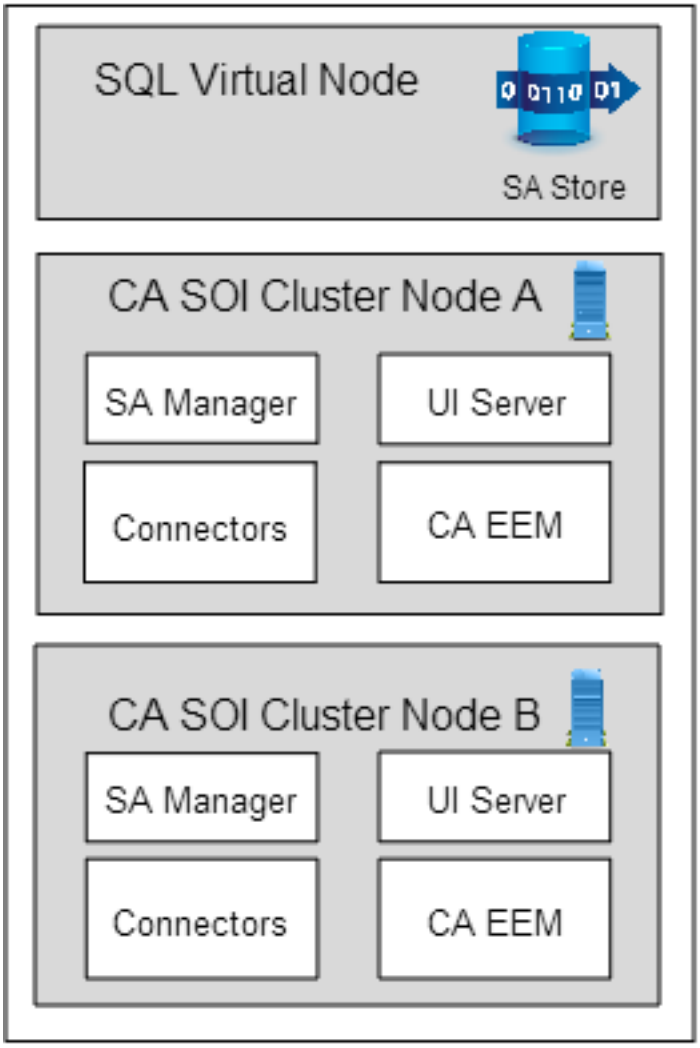


High Availability Deployment

CA SOI supports a deployment in a Microsoft Cluster Server environment for sites that require high availability. A high availability implementation ensures zero service management downtime in the event of a system failure or when a system or product maintenance is required.

The following illustration shows the architecture of a common high availability implementation:

Figure 5: ca eem 2



NOTE
For more information about other high availability architectures and how to implement CA SOI in a Microsoft Cluster Server environment, see Implementation in a Microsoft Cluster Server Environment.

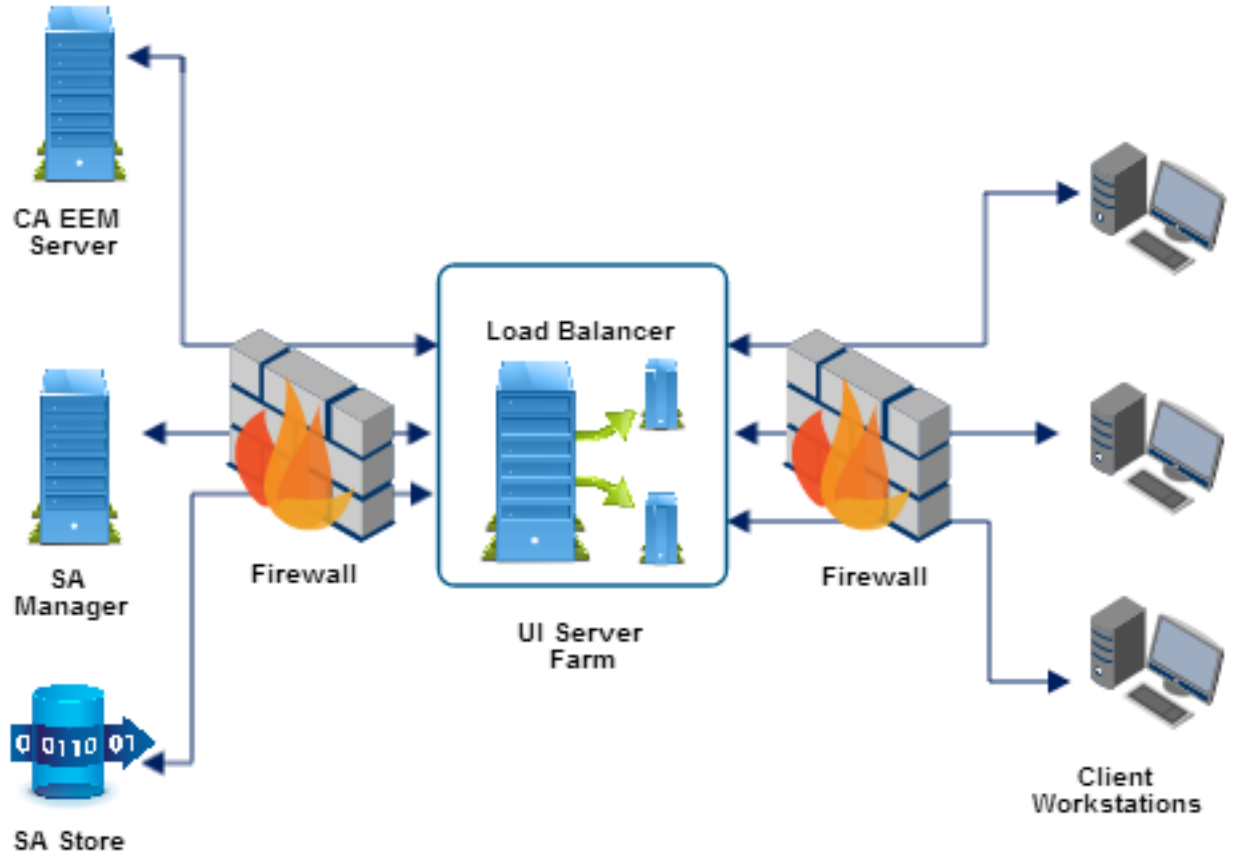
Multiple UI Servers Behind Load Balancer

You can install multiple UI Servers reporting to the same SA Manager server that operate behind a load balancer to improve performance. This deployment typically consists of the following components:

- The SA Manager is installed on one server.
- CA EEM is installed on a separate server.
- The SA Manager connects to a UI Server farm behind the load balancer.
- The load balancer must support server affinity.

The following graphic shows a multiple UI Server deployment:

Figure 6: servers behind load balancer



This graphic also depicts the load balancer in a dual-firewall environment. For more information, see Firewall Environment Deployments.

Firewall Environment Deployments

When installing and running CA SOI in environments with firewalls, verify that communication among components on different servers occurs without blockage. The following sections are common firewall scenarios and the default ports that must be open to allow the product to function:

> **NOTE**
> All ports in this section are the default selections that are provided during installation.

Firewall between SA Manager and Connectors

Connectors may be deployed on domain managers that are in different security domains. In this case, you must open port 61616 for outbound communication between connectors and the MQ Server and port 8020 for inbound communication between connectors and the UCF Broker, which invokes inbound to connector operations on the domain manager. The UCF Broker port is only required if synchronization operations are enabled, for which only certain use cases are supported.

When there are firewalls between the SA Manager and connectors, the best practice is to install the connectors on the domain managers to minimize the number of open ports and keep them consistent. Installing connectors on the SA Manager creates different port requirements for communicating with each domain manager operating behind a firewall.

UI Server in a DMZ environment

You can deploy UI servers outside of firewalls (in DMZs) to protect the internal network while allowing availability of certain services to external clients. See the graphic in Multiple UI Servers Behind Load Balancer for an illustration of this scenario. In this scenario, the following ports must be open for communication between the UI Server and CA SOI components:

> **NOTE**
> A value of RP (random port) in the first Port column designates a unidirectional connection. A unidirectional connection can use any port from the server of the source component to connect to the designated port on the server of the destination component.

Source	Port	Destination	Port	Protocol
UI Server	RP	SA Manager	7090	HTTP
UI Server	RP	SA Manager	7493	HTTPS
UI Server	RP	SA Store	1433	JDBC
SA Manager	RP	UI Server	7070	HTTP
SA Manager	RP	UI Server	7403	HTTPS
Client workstations	RP	UI Server	7070	HTTP
Client workstations	RP	UI Server	7403	HTTPS
UI Server	RP	CA EEM	5250	HTTP
UI Server	RP	BusinessObjects	1433	ODBC
UI Server	RP	Connectors	61616	JMS

In a dual-firewall environment, open port 7070 for inbound and outbound communication between external clients and the UI Server, as described in the table. Also open port 7090 for external access to the Administration UI, which communicates with the SA Manager.

Mobile Dashboard in a DMZ environment

You can expose only the Mobile Dashboard for client access from the Internet, to ease the port requirements in the firewall that separates the DMZ from general Internet access. Perform a standalone deployment of the Mobile Dashboard on a server inside the DMZ. Open port 7070 and 7403 across both firewalls for interface access and port 7090 on the firewall that separates the DMZ from the SA Manager and UI Server. For more information about setting up this environment, see Deploy the Mobile Dashboard on a Standalone Server.

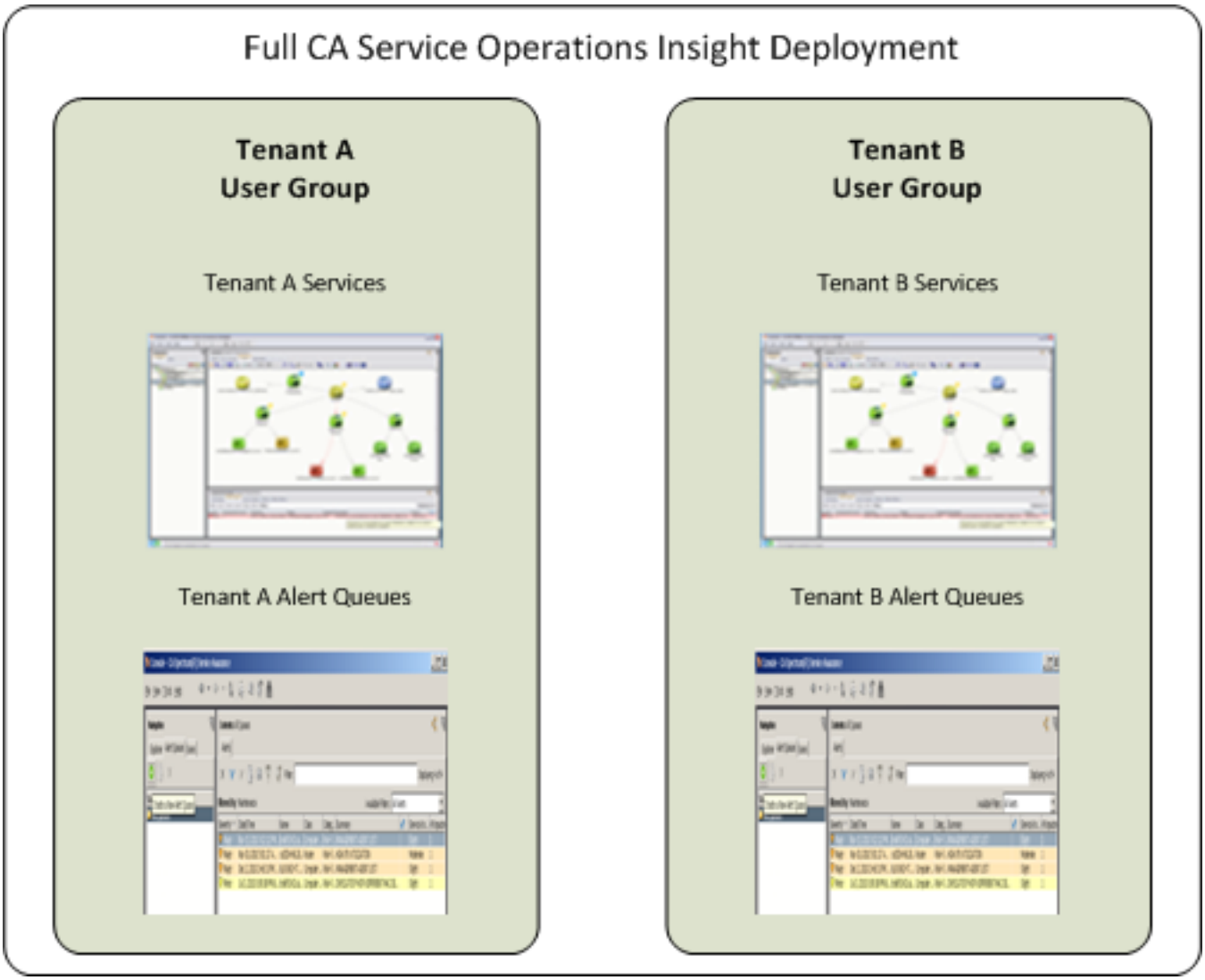
CA SOI and Multi-tenancy

CA SOI does not fully support a multi-tenant architecture using one installation. However, you can deploy the product in a way that supports multiple tenants through user groups or a tiered deployment.

You configure the product for use by multiple tenants with user groups as follows:

- Perform one typical CA SOI deployment.
- Create a separate user group for each tenant, and provide each group access to only the appropriate services and alert queues.
- Change the logo on the Dashboard for each user group if each tenant has a distinct company logo.

The following graphic shows how tenant resources are separated through user groups:



Each user group only has access to its services and alert queues. In this scenario, users in the Administrators user group can access all resources.

> **NOTE**
> For more information about creating user groups and assigning access privileges for services and alert queues, see How to Create and Manage Alert Queues.

Another option for supporting multiple tenants is a tiered SA Manager deployment, which creates a physical separation of resources across SA Managers. In this scenario, each tenant's resources resides on a separate lower-tier SA Manager. The Enterprise SA Manager provides a consolidated view spanning multiple tenants to support unified operations by an administrator.

Multi-NIC Server Deployment

You can install CA SOI on a server with multiple network interface cards if, for example, multiple interfaces are required for high availability or access control. If you are installing on a multiple-NIC server, define the correct host name in the appropriate DNS server for CA SOI to bind to the intended interface and install successfully.

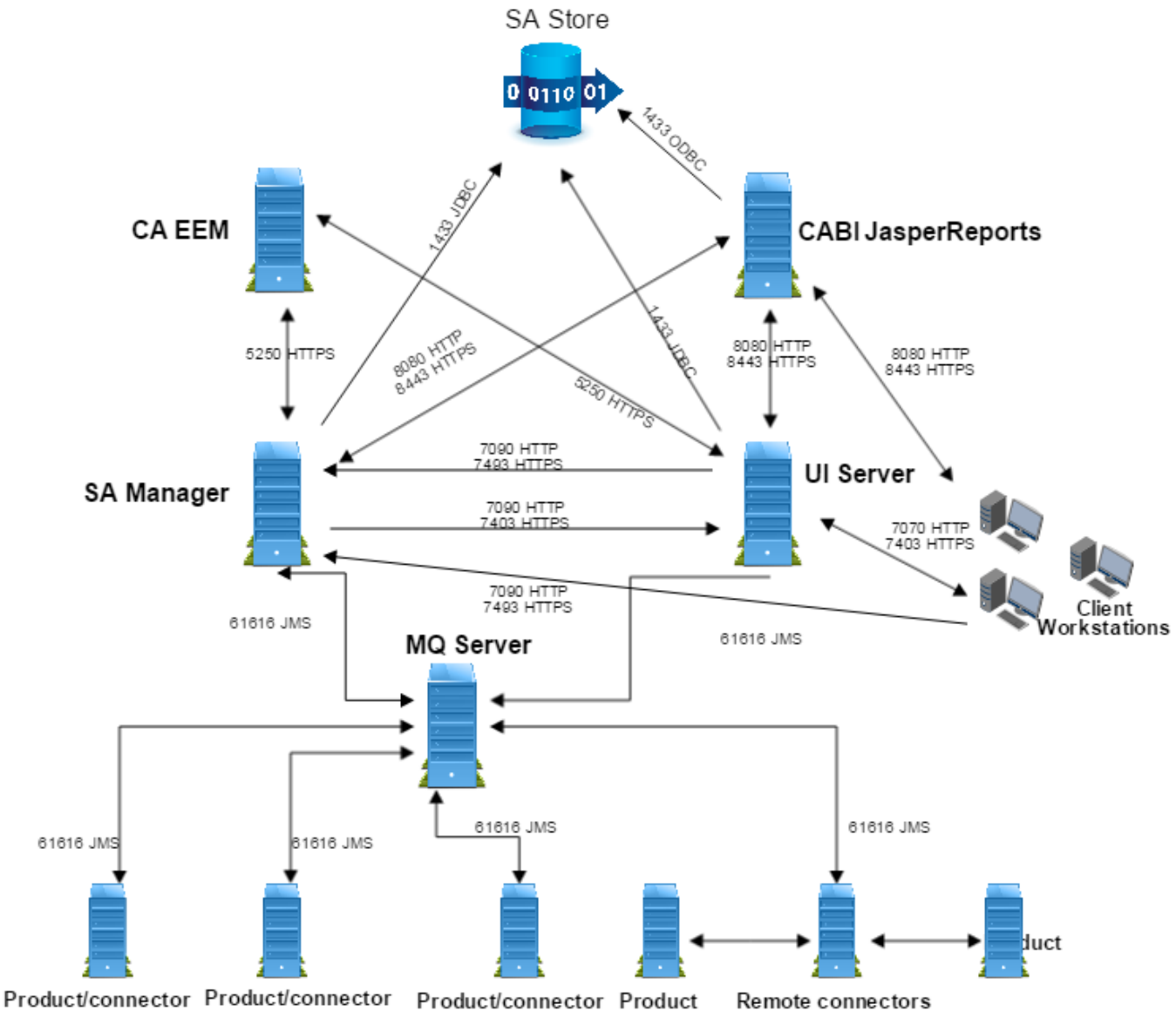
If the product still has trouble binding to the appropriate interface, you can configure an explicit binding to the interface by changing host property to the correct host name in the <SOI_HOME>\tomcat\lib\jmsconnect.properties file.

Communication Ports

CA SOI requires several ports open between its components. You can change the default CA SOI and CA Catalyst ports during installation.You can change some ports after installation if necessary. For more information, see Communication Port Maintenance.

The default ports that are required for communication between CA SOI components are pictured in the following graphic and listed in the following tables:

Figure 7: communication ports



Ports Pictured:

NOTE
A value of RP (random port) in the first Port column designates a unidirectional connection. A unidirectional connection can use any port from the server of the source component to connect to the designated port on the destination component server.

Source	Port	Destination	Port	Protocol
SA Manager	RP	UI Server	7070	HTTP
SA Manager	RP	UI Server	7403	HTTPS
UI Server	RP	SA Manager	7090	HTTP
UI Server	RP	SA Manager	7493	HTTPS
UI Server	RP	Connectors	61616	JMS
SA Manager/UI Server	RP	SA Store	1433	JDBC
Connectors	RP	UCF Broker (on SA Manager)	8020	HTTP
UCF Broker (on SA Manager)	RP	Connectors	7878	HTTP
Connectors	RP	SA Manager/MQ Server	61616	JMS
SA Manager/UI Server	RP	CA EEM	5250	HTTPS
Client workstations	RP	UI Server	7070	HTTP
Client workstations	RP	UI Server	7403	HTTPS
Client workstations	RP	SA Manager	7090	HTTP
Client workstations	RP	SA Manager	7493	HTTPS
Client workstations	RP	BusinessObjects	8080	HTTP
Client workstations	RP	BusinessObjects	8443	HTTPS

Ports Not Pictured:

CABI JasperReports Server

- JasperReports Server Protocol: HTTPS/HTTP
- JasperReports Server Port: 8080

SA Manager

- Tomcat Shutdown Port: 7095
- Integration with CA Process Automation for alert escalation or BMC Remedy integration: 8080
- Integration with help desk products: 8080
- SMTP server port for email configuration (also on UI Server): 25

UI Server

- Tomcat Shutdown Port: 7075
- MQ port for connector communication: 61616

Connectors

- UCF Broker ports: 8030, 8040

Client workstations

- Communication to SA Manager: 7090 (HTTP), 7493 (HTTPS)

Active Directory connections (optional)

- CA EEM: 389

  NOTE
  For more information about configuring CA EEM to connect with an Active Directory, see the CA EEM documentation.

Installation

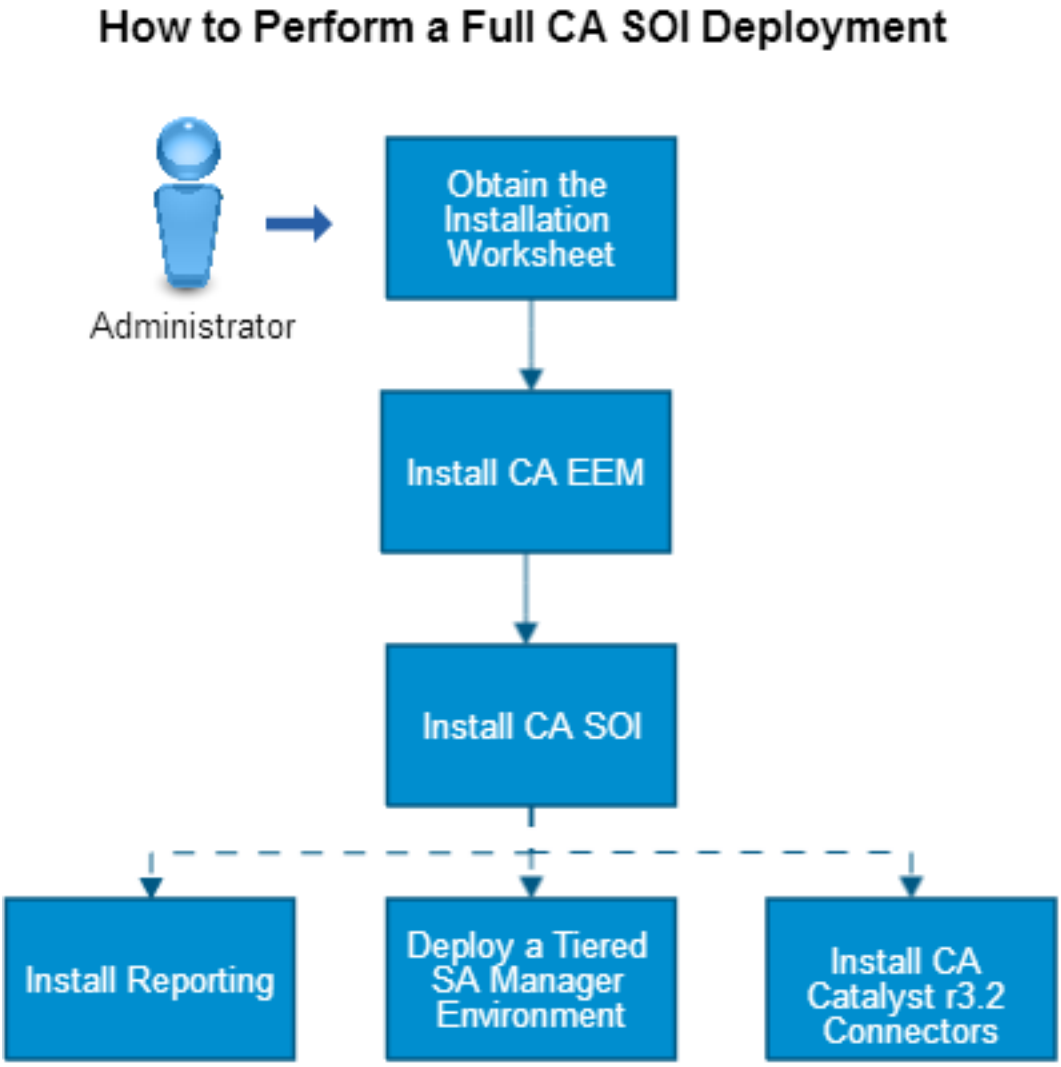
This section describes how to perform an end-to-end installation of CA SOI.

How to Perform a Full CA SOI Deployment

As an administrator, you perform a full CA SOI installation, which consists of prerequisites and core CA SOI components installed in a specific order: CA EEM, CA SOI, reporting (BusinessObjects), and the CA Catalyst connectors. You can also deploy a tiered SA Manager configuration.

Use the following scenario to guide you through the process:

Figure 8: how to perform full soi deployment



Perform a full CA SOI deployment by installing all components in the following order:

1. Obtain the installation worksheet.
2. Install CA EEM (if not already installed).
3. Perform the CA SOI installation.
4. Install connectors.
5. (Optional) Install CA Business Intelligence for reporting.
6. (Optional) Deploy a tiered SA Manager environment.
7. Perform post-installation configuration and customization.

Obtain the Installation Worksheet

The Installation Worksheet is a Microsoft Excel spreadsheet that lets you note particular server names, ports, and credentials as you install each component. During some component installations and configurations, you provide specific data from other components. The components are listed in the installation order. For security reasons, be careful if you decide to include passwords on the Installation Worksheet.

installation_worksheet.xlsx

Install CA EEM

Consider the following items before installing CA EEM:

- Install CA EEM before installing the CA SOI components. For CA EEM version support, see Software Requirements.
- CA EEM requires a JRE to be present before installation.
- Install one CA EEM server only. For detailed installation instructions for installing the latest supported version, see the *CA EEM Getting Started Guide*.
- You can install the CA EEM version that CA SOI provides. Alternatively, you can install a newer version or you can leverage an existing version that CA SOI supports.
- The CA EEM server can exist on the system where you plan to install the SA Manager system or on a separate system.
- If you plan on using CA EEM to integrate with an LDAP server, create the administrator user that you reference during the CA SOI installation in LDAP.
- Complete the CA EEM and JRE sections on your Installation Worksheet.

How to Perform a CA SOI Installation

Contents

As an administrator, you install CA SOI using the main CA SOI installation wizard, which lets you install all core CA SOI components except for reports, which you can install later. The single installer is used to install all the core components [SA Manager, MQ Server, UI Server, IFW, and Universal connector client] of CA SOI. When all components are installed in a single system and if you upgraded one of the core components, you cannot upgrade the other components later.

> **NOTE**
> If you are installing all CA SOI components on a single server, you can select all components during the installation. You do not have to rerun the installer for each CA SOI component.

1. Verify the installation prerequisites.
2. Install MQ Server
3. Install the SA Manager.
4. Install the UI Server.
5. Install the CA SOI Console.
6. Install the Universal Connector Client.
7. Install the Integration Framework (IFW).
8. Verify component services and connections. If necessary, review the troubleshooting information.
9. Continue with other installations as described in How to Perform a Full CA SOI Deployment:
   - Install the reporting component.
   - Deploy a tiered SA Manager environment.
   - Install CA Catalyst r3.2 connectors.

Verify Installation Prerequisites

Review the following items before you install CA SOI:

- Review the topics in Installation Planning for information about prerequisites, best practices, and deployment scenarios.
- Review Release Information?for the operating system, hardware, and software support information.

Consider the following installation items:

- Have your Installation Worksheet available. The installer prompts you to either enter values on the worksheet or to use values on the worksheet to complete the installation screens.
- If CA SOI components are already installed on the system, the Choose Install Folder screen does not appear. Because all CA SOI components must reside in the same folder, any new CA SOI components are installed to the existing CA SOI folder.
- (Optional) Review the installation log file (CA_Service_Operations_Insight_Install_*releasenumber*) that is in the SOI_HOME\log folder to check for installation errors. This folder also provides the installation log files for the other installed components.

Install MQ Server

The MQ Server is the primary component in CA SOI. Install it before any other CA SOI components.

NOTE

- All connectors are running when MQ Server is Up and Running. When MQ Server is restarted, connectors re-connect to the MQ Server. For the synchronization to work, ensure that CA SOI Application Server is restarted when MQ Server is restarted. However, when CA SOI Application Server is down, MQ Server remains in running state. As the MQ Server comes up, the connectors are Online and start processing the data.
- The connectors are up and running and receive alerts from MDR but the alerts do not appear on the Operation Console when CA SOI Application Server is down. The updates appear on the Operation Console when CA SOI Application Server is up and running.
- CA SOI fails to receive the universal connector alerts when the alerts are sent while the CA SOI Application Server is down. Ensure that the alerts are sent only when the Application Server is up and running for Universal Connector.
- The **Current Integration Framework Status** appears as Closing on the SOI Administration Dashboard. This occurs when you stop the Container (Catalyst and IFW) service while MQ Server is up and runnin

Follow these steps:

1. Navigate to **Disk1\SOI** folder of the CA SOI installation image, right-click **soi-installer.exe**, and click **Run as administrator**.
2. Click **Next** in the Introduction page, accept the license agreement, then accept the third-party license agreements, and click **Next**.

   NOTE
   If CA SOI components are already installed on the system, the Choose Install Folder page does not appear. Any new CA SOI components that you install are installed to the existing CA SOI directory because all components must reside in the same directory.
3. Enter or choose the installation folder, and click **Next**.

   NOTE
   The maximum installation path length is 150 characters. The installation blocks paths with more than 150 characters.
4. Select **Custom** in Install Set, click **MQ Server** and then click **Next**.

5. Enter the CA SOI administrator credentials, and click **Next**.
6. The **TCP Port** number and the MQ hostname is populated by default on the **ActiveMQ Server Configuration** screen, click **Next**.
7. Select **Start Services** and click **Next**.
8. Click **Install** in the Pre-Installation Summary page.
9. Click **Done** when the installation completes.
10. To enable the MQ log, navigate to **SOI_Home\apache-activemq\conf** folder and update **log4j.properties** file as follows:
    a. Uncomment **log4j.rootLogger=INFO, console,logfile.**
    b. Comment **log4j.rootLogger=OFF**.
    For example:
    ```
    log4j.rootLogger=INFO, console, logfile
    #log4j.rootLogger=OFF
    ```
11. (Optional) Review the installation log file (CA_Service_Operations_Insight_Install_*releasenumber*) that is in the **SOI_HOME\log** folder to check for installation errors. This folder also provides the installation log files for the other installed components.

The MQ Server is installed in your system.

Install the SA Manager

The SA Manager is the primary management.

The following components are installed automatically with the SA Manager:

- IFW with Event Management
- Mid-Tier Connector
- Service Discovery
- Universal Connector

Follow these steps:

1. Navigate to **Disk1\SOI** folder of the CA SOI installation image, right-click **soi-installer.exe**, and click **Run as administrator**.
2. Click **Next** on the Introduction page.
3. Accept the license agreement, then accept the third-party license agreements, and click **Next**.
4. Accept, enter, or select the installation folder.

   NOTE
   If CA SOI components are already installed on the system, the Choose Install Folder page does not appear. Any new CA SOI components that you install are installed to the existing CA SOI directory because all components must reside in the same directory.
5. Select **Custom** in Install Set, click Manager, and then click **Next**.

   NOTE
   The Universal Connector Client plug-in client utility is installed with the SA Manager regardless of whether you make this selection. You can also install the client on a separate system, if necessary.
6. Click **LGPL Distribution Directory**, select the **disk 2 folder (\SOI x.y\disk2\lgpl)** available in the installation kit, and click **Next**.
7. Enter the CA SOI administrator credentials, and click **Next**.
8. Enter the **MQ Server Host** name and the **TCP Port** number is already populated. Click **Next**.
9. Accept the default values of Manager configurations, and click **Next**.
10. Accept the default values of Integration Services, and click **Next**.

11. Enter the database hostname and credentials, and click **Next**.
12. Enter the EEM Server name and password, and click **Next**.
13. Select **Start Services** and click **Next**.
14. Click **Install** on the Pre-Installation Summary page.
15. Click **Done** when the installation completes.

The SA Manager installation is completed.

Install the UI Server

The UI Server hosts the CA SOI user interfaces and establishes the interface communication with the SA Manager.

- This procedure assumes that you are installing the UI Server on a separate system and enter the information to connect to the installed SA Manager. The best practice is to install the UI Server on a separate server. However, you can install the UI Server on the same server as the SA Manager in a combined installation. If you are installing the components separately, install the SA Manager first.

Follow these steps:

1. Run **soi-installer.exe** from the **Disk1\SOI** folder of the CA SOI installation image.
2. Accept the license agreement, then accept the third-party license agreements, and click **Next**.

   NOTE

   If CA SOI components are already installed on the system, the Choose Install Folder page does not appear. Any new CA SOI components that you install are installed to the existing CA SOI directory because all components must reside in the same directory.
3. Accept, enter, or select the installation folder.

   NOTE

   The maximum installation path length is 150 characters. The installation blocks paths with more than 150 characters.
4. Select the **User Interface Server** checkbox and click **Next**.
5. Follow the installation wizard instructions to complete the UI Server installation.
   Consider the following points for DB configuration:
   – On the Manager DB Configuration page, enter the same database information as you did when installing the SA Manager.
   – Enter SA Manager details on the Manager Selection page.
6. Select **Start Services** and click **Next**.
7. Click **Install** in the Pre-Installation Summary page.
8. Click **Done** when the installation completes.

Install the CA SOI Console

The console is a client of the CA SOI hosted on a different server. The console has an interface communication with the SA Manager. You must install the web start application which runs the javasw to open the JNLP file.

Follow these steps:

1. Backup the SOI folder before installing this patch. **Install the Supported JRE from AdoptOpenJDK**
2. Download the supported JRE AdoptOpenJDK website, see Software Requirements for supported JRE.
3. Launch and follow the installation wizard to complete the installation. **Install the icedtea-web Package**
4. Download the (icedtea-web-1.8.x.msi) installer from the AdoptOpenJDK Web Start website.
5. Launch and follow the installation wizard to complete the installation.

NOTE

For more information on how to open the SOI console, see Access the SOI Console page.

Install the Universal Connector Client

The Universal Connector Client installs the framework necessary to run the Universal connector. The Universal connector lets you add CIs, relationships, services, and alerts through a command-line or programmatic interface.

A fully functional Universal connector installs automatically with the SA Manager. You can also install the Universal connector client on a separate system to enable the use of the Universal connector command-line interface on that system.

NOTE
The merge modules that are included in the same Installers folder of the installation image must remain in the same directory as the installer for the installation to work.

Follow these steps:

1. Run **soi-installer.exe** from the **Disk1\SOI** folder of the CA SOI installation image.
2. Click **Next** on the Introduction page.
3. Accept the license agreement, then accept the third-party license agreements, and click **Next**.

   NOTE
   If CA SOI components are already installed on the system, the Choose Install Folder page does not appear. Any new CA SOI components that you install are installed to the existing CA SOI directory because all components must reside in the same directory.
4. Select **Universal Connector Client** check box, and click **Next**.
5. Click **LGPL Distribution Directory**, select the **disk 2 folder (\SOI x.y\disk2\lgpl)** available in the installation kit, and click **Next**.
6. (Optional) Select **Start Services** and click **Next**.
7. Click **Install** on the Pre-Installation Summary page.
8. Click **Done** after the installation is complete.
   The Universal Connector Client installation is completed.

Install the IFW

Install the latest version of the IFW on any system where you want to install CA Catalyst connectors. This procedure is not required if you are installing connectors with the SA Manager, which already has the latest IFW installed.

Consider the following items:

- Connectors that are provided on the CA SOI installation image do not require a separate IFW installation.
- CA Catalyst r3.4.3 connectors require an IFW Proxy installation instead of an IFW installation. For more information about CA Catalyst r3.4.3 connectors and the IFW Proxy, see How to Install CA Catalyst Connectors.
- Because some connectors do not work with a 64-bit IFW, the IFW does not install as a 64-bit application by default. See the **Change the IFW to operate as 64-bit** section in this article, to change the IFW to operate in 64-bit environment.

Follow these steps:

1. Start the IFW container installation according to your operating system:
   Linux:
   Execute the IntegrationServices.bin command, as root or sudo su, from the installation directory.
   Windows:
   Run IntegrationServices.exe from the Disk1\SOI folder of the CA SOI installation image.
2. Click **Next** on the Introduction page.

3. Accept the license agreement, then accept the third-party license agreements, and click **Next**.
4. Accept or modify the installation folder.

   NOTE

   The maximum installation path length is 150 characters. The installation blocks paths with more than 150 characters.
5. Complete the fields to connect to the **SOI Manager host** and **MQ Server host,** configure connector preferences, and click **Next**. Refer to your Installation Worksheet in the SA Manager section for these values.
6. Specify whether to start the product services after installation, and click **Next**.
7. Click **Install** on the Pre-Installation Summary page.
   An installation summary page opens when the installation finishes.

   NOTE

   In Linux, if the IFW container did not start automatically, start the soi-eventmanager.sh and soi-connectors.sh services manually.

Change the IFW to 64-bit Mode [Only for the Windows System]

Because some connectors do not work with a 64-bit IFW, the IFW installs as a 32-bit application by default. However, you can manually change the IFW to run on 64-bit systems. For more information about your connector, refer to the guide that is provided with your connector.

To avoid performance issues, if the connector server you monitor receives more than 100,000 CIs daily, we recommend enabling the 64-bit IFW.

Follow these steps:

1. Stop the CA SAM Integration Services service on the server where the IFW is installed.
2. Navigate to the <SOI_HOME>\jsw\conf folder.
3. Open the SAM-IntegrationServices.conf file and find the following line:
   ```
   # #include ../conf/IFW-wrapper-jvm-64.conf
   ```
4. Uncomment the line by removing the first pound sign and the space before the second pound sign:
   ```
   #include ../conf/IFW-wrapper-jvm-64.conf
   ```
5. Restart the CA SAM Integration Services service.
6. Repeat Steps 1-7 on other servers where you want to change the IFW to 64-bit mode.
7. To verify that the service is running in 64-bit mode, do the following:
   - Find the PID of the service in the SOI_HOME\jsw\logs\SAM-IntegrationServices_wrapper-java.pid file.
   - Look up the PID number in the Windows Task Manager to locate the running process. If the process displays as java.exe, it is running as 64-bit. If it displays as java.exe*32, it is running as 32-bit.

Verify Component Services and Connections

To verify a successful installation, verify the installed services and the connection status of each installed component.

Follow these steps:

1. Select Start, Programs, Administrative Tools, Services.
2. Verify the existence of the following services on the appropriate servers, depending on where the components were installed:
   - **CA SOI Application Server**
     Controls the operation of the SA Manager. This service is installed on any system that contains the SA Manager component.
   - **CA SOI Event Management**

Controls communication with the Event Store on connector systems for Event Management. This service is installed on any system that contains the SA Manager or a connector.

– **CA SOI Integration Services**
Controls the operation of the IFW, which handles communication between the connectors and the SA Manager. This service is installed on any system that contains a connector or the MQ Server, which is a component of the SA Manager.

– **CA SOI MQ Server**Controls the operation of the MQ Server. This service is installed on any system that contains the MQ Server component.

– **CA SOI Store Indexer**
Controls the indexing of USM data from the Persistent Store for use by the USM Web View. This service is installed on any system that contains the UI Server.

NOTE
For specific information about the Persistent Store, see components.

– **CA SOI User Interface Server**
Controls the operation of the UI Server, including all user interfaces. This service is installed on any system that contains the UI Server component.

– **CA SOI UCF Broker**
Controls the UCF broker, which facilitates create, update, and delete operations from connectors to their source domain managers. This service is installed on any system that contains the SA Manager.

All CA SOI services start automatically after installation unless you specified otherwise.

3. (Optional) Start all CA SOI services that are stopped.
If you notice problems with the services starting or running consistently, view the service log files. The files are at SOI_HOME\jswl\logs and distributed across component-specific subfolders.
4. Enter the following URL in your Web browser:
```
http://uiserver:port/sam/ui
```
– *uiserver*
Defines the host name where you installed the UI Server.

– *port*
Defines the UI Server HTTP port that you specified during installation.
Default: 7070

An authentication dialog opens.
5. Enter the administrator user credentials that you specified during CA Catalyst and CA SOI installation, and click OK.

NOTE
The CA SOI Dashboard opens.

6. Click Console.
7. Click the Connection icon

at the bottom right of the Console.

Installation Troubleshooting

Use the following methods to troubleshoot installation issues when the Install Complete page indicates that the installation that is completed with errors or you notice problems with CA SOI:

- Verify that all servers have valid DNS names and are connected to a valid DNS server.
- Review the installation log file (CA_Service_Operations_Insight_Install_*releasenumber*) that is in the SOI_HOME\log\ folder to check for installation errors.
- Review the soimgr.log file at SOI_HOME\tomcat\logs for errors initializing the SA Manager.
- Review the soiuis.log file at SOI_HOME\SamUI\logs for errors initializing the UI Server

See the following sources for more troubleshooting information:

- See Troubleshooting?for log file information and troubleshooting procedures that are related to product usage.
- See Release Information for known issues and workarounds.

Configure 32-bit Systems

If you installed CA SOI components in discrete systems with a 32-bit version of Windows, configure the systems to support current release of CA SOI.

Follow these steps:

1. Install JDK_1.8.0_45_x86
2. Copy the Server folder from <JDK_HOME>\jre\bin\ and paste it in the <SOI_HOME>/jre/bin" and "SOI_HOME/jre-32/bin"
3. Open the command prompt.
4. Browse to SOI_HOME/Tools and execute soiToolBox —-restartAllServices.

Connector Installation

CA SOI provides the following connectors either on the installation image or as related downloads:

- Universal connector?(integrated into the main CA SOI installer)
- Sample connector
- Mid-tier connector (installed when you install the SA Manager)
- CA SOI Domain connector

Consider the following when installing the Sample and Domain connectors provided on the CA SOI image:

- If you are copying the installation program from the installation image to another system, verify that the merge modules file (*component*.iam.zip file) is in the same folder as the installation executable.
- For specific installation requirements, see the topics for each connector.

Connector Installation Considerations

Consider the following before you install connectors in your CA SOI environment:

- Verify that the SA Manager is already installed before you install connectors. The following SA Manager information is required during connector installation:
  - Host name
  - MQ Server port number (61616 by default)
  - Administrator user credentials
- Install all connectors from a local image or DVD or from a network share. Installers for the connectors do not support the use of UNC shares.
- Multiple connector installations are supported on one system. If you plan to install connectors on a system with other CA SOI components or multiple connectors on the same system, see Hardware Requirements to verify that the system has the appropriate resources to support the installations.
- As a best practice to avoid memory issues, connectors on one system must manage no more than 200,000 CIs combined. For example, if you have four CA eHealth installations that manage a total of 300,000 CIs, you must distribute the four remote CA eHealth connectors across at least two systems. Also, consider the number of events that each connector will manage if your overall deployment represents a significant event stream. In this case, try

to distribute the event stream evenly across remote connector systems and (when possible) group connectors that require the same type of processing.

- Some connectors require installation directly on their domain manager and others support a remote installation. The installer blocks installation of any connector that requires a local installation when the domain manager is not present on the system. For more information about whether a connector supports remote installation and other prerequisites and considerations, see the product-specific *Connector Guide* that is provided with that connector.

  NOTE
  For a list of provided product connectors and the domain manager versions they support, see the CA SOI product page on CA Support Online.

How to Install CA Catalyst Connectors (Pre-r3.2)

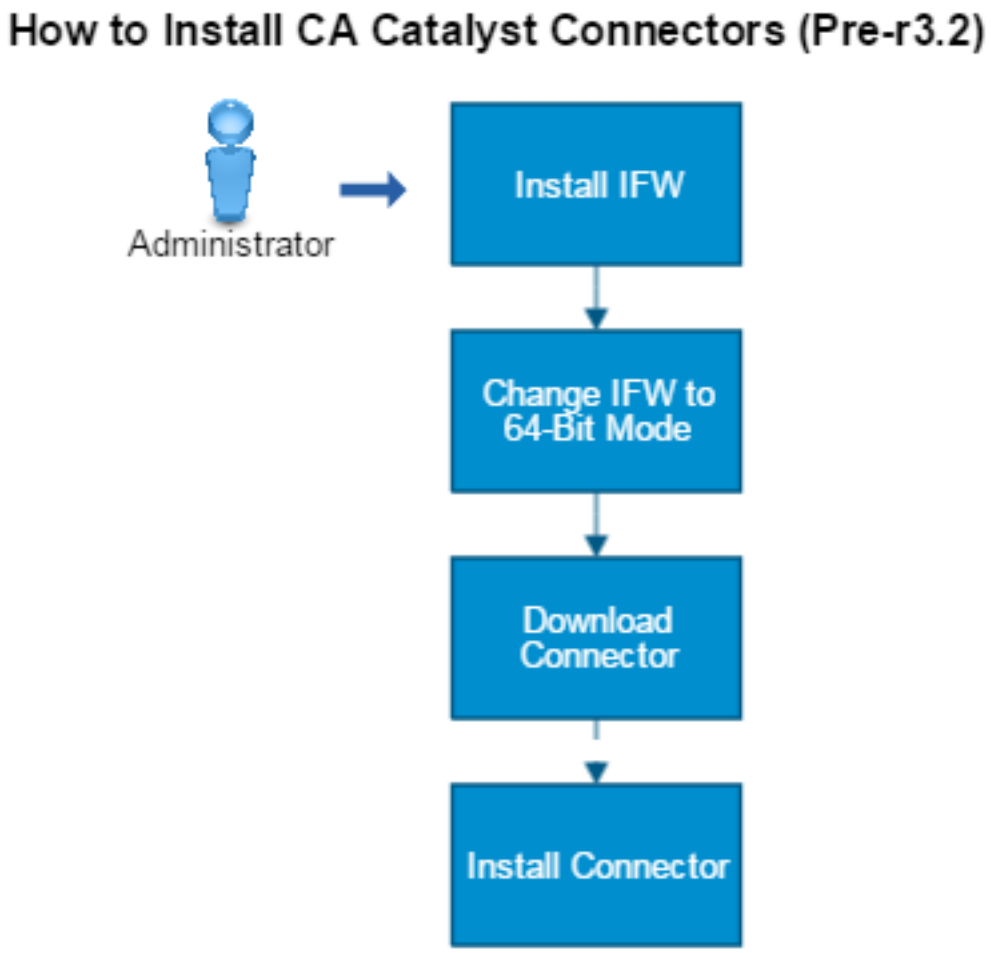
Contents

CA Catalyst connectors prior to CA Catalyst r3.2 integrate with a specific domain manager or generic data source (such as SNMP traps) and are provided as downloadable published solutions.

For the most recent list of available connectors, see the connector download page on CA Support.

Use this scenario to guide you through the process:

Figure 9: How to Install CA Catalyst Connectors (Pre r3.2)



Complete the following process to install CA Catalyst connectors:

1. Install the IFW from the CA SOI image on the connector system.

   NOTE
   If you are installing connectors with the SA Manager, which already has the latest IFW installed, skip to Step 2.
2. (Optional) If your connector supports 64-bit, change the IFW to 64-bit mode.
3. Download the connector and prepare for installation.
4. Install the connector.
5. Continue with other installations as described in How to Perform a Full CA SOI Deployment:
   – Install the reporting component.
   – Deploy a tiered SA Manager environment.

Download Connectors and Prepare for Installation

The base CA SOI image does not provide CA Catalyst connectors. Download CA Catalyst connectors from the CA SOI product page on CA Support Online and install them separately. For the connector installation to work, you must extract downloaded connector files so that an exact folder structure is created.

Follow these steps:

1. Access CA Support Online and log in.
2. Click the Support by Product link and select CA Service Operations Insight from the Select a Product page drop-down list.
3. Locate the Recommended Reading section and click CA Service Operations Insight - Connectors.
4. Download the patch for the CA Catalyst connector that you want to install, and copy it to the system where you have the CA SOI installation image.
5. Extract Connector_*ProductName*.zip to the Disk1\SOI folder of the installation image.
6. Create the Merge_Modules folder in the Connector installer and copy IntegrationServices.iam.zip from the installation image in the Disk1\SOI folder.
   The connector installation fails unless the following folder structure exists in the folder where you extracted the zip files:
   - SOI
     a. Merge_Modules (contains IntegrationServices.iam.zip)
     b. Connector_*ProductName*.exe
   - Documentation
     a. PDFs
     b. Readme

   When you extract the zip files from the same location, the correct folder structure is automatically created.
   Multiple Connector_*ProductName*.exe files can use the same Merge_Modules folder, if you want to install multiple connectors on the same system.
7. Check the *ProductName Connector Guide* and Connector Readme provided with the connector package (in the Documentation folder) before you start the installation.
   Many connectors have required preinstallation steps.

Install CA Catalyst Connectors

Install CA Catalyst connectors to establish integrations with specific domain managers and data sources.

Consider the following items:

- Connectors that are provided on the CA SOI installation image do not require a separate IFW installation.
- CA Catalyst r3.2 connectors require an IFW Proxy installation instead of an IFW installation. For more information about CA Catalyst r3.2 connectors and the IFW Proxy, see How to Install CA Catalyst r3.2 Connectors.
- Because some connectors do not work with a 64-bit IFW, the IFW does not install as a 64-bit application by default. However, you can change the IFW to operate as 64-bit.

> **NOTE**
> CA Catalyst connectors reference the old product name CA Spectrum Service Assurance and the previous version number r2.5 in several places, such as the installer, connector-specific documentation, and the installed Start menu shortcut. However, these connectors do work with the current release of CA SOI.

Follow these steps:

1. Double-click the Connector_*ProductName*.exe file located in the SAM folder at the location where you extracted the connector.

   > **NOTE**
   > The zip files must be in the Merge_Modules folder.

2. Click Next.
3. Scroll to the bottom of the agreement, select "I accept the terms of the License Agreement" and click Next.
   The *Product Name* Connector Configuration page opens.
4. Enter the required information for connecting to the domain manager, and click Next:

   NOTE
   The product-specific *Connector Guide* provided in the Documentation folder of the connector package contains descriptions for all required domain manager information.
5. Specify whether to start the product services automatically after installation, and click Next.
6. Review your selections, and click Install.
   The Install Complete page opens when the installation finishes.

Most connectors install a log file for troubleshooting installation errors. For more information, see the product-specific*Connector Guide* that is provided in each connector package. To verify that the connector installed and initialized correctly, view the connector status on the Administration tab.

How to Install CA Catalyst r3.x Connectors

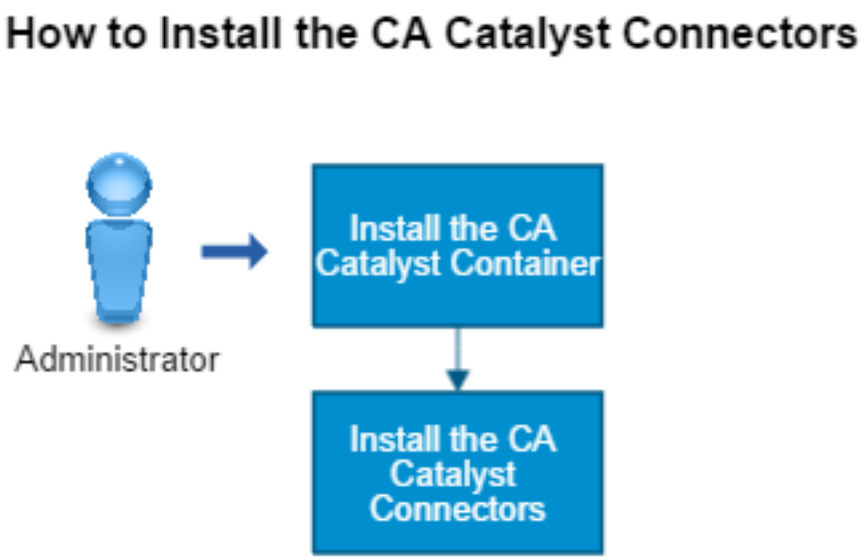
Contents

As an administrator, you install the CA Catalyst r3.x Container to facilitate connector communication with the IFW.

CA Catalyst r3.x connectors support the CA Catalyst r3.x infrastructure that is not embedded in the SA Manager. Once the communication is established between the IFW and the CA Catalyst r3.x connectors, the connectors operate the same as traditional CA Catalyst connectors.

Use this scenario to guide you through the process:

Figure 10: how to install connectors



1. Install the CA Catalyst r3.4.1 Container.
   Download the Container on the CA Support Online connector download page.

NOTE
The CA Catalyst r3.4.1 documentation contains detailed information about installing the Container, verifying the installation, working with connectors, and more.

2. Install the CA Catalyst r3.2 connectors.
   On the connector download page, connectors that display Container in the Prerequisites column are CA Catalyst r3.2 connectors.

   NOTE

   For basic information about installing r3.2 connectors, see the CA Catalyst r3.4.1 documentation. For detailed information about installing specific connectors, see the *Connector Guide* included in the package for that connector.
3. Continue with other installations as described in How to Perform a Full CA SOI Deployment:
   – Install the reporting component.
   – Deploy a tiered SA Manager environment.

Install CABI (JasperReports Server)

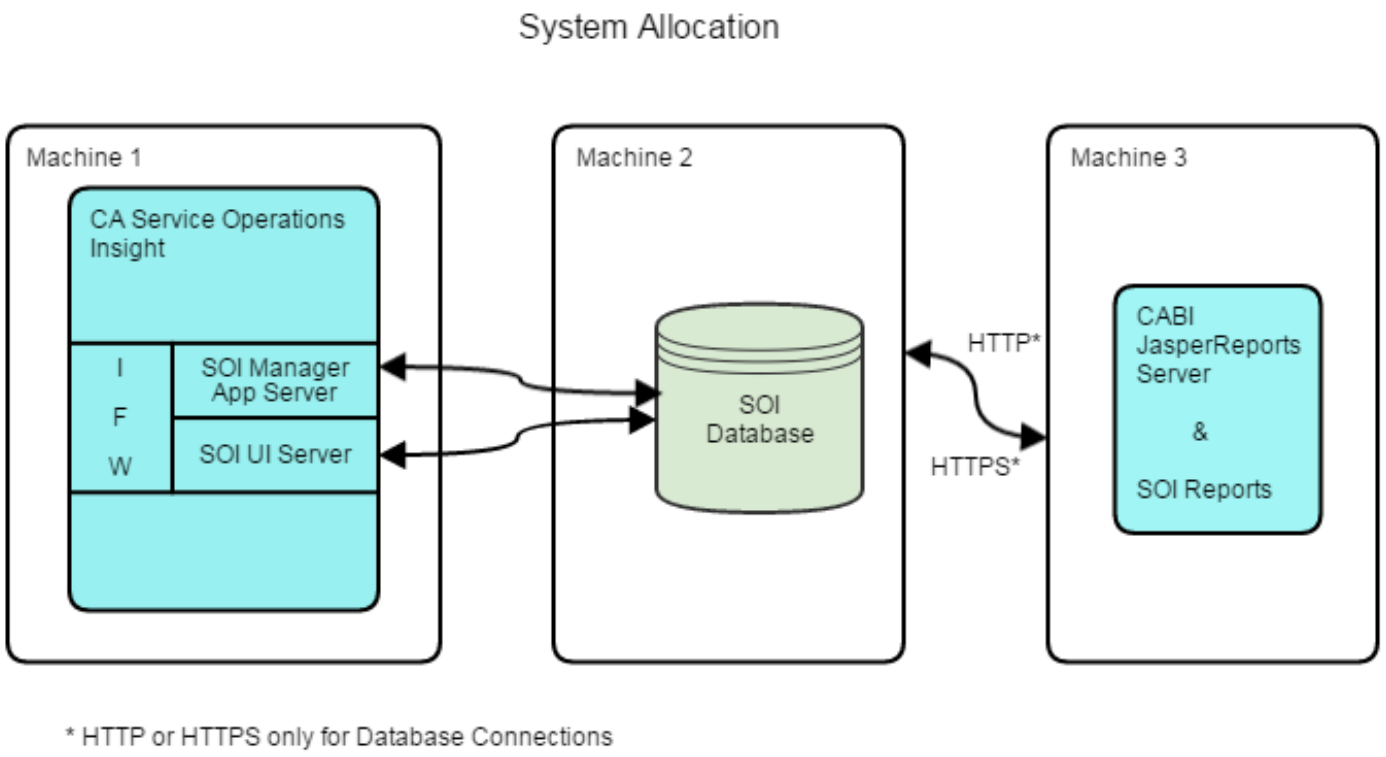
Contents

As an administrator, you install CA Business Intelligence (CABI) to provide reporting capabilities. Starting this version, CABI is packaged with TIBCO JasperReports Server. If you have not installed the previous version of CABI reporting solution, install SOI Reports with CABI JasperReports Server to run and view reports. You need to perform extra configuration for reports to function correctly. The following diagram shows how to work in CABI JasperReports Server.

Understand the System Allocation

SOI Reports and CA Business Intelligence JasperReports Server must reside on a standalone system. The system that hosts CABI JasperReports is to have a minimum of 10 GB hard disk space with an Eight GB memory. (Please refer this Certification Matrix for more information about Minimum Hardware Requirements.) You can install SOI Reports with or without CABI JasperReports Server. If you plan to install only SOI Reports, install CA Business Intelligence JasperReports Server r6.1.0 on a dedicated system before you install SOI Reports.

Figure 11: SOI-Jasper System Allocation



Installation Methods

- Install latest CABI JasperReports Server by following the procedure provided here. Then, install SOI Reports and configure it.

SOI Reports installer supports only Microsoft SQL Server database. So, if you are using the Microsoft SQL Server database and you want to install CABI JasperReports Server, use the SOI Reports installer. If you are using the MySQL or PostgreSQL database, follow the installation procedure provided in the CABI JasperReports Server document.

Install SOI Reports

SOI Reports include scheduled and predefined reports. You can enable CA SOI reporting by importing the SOI Reports into CA Business Intelligence JasperReports Server. To install SOI reports on Windows, see Install SOI Reports.

Prerequisites:

- Confirm you are installing CABI JasperReports Server and the CA SOI reports on a dedicated server with no other CA SOI components.
- Ensure that the CABI JasperReports Server is installed.

> **NOTE**
> To install CABI JasperReports Server, click here to see the procedure. Remember, SOI Reports supports CA Business Intelligence JasperReports Server r6.3.0.

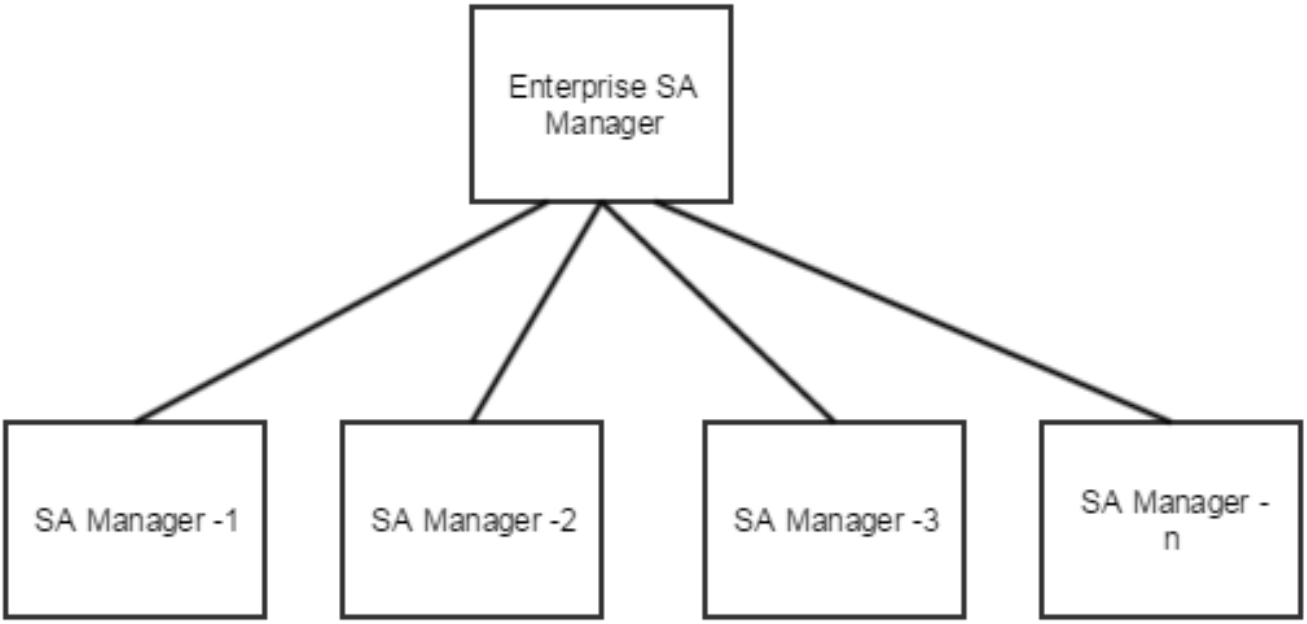
Check the default values in Installation Worksheet for CA Business Intelligence JasperReports Server.

Deploy SA Manager in Tiered Environment

As an administrator, you can deploy a tiered SA Manager that consists of multiple source SA Manager tiers reporting to an Enterprise SA Manager through the CA SOI Domain connectors. You can tier your CA SOI environment in the following situations:

- You want to provide a multi-tenant environment with a physical separation of resources.
- You want to structure your enterprise by region, location, or department.

Figure 12: Tiered SA Manager



Multiple CA SOI Deployments

Perform as many full CA SOI installations as required for your environment. For example, install three SA Managers and the necessary accompanying components if you require two full installations reporting to an enterprise level. Consider the following items while completing the deployment phase:

More than two tiers are supported.

- Install only the required components for each SA Manager. For example, the Enterprise SA Manager may not require any connectors.
- You can configure services on the source SA Managers before or after installing the CA SOI Domain connectors. The services that are created at Domain SA Manager after setting the Enterprise SA Manager can also be imported without restarting the Domain connector.
- Assign the appropriate user privileges for each installation. For example, if the Enterprise SA Manager is an enterprise view of multiple tenants' data, only provide access to administrator-level users that require a consolidated view of all tenant data. For more information about security, see. How to Configure Role-Based Security.
- Each SA Manager requires separate configuration. For example, if you want to configure all tiers to use the same email server for escalation actions, configure the email separately on each SA Manager.

Follow these steps to deploy SA Manager in a tiered environment:

1. Install a CA SOI Domain connector on an Enterprise SA Manager. You can install multiple domain connectors on a single IFW system that points to the Enterprise SA Manager.
2. Import the services into the Enterprise SA Manager.
3. (Optional) Review the CA SOI Domain connector properties.
4. Continue with other installations as described in How to Perform a Full CA SOI Deployment:
   – Install the reporting component.
   – Install CA Catalyst r3.x connectors.

CA SOI Reports with Unified Dashboards and Reporting for Infrastructure Management

You can install CA SOI reports and dashboards on the system where CABI JasperReports Server for Unified Dashboards and Reporting for Infrastructure Management 6.3 version.

As an administrator, you install CA Business Intelligence 6.3 (CABI) to provide reporting capabilities. You can share a single instance of CABI JasperReports Server with the following CA Agile Operations products for combined views of network performance:

- CA Service Operations Insight (CA SOI)
- CA Unified Infrastructure Management (CA UIM)
- CA Spectrum
- CA Performance Management (CA PM)

The major benefits of Unified Dashboards and Reporting for Infrastructure Management are as follows:

- Reduces the number of CABI instances that you must deploy and maintain for different CA products.
- Enables you to view dashboards and dashlets for multiple CA products, which provide you better insight into your business.

Perform the following procedures to configure dashboards and run reports.

Prerequisites

- Ensure that the CABI JasperReports Server for Unified Dashboards and Reporting for Infrastructure Management 6.3 is installed on your system. You can install **CABI JasperReports Server for Unified Dashboards and Reporting for Infrastructure Management** from CA Support Site.
- Ensure that CABI JasperReports Server for Unified Dashboards and Reporting for Infrastructure Management must be installed on Windows server to integrate with CA SOI 4.2.

Install and Configure SOI Reports

This section describes how to install and configure SOI reports on Windows and Linux .

Install CA SOI Reports on Windows

The CA SOI Reports include dashboards and reports.

NOTE

Install SOI reports on the system where *CABI JasperReports Server for Unified Dashboards and Reporting for Infrastructure Management* is installed

Follow these steps:

1. Double-click **reports.exe** file from SOI 4.2 installation media.
2. Click **Next** on the Introduction window.
3. Accept the license agreement and click **Next**.
4. Specify the location where you want to install the reports, and click **Next**.
5. Browse for the location where **CABI JasperReports Server** is installed. Select the **CA Business Intelligence** folder, and click **Next**.
6. Enter the following database details and click **Next**.
   a. **Host:** Specifies the hostname of the server where the CA SOI database (SAMStore) repository is already available.
   b. **JDBC Port:** Specifies the port number to connect to the CA SOI database.
      a. **Dynamic:** Specifies the dynamic instance if not using the default port number.
      b. **Static:** Specifies the port number to connect to the CA SOI database. Default: 1433.
   c. **DB Credentials:** Enter the following values:
      a. **User:** Specifies the username of the database repository.
      b. **Password:** Specifies the password of the database repository.
      c. **Database Name:** Specifies the CA SOI repository name.
   d. Enter the following JasperServer details and click **Next**.
      a. **Select Protocol:** Enter the protocol, HTTP, or HTTPS.
      b. **JasperReports Server Port:** Enter the port. HTTP/HTTPS port that is configured for the CABI JasperReports Server**. Default:** 8080
      c. **JasperReports Server instance name:** Enter the WebApp Server instance name. **Default:** Jasperserver-pro
      d. **User id/Password:** Login credentials of the user with administrator privileges who do not belong to an organization in JasperReports Server. **Default:** superuser/superuser
7. After the connection is established with CABI JasperReports Server, the preparing for installation Summary page appears.
8. Review the pre-installation summary, and click **Install**.
9. Click **Done** when the installation finishes.
   The Reports and dashboards are installed.

   NOTE

   For superuser and a valid SOI user, the following CA SOI folders are created in CABI JasperReports Server interface:

   - For superuser, the folder structure is as follows:
     – **Existing (Older) Reports:** root > organization > Service Operations Insight
     – **New Reports:** root > Public > ca > Service Operations Insight
   - For a valid SOI user, the folder structure is as follows:
     – **New Reports:** Public > ca > Service Operations Insight
     – **Existing (Older)** Reports: Service Operations Insight

   The new CA SOI contentis available in the new folder structure (N**ew Reports**) as mentioned above. If you import older reports, the content of the reports is available in the older folder structure (**Existing (Older)** Reports) as mentioned above.

Configure and Run CA SOI Reports

After you have installed the reports, configure and run CA SOI reports.

NOTE

Before you run the CA SOI reports:

- A valid SOI user must exist in JasperReports Server.
- Ensure that you enable the report functionality and configure CA SOI to automatically synchronize CA SOI (new users and existing users) with CABI JasperReports Server user. For more information about Configuring CA SOI Reports, see Configure CA SOI Reports for CABI JasperReports Server.
- Ensure that you restart **CA SOI Application Serve**r for the existing CA SOI user to synchronize with CABI JasperReports Server.

Follow these steps:

1. Enter **http://*<Jaspersoft hostname>:<tomcat portnumber>/<webappname>*** . For example, http://localhost:8080/jasperserver-pro
2. Log in to the JasperReports Server with superuser.
3. Create an SOI user under soi organization in the JasperReports Server.
4. **Generate Reports:**
   a. Log in to JasperReports User interface with a valid SOI Report user.
   b. Select a required dashboard or report, and click **Run.**
   When you log in to CABI JasperReports Server as superuser, you have administrator rights. You may not able to run reports. To run and view reports, log in as a valid SOI Report user. To add users, see Adding Users.

   The CA SOI reports are configured.

Configure and Run CA SOI Reports on Linux

Linux Files

To download the Linux Files, navigate to **CA Service Operations Insight Solutions and Patches** page, download and extract **RO97190.zip** folder.

To configure and run CA SOI reports on Linux, use the following files from the extracted folder:

- applicationContext.xml
- ca_customMethods.jar
- dashboardHyperlinkHandlerCabiDashboardDrillAction.js
- loadTPA.jsp
- soi_reports_content6.3.zip
- soi_roles_201504121522.zip

Follow these steps:

1. Log in to JasperReports Server with superuser
2. **Create soi Organization:**
   a. Click **Manage**, **Organizations**, and **Add Organization.**
   b. Provide the organization details such as **Organization Name**, **Organization ID**, **Organization Description, and** Description.
3. Import zip Files from the extracted folder (**RO97190.zip**): Import the following files individually in the JasperReports Server system by clicking **Manage, Server Setting, Import**.
   a. **soi_roles_201504121522.zip**
   b. **soi_reports_content6.3.zip**
4. **Update the JDBC Configuration in CABI JasperReports Server:**
   a. Click **View, Repository** and expand **root, Public, ca, Service Operations Insight** folder**.**
   b. Click **datasources** folder**,** and edit the **"soi ds"** datasource**.**
   c. Update the datasource to point to CA SOI database (SAMStore), and click **Test Connection**.

   d. Click **Save** after the successful connection.
   e. Select the correct datasources folder under **Service Operations Insight** folder, and click **Save** in the Save window.
5. Copy **ca_customMethods.jar** file from the extracted folder (**RO97190.zip**) to **<CABI Installed Folder/<apache tomcat server>/webapps/jasperserver-pro/WEB-INF/lib** folder.
6. Copy the **dashboardHyperlinkHandlerCabiDashboardDrillAction.js** file from the extracted folder (**RO97190.zip**) to <CABI Installed Folder>/**<apache tomcat server>**/webapps/jasperserver-pro/scripts and <CABI Installed Folder>/**<apache tomcat server>**/webapps/jasperserver-pro/optimized-scripts
7. Open **validation.properties** file located <CABI Installed Folder>/**<apache tomcat server>**/webapps/jasperserver-pro/WEB-INF/classes/esapi folder and change the **Validator.ValidSQL** value as follows:

```
Validator.ValidSQL=(?is)^\\s*(select|with)\\s+[^;]+;?\\s*$
```

8. **Update the Time Zone:** As CA SOI supports different timezones, update the default jasper configuration as follows:
   a. Open **applicationContext.xml** located in the extracted folder (**RO97190.zip**), search for **<property name="timeZonesIds">** attribute and copy all the attribute from **<value>Asia/Kolkata</value>** until **<value>Etc/GMT-14</value>**
   b. Open the **applicationContext.xml** file which is in the <CABI Installed Folder>**<apache tomcat server>**\webapps\jasperserver-pro\WEB-INF folder and paste the attributes that you have copied (mentioned in step a).
9. Restart the **CA Business Intelligence Tomcat** Service.
10. Create an SOI user under soi organization in the JasperReports Server.
11. **Generate Reports:**
   a. Log in to JasperReports User interface with a valid SOI Report user.
   b. Select a required dashboard or report, and click **Run.**
   When you log in to CABI JasperReports Server as superuser, you have administrator rights. You may not able to run reports. To run and view reports, log in as a valid SOI Report user. To add users, see Adding Users.

   The CA SOI reports are configured.

Configure Single Sign-On

Single sign-on (SSO) is available for reports so that you can run reports from the CA SOI Dashboard without providing the login credentials for CABI JasperReports Server. You can still access reports without SSO configured, but you need to enter the login credentials each time. For more information, see Configure Single Sign-On.

Upgrade CA SOI Reports to CABI JasperReports Server 6.3

Consider the following points when you upgrade CA SOI Reports to CABI JasperReports Server 6.3:

- The CA SOI folder structure is upgraded and the SOI content is placed under **Public > ca** folder.
- If you perform customizations such as updating reports, creating reports, then ensure that you export the old or customized reports and import them to CABI JasperReports Server.
- Exporting of existing reports and importing them to a new CABI JasperReports Server creates the following folder structure:
  - For superuser, the folder structure is as follows:
    - **Existing (Older) Reports:** root > organization > Service Operations Insight > capability > reports
    - **New Reports:** root > Public > ca > Service Operations Insight > reports
  - For a valid SOI user, the folder structure is as follows:
    - **New Reports:** Public > ca > Service Operations Insight > reports
    - **Existing (Older)** Reports: Service Operations Insight > reports

- Both the existing and new reports work as expected. Ensure that you delete the older reports manually from CABI JasperReports Server after complete customizations on new reports. Delete the reports that are in the folder structure, **Service Operations Insight > reports.**

  NOTE

  As an operator, you can add or remove users in CA SOI report group and schedule or run predefined reports in JasperReports Server. You use these reports to view metrics and service details over specified time ranges. Your administrator defines your access permission to generate reports.

Migrate Data to Unified Dashboards and Reporting for Infrastructure Management

If you perform customizations such as updating reports, creating reports, then ensure that you export the old or customized reports and import them to CABI JasperReports Server. Use the following process to migrate data from a standalone CABI JasperReports Server to a shared CABI Server.

Follow these steps:

1. Export report from the standalone CABI server.
   a. Log in to the standalone CABI instance as a system administrator (for example, superuser).
   b. Navigate to Manage, Server Settings, and click Export.
   c. Specify the Export Data File Name.
   d. Select Reports, and then click Export.
2. Import the exported report data into the shared CABI server.
   a. Log in to the shared CABI server as a system administrator.
   b. Navigate to Manage, Server Settings, then click Import.
   c. Select the file to import, then click Import.

Dashboards in CABI JasperReports Server

CA SOI provides the following two dashboards in CABI JasperReports Server.

- Alerts View
- Business Service View

These dashboards display information of the alerts and services and contain the following five dashlets:

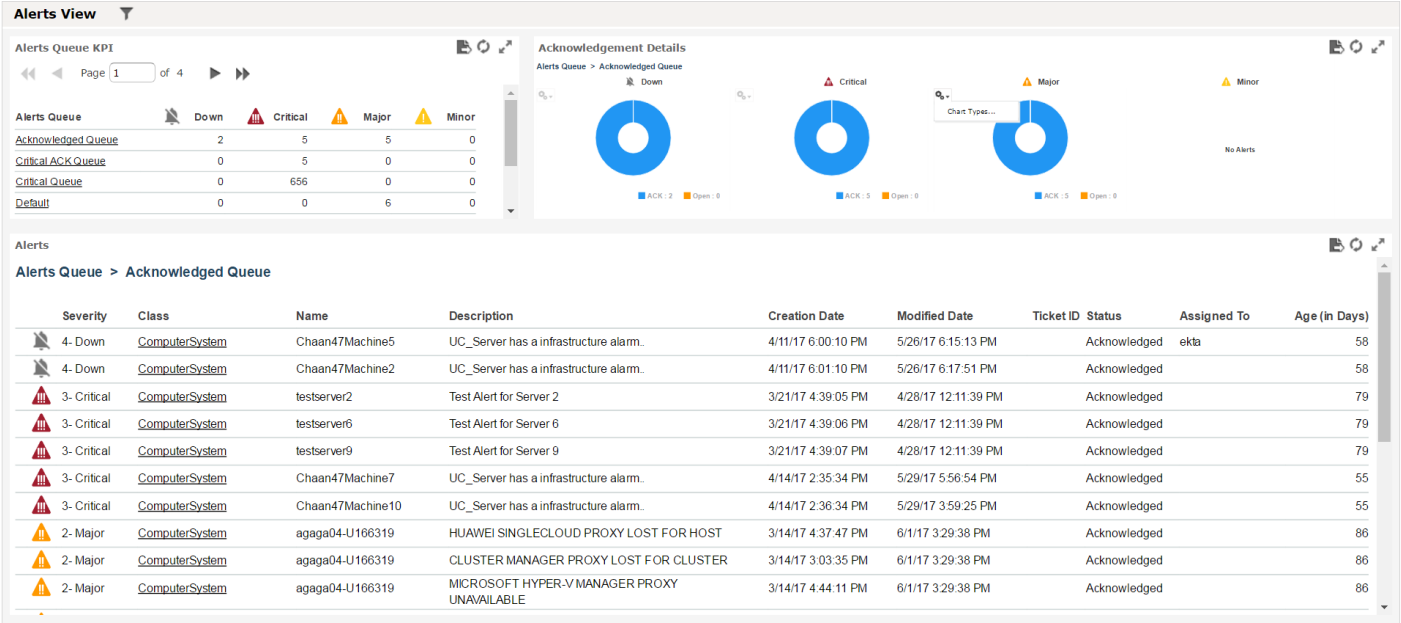
- Alerts Queue KPI
- Acknowledgment Details
- Alerts
- Business Service View
- Alerts per Business Service

  NOTE

  The dashlets depend on each other. For example, when you select **alertqueue1** in the Alerts Queue KPI dashlet, the information specific to the **alertqueue1** appears in the Acknowledgment Details and Alertsdashlet.

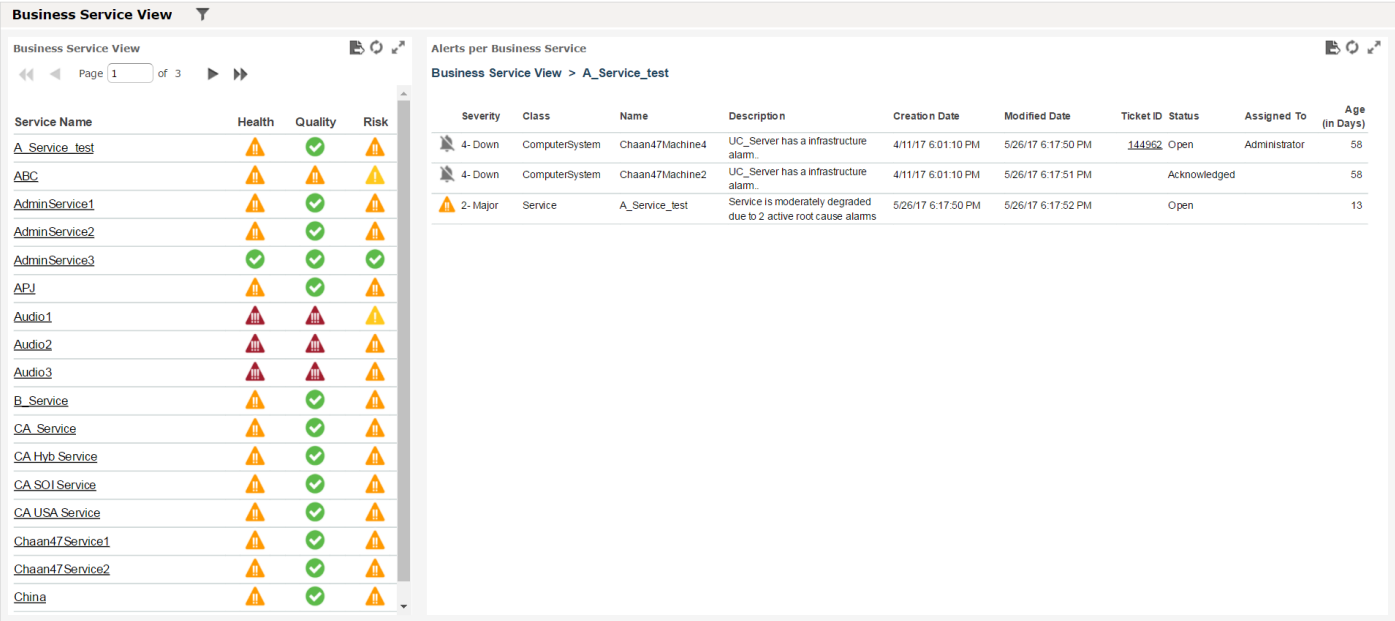
- **Alerts View:** Displays detailed information about the alerts. This dashboard contains the following dashlets:
  – Alerts Queue KPI
  – Acknowledgment Details
  – Alerts

  Alert View Path: /public/ca/Service Operations Insight/dashboards/common/Alerts View



- • **Alerts Queue KPI:** Displays information about alert queues that are sorted alphabetically. By default, the first alert queue is selected.
  Path: /public/ca/Service Operations Insight/reports/common/alerts/Alerts Queue
  - **Acknowledgment Details:** Displays the list of alerts that are acknowledged in the form of a pie chart. The acknowledgment details appear only for the selected alert queue.
    Path: /public/ca/Service Operations Insight/reports/common/alerts/Acknowledgment Details
  - **Alerts:** Displays detailed alert information for each alert queue that you select in the Alerts Queue KPI dashlet. The alerts are sorted by severity. Information such as Severity, Class, Name, Description appears in the Alerts dashlet.
    Path: /public/ca/Service Operations Insight/reports/common/alerts/Alerts
- **Business Service View:** Displays detailed information about the services. This dashboard contains two dashlets:
  – Business Service View
  – Alerts per Business Service

  Business Service View Path: /public/ca/Service Operations Insight/dashboards/common/Business Service View



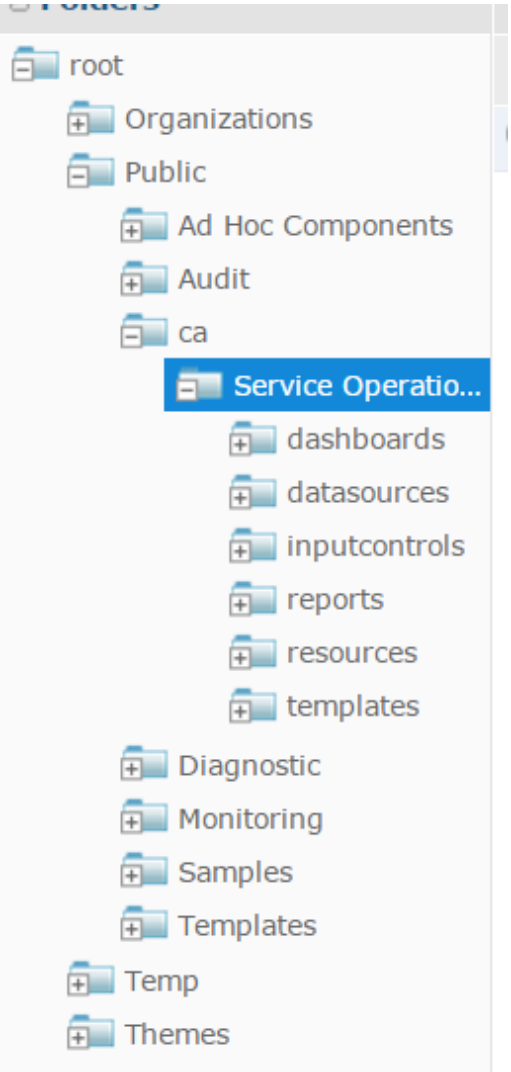
– • **Business Service View:** Displays list of all the services. This dashlet contains information such as service name, health, quality, and risk of the selected service.
  Path: /public/ca/Service Operations Insight/reports/common//business service/Business Service View
  • **Alerts per Business Service:** Displays detailed alert information for each service that is selected in the Business Service View dashlet.
  Path: /public/ca/Service Operations Insight/reports/common/business service/Alerts per Business Service

CA SOI Content Folder Structure:

The CA SOI product content such as dashboards, reports, input controls, data sources, images is now located in the public folder in CABI JasperReports Server. For example, the new folder structure for the SOI content is as follows:

For a valid SOI user: Public > ca > Service Operations Insight > reports

For superuser: root > Public > ca > Service Operations Insight > reports



Configure Dashboards Tabs on CA SOI Interface

You can configure the dashboards of CABI JasperReports Server on CA SOI Interface.

Follow these steps:

1. Log in to the CABI JasperReports Server as a valid CA SOI user.
2. Click the required dashboard, and copy the URL from the address bar.
3. Log in to the CA SOI interface.
4. Click **Preference** on the left side of the Dashboard screen.
5. Click Custom Links tab and enter the following details:
   a. **Web Address:** specifies the dashboard url that you have copied in step 2.
   b. **Tab Tile:** specifies the dashboard tab name.
   c. **Show:** Select this option for the dashboard tab to be added in the CA SOI interface.
6. Click **Save**.

The dashboard tab is added in the CA SOI interface.

Reports Generation in CABI JasperReports Server

As an operator, you can add or remove users in CA SOI report group and schedule or run predefined reports in JasperReports Server. You use these reports to view metrics and service details over specified time ranges. Your administrator defines your access permission to generate reports.

As the viewing and managing reports are performed in JasperReports Server, you can access the online help from JasperReports Server.

For more information about generating the reports, see Generate Reports in CABI JasperReports Server

Uninstall CA SOI Reports

Prerequisite:

- Ensure that you remove all the references of CA SOI dashlets from the customized dashboards before you uninstall CA SOI reports.

Uninstall CA SOI Reports on Windows

To uninstall soi reports, follow these steps:

1. You can uninstall the CA SOI reports in *one* of the following ways:
   – Navigate to **<CA SOI reports Installed directory>\Reports_uninstaller**, and click **Uninstall SOI-Reports.** For example, \Program Files\CA\SOI\Reports_uninstaller.
   – Click **Start, All Programs, CA**, **Uninstall SOI-Reports.**

   Perform the following steps after complete uninstallation of reports.
2. (Optional) Navigate to **<CABI Installed Folder>\apache-tomcat\webapps\jasperserver-pro\WEB-INF** folder and remove **applicationContent.xml** file. Rename the applicationContext.xml_backup_ to applicationContext.xml
3. (Optional) Navigate to **<CABI Installed Folder>/apache-tomcat/webapps/jasperserver-pro/WEB-INF/classes/esapi folder, and** remove **validation.properties** file. Rename the validation.properties_backup_ file to validation.properties.
4. Restart CA Business Intelligence Tomcat Server service.

   NOTE

   You can place the loadTPA.jsp (backup file) in **<SOI_Home>\SamUI\webapps\sam\ui** folder. Ensure that you restart the UI Server after placing the loadTPA.jsp file.

Uninstall CA SOI Reports on Linux

To uninstall soi reports on Linux, follow these steps:

1. Log in to JasperReports User interface with superuser.
2. Navigate to View, Repository, and expand **root, Public, ca,** right click on the **service Operations Insight folder** and click Delete.
3. Navigate to Manage, Organizations, select root folder, soi organization and click **Delete Organization**.
4. Remove **ca_customMethods.jar** file from **<CABI Installed Folder>/apache-tomcat/webapps/jasperserver-pro/WEB-INF/lib** folder.
5. Remove **dashboardHyperlinkHandlerCabiDashboardDrillAction.js file** from the following locations:
   – <CABI Installed Folder>/apache-tomcat/webapps/jasperserver-pro/scripts
   – <CABI Installed Folder>/apache-tomcat/webapps/jasperserver-pro/optimized-scripts.

6. (Optional) Open applicationContext.xml file that is located in **<CABI Installed Folder>\apache-tomcat\webapps \jasperserver-pro\WEB-INF folder**, search for **id="userTimeZonesList"** and remove the time zone attributes from **<value>Asia/Kolkata</value>** to **<value>Etc/GMT-14</value>**

```
<value>Asia/Kolkata</value>

<value>Asia/Shanghai</value>

.

.

.

<value>Pacific/Kiritimati</value>

<value>Etc/GMT-14</value>
```

7. (Optional) Open validation.properties file that is located in <CABI Installed Folder>/apache-tomcat/webapps/ jasperserver-pro/WEB-INF/classes/esapi folder and update Validator.ValidSQL property as follows:

```
Validator.ValidSQL= (?is)^\\s*(select|call)\\s+[^;]+;?\\s*$
```

8. Navigate to **<SOI_Home>\SamUI\webapps\sam\ui** folder, and paste the loadTPA.jsp file that you have taken as a backup after the installation of CA SOI reports.
9. Restart CA Business Intelligence Tomcat Server service.

   NOTE

   You can place the loadTPA.jsp(backup file) in **<SOI_Home>\SamUI\webapps\sam\ui** folder. Ensure that you restart the UI Server after placing the loadTPA.jsp file.

Performance Results of CA SOI Dashboards

This article provides the performance results of CA SOI Dashboards.

The environment details are as follows:

Infrastructure Details		
	CA SOI	**CABI JasperReports Server for Unified Dashboards and Reporting for Infrastructure Management**
System Type	Virtual Machines	Virtual Machines
Processor on All Virtual Machines	Intel® Xeon® CPU E5-2660 v3 @ 2.6-GHz	Intel® Xeon® CPU E5-2680 v3 @ 2.8-GHz

Operating System	Windows Server 2012 R2 Standard 64-bit OS	Windows Server 2008 R2 Enterprise Service Pack 1 64-bit OS

System Usage	CA SOI			**CABI JasperReports Server for Unified Dashboards and Reporting for Infrastructure Management**		
	CPU	**Memory**	**Disk Space**	**CPU**	**Memory**	**Disk Space**
Database	8	8 GB	100 GB	N/A	N/A	N/A
UI Server	8	8 GB	100 GB	N/A	N/A	N/A
SA Manager	8	8 GB	100 GB	N/A	N/A	N/A
IFW	8	8 GB	100 GB	N/A	N/A	N/A
CABI JasperReports Server for Unified Dashboards and Reporting for Infrastructure Management	N/A	N/A	N/A	8	16 GB	100 GB

Total Number of Created Alerts, Alerts Queues, Services, and CIs Count	
Alerts	186000
Alerts Queue	6
Services	8
CIs	195500

For the mentioned environment details the results are as follows:

Dashboard	Operation	Alerts Count	Loading of Dashlets for the First Time	Loading of Dashlets for the Second Time
Alerts View	Loading of Alert Queue KPI section	10000	4 seconds	4 seconds
	Loading of Acknowledgement Details section	10000	3 seconds	3 seconds
	Loading of Alerts Section	10000	15 seconds	15 seconds
Business Service View	Loading of Business Service View section	10000	3 seconds	3 seconds
	Loading of Alerts per Business Service section	10000	20 seconds	20 seconds

Post-Installation Configuration and Customization

Once you complete the product installations, you can configure the following features:

- help desk integration
- Google Earth integration
- failure email notifications

You can customize the following features:

- PC dashboard
- mobile dashboard
- operations console
- connectors
- passwords
- communication ports

For complete configuration and customization information, see General Administration.

Help Desk Integrations

This section describes how to implement integrations with help desk products for alert escalation and incident management.

How to Configure a BMC Remedy Integration

As an administrator, you can integrate CA SOI with BMC Remedy through CA Process Automation, the connectors, and the provided gateway files that enable end-to-end integration workflows.

Install a CA Process Automation server with the required third-party components and prerequisites to prepare for help desk integration with BMC Remedy. CA Process Automation must reside on a separate server from CA SOI and BMC Remedy.

> **WARNING**
> Verify that CA SOI and BMC Remedy are installed before installing CA Process Automation.

This section provides details pertaining to installing CA Process Automation for integrating CA SOI with BMC Remedy. For more detailed information about installing CA Process Automation, see the *CA Process Automation Installation Guide*.

CA Process Automation provides a centralized and structured approach to operations management. CA Process Automation supports a fully integrated development environment and supports client applications. The applications let you schedule, start, and monitor automated processes.

The CA Process Automation Remedy connector, using underlying CA Process Automation technology, links alert data that CA SOI generates to BMC Remedy. You can use the CA Process Automation Remedy connector to perform the following actions:

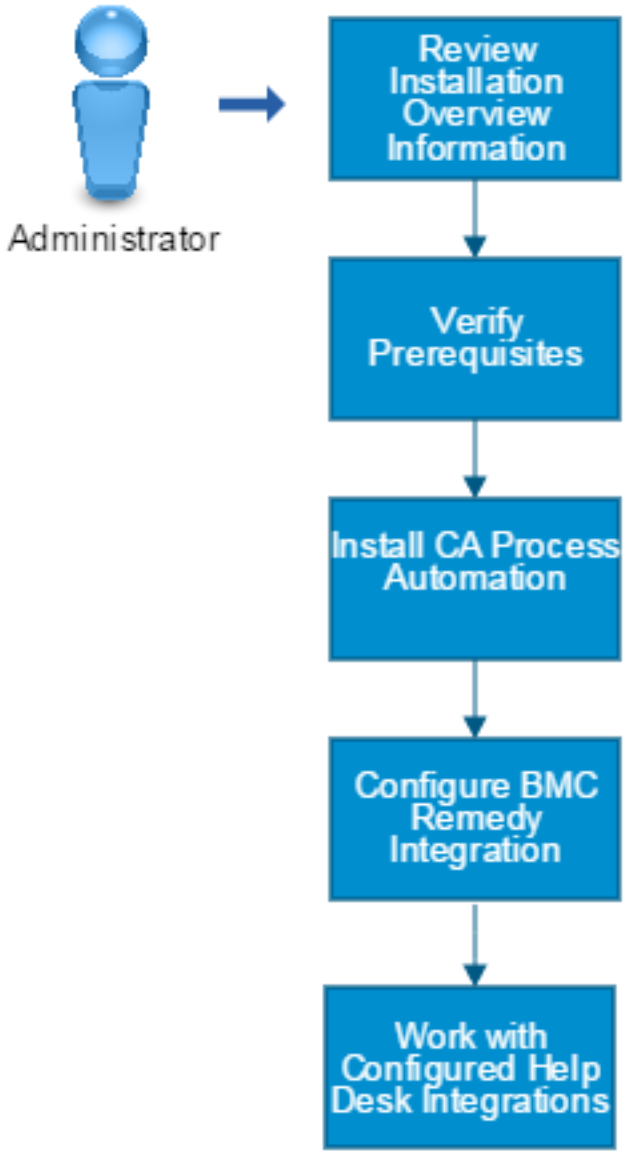
- Manually generate a BMC Remedy incident that is based on an alert in the Operations Console
- Automatically generate an incident using alert escalation
- View a generated incident from the Operations Console
- Implement the communication between CA SOI and BMC Remedy, so that when an alert is cleared, the associated BMC Remedy incident is closed

> **NOTE**
>
> The BMC Remedy integration supportscreation of BMC Remedy incidents only as the ticket type, not problems or requests.

The installation overviews in this section require the CA Process Automation installation media. Download the installation media from CA Support.

Use this scenario to guide you through the process:

How to Configure a BMC Remedy Integration



1. Review the installation overview information:
   - CA Process Automation and BMC Remedy terminology.
   - Component overview.
   - Integration architecture.
   - Software requirements.
   - Installation requirements.
2. Verify the prerequisites.
3. Install CA Process Automation.

4. Configure the BMC Remedy integration.
5. Work with the configured help desk integration.

BMC Remedy Integration Preparation

Contents

Review the topics in this section before starting the BMC Remedy integration.

CA Process Automation and BMC Remedy Terminology

The following list describes commonly used terms that are associated with CA Process Automation and BMC Remedy:

> **NOTE**
> For more information about the BMC Remedy terms in this list, see the BMC Remedy documentation.

- **CA Process Automation Orchestrator**
  The CA Process Automation Orchestrator is a CA Process Automation server that runs modules, schedules and runs processes, maintains state information, and manages authentication. Because one CA Process Automation server is required to integrate with BMC Remedy, the terms CA Process Automation Orchestrator and CA Process Automation server refer to the same server in these procedures.

  > **NOTE**
  > For more information about CA Process Automation architecture and terminology, see the *CA Process Automation Installation Guide*.
- **Remedy Incident**
  A Remedy incident is an event-tracking mechanism in BMC Remedy. An incident describes a Help Desk Case opened with the Case Type of Incident, which is used for any abnormal service-operational event. When used in the context of the CA Process Automation Remedy connector, an incident refers to the Help Desk Case opened for a CA SOI alert. Within CA SOI, a Trouble Ticket ID value tracks the BMC Remedy incident that is created for an alert. Therefore, the terms incident and ticket can be used interchangeably.
- **Remedy Administrator Tool**
  The Remedy Administrator Tool is a program that is used to make administrative changes to BMC Remedy.
- **Remedy Mid-Tier Configuration Tool**
  The Remedy Mid-Tier Configuration Tool is a program that is used to view and configure BMC Remedy Mid-Tier system settings.
- **Remedy User Tool**
  The Remedy User Tool is a program that is used with BMC Remedy to make the user level changes on how the product functions.
- **ITSM**
  Refers to the ITIL version of BMC Remedy.
- **HelpDesk**
  Refers to the non-ITIL version of BMC Remedy.

Component Overview

The following list describes the components that are required for the integration between CA SOI and BMC Remedy. The list also provides information about where to find these components and where to install or implement the components.

- **CA Process Automation Installation Media**
  Provides required third-party components, the CA Process Automation product, and CA Process Automation documentation. Install CA Process Automation on a separate server.
- **EEMServer_8.4.100.0***

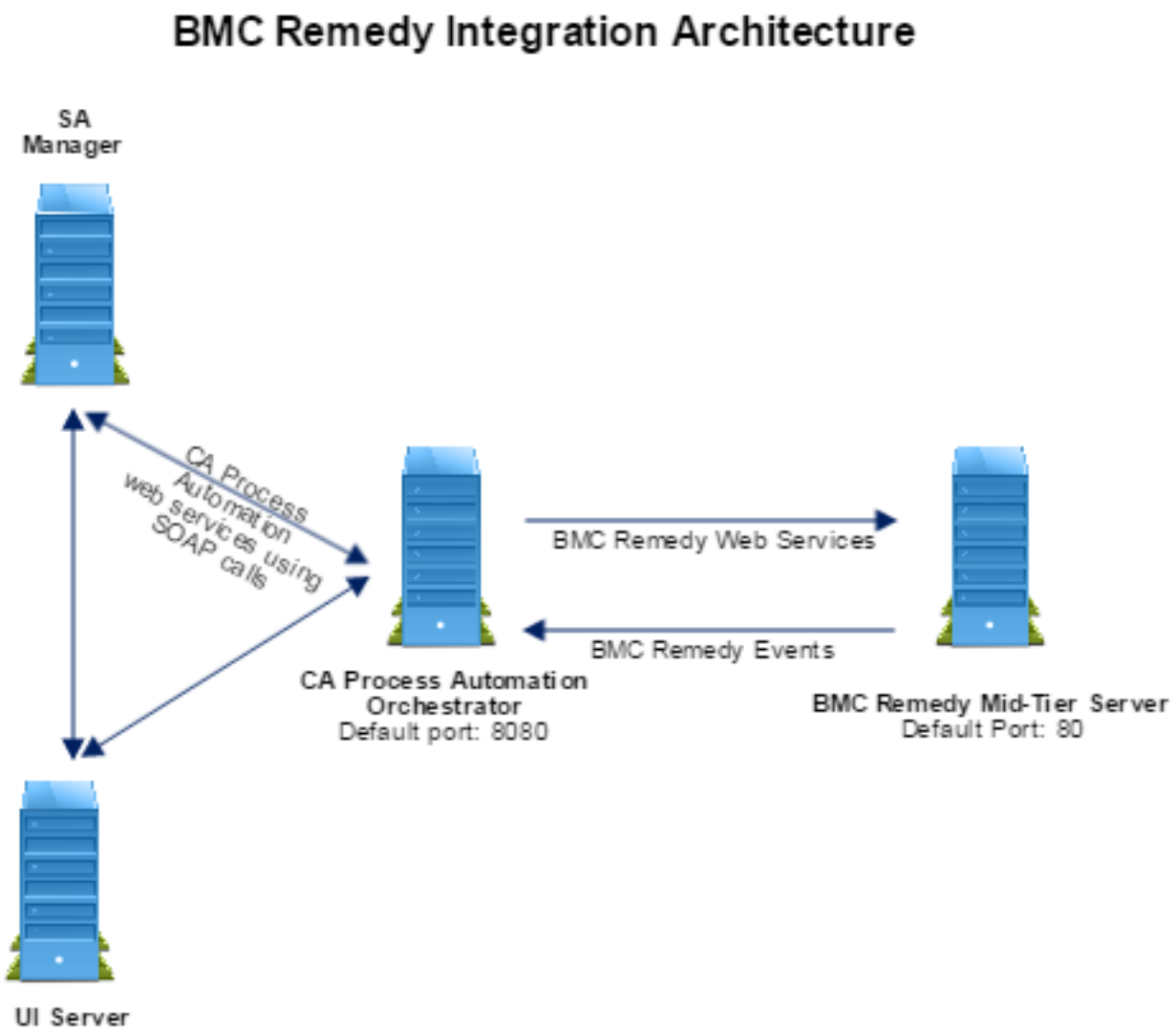
Provides CA EEM for all supported platforms. CA EEM is required if you want to use it to authenticate CA Process Automation users. You can find the CA EEM installer on disk 2 of the CA Process Automation installation media. However, we recommend using the existing CA EEM server that is used with CA SOI for CA Process Automation authentication.

- **ITPAM_eem.xml**
  Provides CA Process Automation-specific definitions for CA EEM. Find this file on disk 2 of the CA Process Automation installation media. This file belongs where CA EEM is installed.
- **ITPAM_Server_Files**
  Provides the CA Remedy Gateway, which is the set of CA SOI process definitions for CA Process Automation. Find this directory on the CA SOI installation media at Disk1\Integrations\ITPAM-Remedy. The files in this directory belong on the CA Process Automation server, or the location where the CA Process Automation JNLP Client runs.
- **Remedy_Server_Files**
  Provides web services definitions and other configuration information for BMC Remedy. Find this directory on the CA SOI installation media at Disk1\Integrations\ITPAM-Remedy. The files in this directory belong on the BMC Remedy AR System Server or the location where the BMC Remedy Administrator runs.
- **InstallCerts.zip**
  Provides the tool to install an SSL certificate for the Remedy Mid-Tier Server on CA Process Automation. Find this file on the CA SOI installation media at Disk1\Integrations\ITPAM-Remedy\Supporting Remedy Mid-Tier on SSL. This file belongs on the CA Process Automation server.

Integration Architecture

Implementing the integration between CA SOI and BMC Remedy configures communication among CA SOI, CA Process Automation, and BMC Remedy. Each product has its own server and interfaces. The following diagram shows the communication that is required between the products, including protocols and port information:

Figure 13: BMC Remedy Integration Arch



NOTE
This diagram assumes that the BMC Remedy AR System Server is on the same system as the Mid-Tier Server. The AR System Server can also reside on a separate system.

This section describes the installation requirements for configuring the integration, firewall considerations for enabling communication among the servers, and recommended configuration information.

Software Requirements

You must have the following product versions for the BMC Remedy integration to work properly:

- CA Process Automation r3.1, r4.0, or r4.1

  NOTE
  CA Process Automation r2.1, r2.2, r2.2 SP1, and r3.0 are no longer supported with this integration and CA SOI r3.2. You can attempt to use these versions. If the integration fails, upgrade CA Process Automation to r3.1, r4.0, or r4.1.

CA Process Automation r3.1, r4.0, or r4.1 integrations with BMC Remedy are supported using either of the following configurations:

CA Process Automation Player
Provides the CA Process Automation product with predefined workflows.

Full CA Process Automation product
Provides full product functionality, including the ability to define custom workflows. A full product license is required to integrate with CA Process Automation for escalation action workflows.

- CA Catalyst Connector for BMC Atrium/Remedy, which is available on CA Support.
- CA SOI r3.2
- BMC Remedy AR System 6.3, 7.0.x, 7.1, 7.6, 7.6.4, or r8.0 with ITIL or non-ITIL based HelpDesk
  The HPD:HelpDesk form is required for non-ITIL based Remedy, and the HPD:IncidentInterface_Create form is required for the ITIL-based Remedy versions. View your BMC Remedy Administrator interface to find this form.

Installation Requirements

Integrating CA SOI and BMC Remedy through CA Process Automation requires the installation of the following products on separate servers:

- CA SOI
  Install CA SOI according to the instructions.
- **BMC Remedy Mid-Tier Server and AR System Server**
  Install these BMC Remedy components on a separate system from CA SOI or CA Process Automation. The AR System Server and Mid-Tier Server can reside together on the same or on separate systems. Perform configuration steps in BMC Remedy after installation to enable integration with CA Process Automation. If you use custom fields on your BMC Remedy form, additional configuration is required to support these fields. For more information about installing BMC Remedy, see the BMC Remedy documentation.

  NOTE
  For more information about configuring BMC Remedy to integrate with CA Process Automation, see Configure BMC Remedy Web Services and Filters. For more information about configuring custom fields, see How to Customize BMC Remedy Ticket Fields with CA SOI Alert Parameters.

  WARNING
  CA SOI and BMC Remedy must be installed before installing CA Process Automation.

- **CA Process Automation**
  Install CA Process Automation on a separate, dedicated, physical, or virtual server. Installing CA Process Automation on the same server as CA SOI or BMC Remedy is not recommended.
  CA Process Automation is the core component for processing and synchronizing events between CA SOI and BMC Remedy. CA Process Automation provides support for scalability of up to hundreds of alerts at a time using a dedicated CA Process Automation system. Install the CA Process Automation Remedy connector with CA Process Automation.
  Note: For more information about CA Process Automation system requirements, see the CA Process Automation documentation.
- **CA EEM**
  Use the CA EEM server for use with CA SOI to provide authentication and authorization for CA Process Automation. You can install a dedicated CA EEM on a separate server or on the same server as CA Process Automation. However, we recommend sharing the CA EEM version that is already installed with CA SOI.

Verify Prerequisites

CA Process Automation has the following prerequisites:

- Installation media for CA Process Automation r3.1, r4.0, or r4.1.
- Java SE Development Kit (JDK) 6

NOTE
CA Process Automation requires Java SE Development Kit (JDK) 6 specifically. On the Java SE Downloads page, ensure that you select the download that is labeled Java SE Development Kit (JDK). To download the JDK and for more information about Java, see http://java.sun.com/.

- Microsoft SQL Server 2005 and higher
  We recommend that you use the same database that you used with the CA SOI installation. However, you can use a separate database on the CA Process Automation server or a separate server. The database must meet the following requirements:
  – TCP/IP protocol must be enabled.
  – If you are using an instance other than the default, the instance must be running on a different port.
  – A JDBC driver is required for the database. The CA Process Automation installation selects the appropriate driver.

  NOTE
  For more information about database server prerequisites, see the *CA Process Automation Installation Guide*.

- CA EEM
  We recommend that you use the same CA EEM server that you used with the CA SOI installation. However, you can use a separate installation on the CA Process Automation server or a separate server. For more information about configuring CA EEM, see Configure CA EEM.

Install CA Process Automation for BMC Remedy

Contents

This section summarizes the process of installing the CA Process Automation server with its required third-party components and prerequisites.

This section provides details pertaining to installing CA Process Automation for integrating CA SOI with BMC Remedy. For more information about installing CA Process Automation, see the *CA Process Automation Installation Guide*.

The installation overviews in this section require the CA Process Automation installation media. Download the installation media from CA Support Online.

CA Process Automation Firewall Considerations

To enable communication across the multiserver configuration that is required for the BMC Remedy integration, the following connections must be enabled:

- CA Process Automation server to the Remedy Mid-Tier server on the BMC Remedy HTTP port (usually 80)
- SA Manager to CA Process Automation server on the CA Process Automation port (8080 by default)

Verify User Availability

If the CA EEM installation is using an external datastore (such as Active Directory) instead of the internal datastore, verify that the pamuser and pamadmin users are also available in the EEM ITPAM Application. These users must be a part of the PAMUsers group, and admin users must be a part of the PAMAdmins group.

Follow these steps:

1. Select Start, Programs, CA, Embedded Entitlements Manager, EEM UI.
2. Enter the Eiamadmin credentials.
3. Select ITPAM from the Application drop-down list.
4. Click Log In.
5. Click Manage Identities.

6. Click Go using the default entries.
   The or pamuser and or pamadmin user names display under Users in the Users pane if the names were created correctly. The default passwords are the same as the user names.

   NOTE
   Select theuser name to change any user settings. For more information about user settings, see the CA EEM documentation.
7. Select Groups.
8. Select Show application groups and click Go.
   The PAMAdmins and PAMUsers groups display under Application Groups in the User Groups pane if the groups were created correctly.

Install Third-Party Components

The CA Process Automation installation media provides the following third-party components, which are required before you configure integration with BMC Remedy or HP Service Manager:

- Jboss
- Hibernate
- JDBC driver

To install third-party components, run Third_Party_Installer_windows.exe from the CA Process Automation installation media.

For a detailed installation procedure, see the *CA Process Automation Installation Guide*.

Adhere to the following general guidelines:

- You can accept the default installation directories for Jboss and Hibernate.
- On the JDBC Jars Installation page, click Add Files and select MS SQL 2005. The appropriate JDBC driver displays and installs after you click Next.
- You can skip the TAPI installation; this component is not required for a CA Process Automation installation to integrate with BMC Remedy or HP Service Manager.

Install the Domain Orchestrator

Adhere to the following general guidelines to install the CA Process Automation server, which is also known as the Domain Orchestrator:

NOTE
This list describes only the configuration options specific to requirements for the CA SOI and BMC Remedy or HP Service Manager integration. For more information about a full installation, see the *CA Process Automation Installation Guide*.

- All necessary third-party components must be installed before you begin the Domain Orchestrator installation.
- Install CA Process Automation on a dedicated physical or virtual server.
- The installing user must have Administrator privileges on Windows.
- Take note of the settings that you enter on the General Properties page, especially the HTTP Port, which is how CA SOI communicates with CA Process Automation (8080 by default). All ports must be uniquely assigned on the server.

  NOTE
  For information about changing port numbers that you defined during installation, see the *CA Process Automation Installation Guide*.
- We recommend selecting the Install as Windows Service check box to run the Orchestrator as a service.
- Select EEM as the Security Server on the Select Security Server Type page. On the ensuing EEM Security Settings page, enter the CA EEM server name that you configured for CA Process Automation, enter ITPAM for the EEM

Application Name, navigate to the location of the itpamcert.p12 file in the EEM Certificate File field, and enter the EEM Certificate Password as specified in the ITPAM_eem.xml file (itpamcertpass by default).

- You can test the CA EEM settings to ensure that CA EEM is configured correctly. Click Test EEM Settings and enter pamadmin for the user name and password (if you retained the default). A window opens showing the result of the test.
- Do the following on the Database Settings page:
  - Specify MSSQL 2005 in the Type of Database drop-down list and enter the appropriate database settings. We recommend sharing the database used by CA SOI.
  - To connect to an instance other than the default, specify only the server name in the Database Server field (without the instance name), and specify the port where the instance is listening in the Database Port field.
  - You can click Test Database Settings to test the database connection. If a message indicates that the CA Process Automation database is missing, close the dialog and click Create Database to create the database.

Start the Domain Orchestrator

After the installation completes, start the CA Process Automation Orchestrator service before configuring the integration.

To start the Domain Orchestrator, perform one of the following actions:

- (CA Process Automation r3.1) Select Start, Programs, CA, CA Process Automation Domain, Start Orchestrator Service.
- (CA Process Automation r4.0 and r4.1) Open the Windows Services dialog and start the CA Process Automation Orchestrator service.
  The CA Process Automation Orchestrator service is configured for manual startup by default. You can change this setting to automatic in the service properties to configure the service to start automatically when Windows starts.

How to Configure BMC Remedy Integration Components

This section describes how to install and configure the CA Process Automation Remedy connector and CA Remedy Gateway files to integrate CA SOI with BMC Remedy.

To configure the BMC Remedy integration, follow this process:

1. Start the CA Process Automation Management Console.
2. Configure the processes for the BMC Remedy integration.
3. Reinstall the connectors.
4. Configure the CA SOI CA Process Automation connection.
5. Configure the SSL connection with CA Process Automation.
6. Configure the BMC Remedy web services and filters.
7. Install the SSL certificate for BMC Remedy and mid-tier server.
8. Configure and Test the BMC Remedy mid-tier server.
9. Configure and test the BMC Remedy connection.
10. Configure and test the CA SOI connection.
11. Test the BMC Remedy integration.
12. Verify the web context URL link.

Start the CA Process Automation Management Console

The CA Process Automation Management Console is the web-based console that you use to manage the processes behind the integration with BMC Remedy.

Follow these steps:

1. Perform one of the following actions:

- (CA Process Automation r3.1) Select Start, Programs, CA, CA Process Automation Domain, Start CA Process Automation Client on the CA Process Automation server.
- (CA Process Automation r4.0 and r4.1) Enter the following URL in a web browser from any server:

  ```
  http://<hostname>:<port>/itpam
  ```

  Use the CA Process Automation server for the host name, and the HTTP port for the port setting (8080 by default). Refer to the Installation Worksheet CA Process Automation section for this information.

  NOTE
  If CA Process Automation is not available, verify that the CA Process Automation Server service is running.

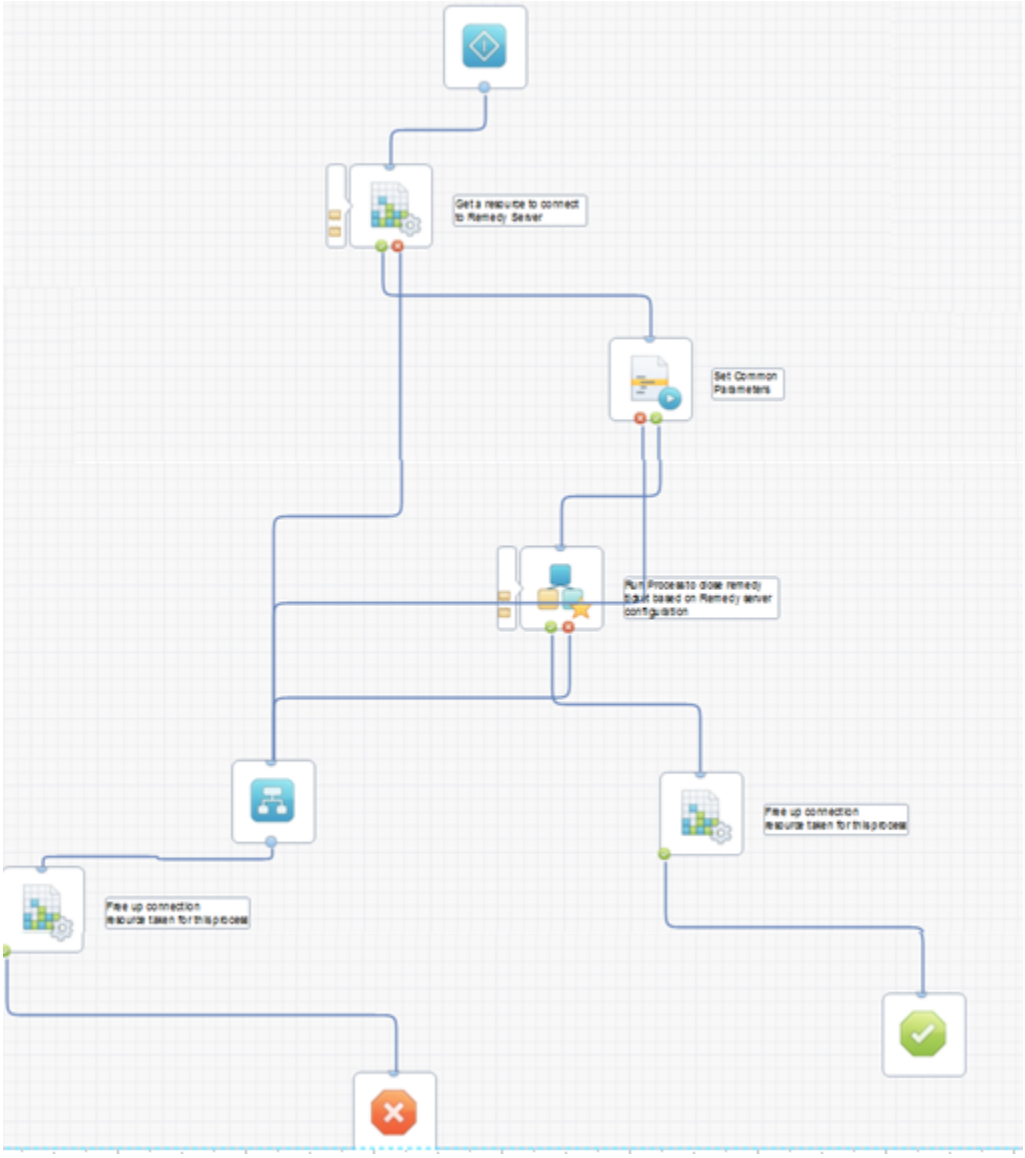
2. Enter a valid user name and password and click Log In.

Configure Processes for BMC Remedy Integration

You configure the processes by which CA Process Automation communicates with CA SOI and BMC Remedy. This configuration requires the CA Remedy Gateway.xml and ITPAM-Remedy_Custom Operators.xml files included with the CA SOI installation media.

Follow these steps:

1. Follow these steps for CA Process Automation r3.1 or go to Step 2 for CA Process Automation r4.0/4.1:
   a. Click CA Process Automation Client at the upper right of the CA Process Automation Management Console.
   A JNLP application starts and the Process Automation Client displays.
   b. Select File, Open Configuration Browser.
   c. Expand Default Environment in the Browser pane and double-click Orchestrator.
   The Folders pane displays on the left and an empty Environment pane displays on the right.
   d. Continue with Step 3.
2. Click the Library tab.
3. Right-click the root folder in the Folders pane and select Import.
   A page opens for selecting a file to import.
4. Locate the CA Remedy Gateway.xml file on the CA SOI installation media at Disk1\Integrations\ITPAM-Remedy \ITPAM_Server_Files and click Open.
   The Import Object page displays.
5. Select the Set imported version as current check box to make the object current and click OK.
   The processes are imported and the CA Remedy Gateway folder displays in the Folders pane.
6. Repeat Steps 3-5 and import the ITPAM-Remedy_Custom Operators.xml file located on the CA SOI installation media at Disk1\Integrations\ITPAM-Remedy\ITPAM_Server_Files.
   The Custom Operators folder displays in the Folders pane.
7. Select the CA Remedy Gateway folder.
   The imported integration processes display in the Default Environment pane.
8. Double-click the \CA Remedy Gateway\ITSM\CreateITSMTicket (for the ITIL Remedy version) or the \CA Remedy Gateway\HelpDesk\CreateHelpDeskTicket (for the non-ITIL Remedy version) process and click Edit User Preferences at the bottom-left of the screen..
9. Select all options except for "Default to horizontal line orientation" on the Configuration tab, select the Palette Properties tab and enter 12, and click OK.
10. The process for CA Process Automation r4.0/4.1 looks similar to the following graphic. The functionality of the CA Process Automation r3.1 flow is the same but the icons are different.



NOTE
If any

icons display in the process, the required CA Process Automation Remedy connector is not installed properly. For more information about reinstalling connectors, see Reinstall Connectors.

Reinstall Connectors

If an icon displays anywhere in the imported CA SOI-BMC Remedy integration processes, the CA Process Automation Remedy connector is not installed correctly. You must reinstall the nonfunctioning connector before continuing the configuration.

Follow these steps:

1. Close the CA Process Automation Client and CA Process Automation Management Console, and stop the CA Process Automation Orchestrator service.
2. Run the CA_ITPAM_Domain program file to start the installation.
3. Perform one of the following actions:
   – Use the Reinstall option on the ITPAM Domain Reinstall/Configure page.
   – When prompted for the connectors to install or update, ensure that the connector corresponding to the icon is unchecked. The CreateRemedyTicket process uses the CA Process Automation Remedy Connector.

   The installation runs and completes.
4. Start and run the installation again and perform the following actions:
   a. Use the Reinstall option on the ITPAM Domain Reinstall/Configure page.
   b. When prompted for the connectors to install or update, ensure that the connector that you unchecked in the last step is now selected.
5. Start the CA Process Automation Orchestrator service.
6. Start the CA Process Automation Management Console.
7. Access the CreateRemedyTicket process as described in Configure Processes for BMC Remedy Integration.
8. Verify that no icons display in the process.

If any icons display after reinstalling connectors, contact Technical Support.

Configure CA SOI-CA Process Automation Connection

CA SOI connects to CA Process Automation, which in turn connects to BMC Remedy. To connect to CA Process Automation, enter the CA Process Automation server information in the CA SOI Administration UI.

Follow these steps:

1. Click the CA SOI Administration tab on the Dashboard.
2. Click the Administration tab, expand CA Service Operations Insight Manager Configuration and the server name and click Help Desk Configuration.
   The Help Desk Configuration page opens.
3. Select BMC Remedy IT Service Management Suite in the Help Desk Type drop-down list.
4. Enter the CA Process Automation server properties in the ITPAM/ Remedy Gateway Server Details pane, and click Save.
5. Click Test to test the connection.
   The CA Process Automation server information is saved in CA SOI.

   NOTE
   If you changed an existing Help Desk Configuration setting, restart the CA SAM Application Manager service for the change to take effect. A change includes a CA Service Desk or a different BMC Remedy server integration.

Configure SSL Connection with CA Process Automation

For CA SOI to communicate with a CA Process Automation server that has been configured to use SSL, you must import a certificate into the CA SOI trust store.

Follow these steps:

1. Copy the itpamcertificate.cer file from the following location on the CA Process Automation server to a directory on your SA Manager server:
   ```
   <PA_HOME>\ITPAM\server\c20\.c20repository
   ```
2. Make a backup copy of the <SOI_HOME>\tomcat\conf\ssa.jks file.
3. Run the following command from a command prompt on the SA Manager system:
   ```
   "<JAVA_HOME>\bin\keytool.exe" -v -importcert -storepass <password> -file <DIR>\itpamcertificate.cer -
   keystore "<SOI_HOME>\tomcat\conf\ssa.jks" -trustcacerts -noprompt
   ```
   - **password**
     Defines the password for the CA SOI administrator user.
   - **DIR**
     Defines the path to the directory to which you copied the itpamcertificate.cer file.

   The CA Process Automation certificate is imported into CA SOI.
4. Restart the CA SAM Application Server service on the SA Manager.
5. Configure CA Process Automation integration in the Administration UI. Select the SSL check box and use the SSL port number.
6. Click Test.
   A message confirms that the SSL connection was successful.

Configure BMC Remedy Web Services and Filters

You must install specific CA Process Automation definition files on the BMC Remedy AR System Server for use by the BMC Remedy web services so that the two products can communicate. This procedure requires the files provided on the CA SOI installation media at Disk1\Integrations\ITPAM-Remedy\Remedy_Server_Files

Follow these steps:

1. Copy all of the files from the CA SOI installation media directory Disk1\Integrations\ITPAM-Remedy \Remedy_Server_Files into a directory on the BMC Remedy AR System Server, and note the location of the files for future reference.
2. Locate and run ConfigITPAM.bat (for Windows) or ConfigIITPAM.sh (for Linux) on the BMC Remedy system.
   The CA Spectrum Service Assurance / Remedy Integration Kit dialog opens.
3. Click the Configure Paths tab and complete the following fields:
   - **Installation Directory**
     Defines the directory where you copied the BMC Remedy server files.
   - **JAVA_HOME**
     Defines the JAVA_HOME location.
4. Click the Configure IT-PAM tab and complete all fields to connect to the CA Process Automation system.
5. Click Save, then click Test.
6. Select one of the following from the Start menu depending on your BMC Remedy version:
   - Programs, Action Request System, BMC Remedy Administrator
   - Programs, BMC Software, AR System 1, BMC Remedy Developer Studio

   A log in page opens.
7. Enter a valid user name and password and click Login.
   The BMC Remedy Administrator page or Developer Studio page opens.
8. Do one of the following depending on your BMC Remedy version:

- Versions below 7.6: Expand the folder for the Remedy Mid-Tier server under the Servers node and select Web Services.
- Version 7.6: Expand the folder for the Remedy Mid-Tier server, expand All Objects, and double-click Web Services.

9. Do one of the following depending on your BMC Remedy version:
   - Versions below 7.6: Select Import Definitions, From Definition File from the Tools menu.
   - Version 7.6: Select File, Import. In the dialog that appears, expand BMC Remedy Developer's Studio, select Object Definitions, and click Next. Select your server and click Next.

   A file selection page opens.
10. Locate and select the file ITPAM_<Remedyversion>_WS.def and click Open. The Remedyversion is HelpDesk if you are using a non-ITIL version of BMC Remedy or ITSM if you are using an ITIL version of BMC Remedy.
    The Import Definitions page opens (versions below 7.6), or the file appears in the Import Objects dialog (version 7.6).
11. Do one of the following depending on your BMC Remedy version:
    - Versions below 7.6: Select Web Services from the Available Objects pane and click Add>>>. After the item moves to the Objects to Import pane, select the Replace Objects on the Destination Server check box and click Import.
    - Version 7.6: Click Next. On the Object Selection pane, select all available objects and click Add. Select the Replace Objects on the Destination Server check box and click Finish.

    The Import complete page opens.
12. Close the Import Definitions window.
13. Repeat Steps 8-12 for the ITPAM_<Remedyversion>_Trigger_Filters.xml file.
14. Select one of the following from the Start menu:
    - Versions below 7.6: Programs, Action Request System, BMC Remedy Mid Tier, Configure ARSYSTEM
    - Version 7.6: Programs, BMC Software, AR System 1, BMC Remedy Mid Tier, Configure Mid Tier on Localhost

    A log in page opens.
15. Enter a valid password and click Login.
    The Remedy Configuration Tool page opens.
16. Select Cache Settings and click Flush Cache.
    The cache is updated, which helps ensure that the CA Process Automation definition files are used.

View BMC Remedy Web Services

After you successfully import the CA Process Automation web services definition files on the BMC Remedy AR System Server, you can display the BMC Remedy Web Services WSDL.

Follow these steps:

1. Enter the following URLs depending on your BMC Remedy version:

   ```
   http://<ar_server>:<port>/arsys/WSDL/public/<ar_server>/ITPAM_<RemedyVersion>_*
   ```

   > **NOTE**
   > If possible, test the URL on a CA Process Automation server browser to verify that the BMC Remedy server and port is accessible from the CA Process Automation server.

   - **ar_server:port**
     Specifies the BMC Remedy AR System Server host name or IP address and port. If the port is omitted, it defaults to 80.

     > **NOTE**
     > Take note of these values, because they are required when configuring the BMC Remedy server in CA Process Automation.
   - **ar_server**
     Specifies the BMC Remedy AR System server name.

NOTE
Record this value because it is required when configuring the BMC Remedy server in CA Process Automation.

– **ITPAM_RemedyVersion_***
Specifies the BMC Remedy version and the desired action. The value can be *one* of the following actions:
- **ITPAM_HelpDesk_Modify_Service**
Lets you modify information from the non-ITIL version of BMC Remedy.
- **ITPAM_HelpDesk_Query_Service**
Lets you query information from the non-ITIL version of BMC Remedy.
- **ITPAM_HelpDesk_Submit_Service**
Lets you create information from the non-ITIL version of BMC Remedy.
- **ITPAM_HPD_HelpDesk_WS**
Lets you modify information from the ITIL version of BMC Remedy.
- **ITPAM_HPD_IncidentInterface_WS**
Lets you query information from the ITIL version of BMC Remedy.
- **ITPAM_HPD_IncidentInterface_Create_WS**
Lets you create information from the ITIL version of BMC Remedy.

2. Test the URL on a CA Process Automation server browser to verify that the BMC Remedy server and port are accessible from the CA Process Automation server.

Install SSL Certificate for BMC Remedy Mid-Tier Server

If your BMC Remedy Mid-Tier server is running on SSL, install an SSL certificate on the CA Process Automation server. This procedure requires the InstallCerts.jar file that is on the CA SOI installation media at Disk1\ITPAM-Remedy \Supporting Remedy Mid-Tier On SSL\InstallCerts.zip.

To determine if BMC Remedy Mid-Tier is running on SSL, enter the following URL in a browser, using the BMC Remedy Mid-Tier server name and port:

```
https://<host_name>:<port>/arsys/shared/login.jsp
```

If a login screen opens, the server is running on SSL. Install an SSL certificate if non-SSL access is blocked on the Mid-Tier server or you want to configure a secure communication between CA Process Automation and BMC Remedy. If you enter the URL and a login screen opens, then the server is running on SSL. Otherwise, install the certificate.

Follow these steps:

1. Extract the files in InstallCerts.jar onto the CA Process Automation server.
2. Run the following command:
```
Java -jar installcert.jar <host_name>[:port]
```
– **host_name**
Specifies the name of the BMC Remedy Mid-Tier server.
– **port**
(Optional) Specifies the port that the BMC Remedy Mid-Tier server uses.

NOTE
Do not use a passphrase.

The following message appears:
```
Enter certificate to add to trusted keystore or q to quit: Type 1 and press Enter.
```
The command creates a file named jssecacerts in the same folder.
3. Copy the itpamcert file to the <jdk_home>\jre1.6.0_0x\lib\security folder on the CA Process Automation server.
4. Open the c2osvcw.conf file that is at <itpam_home>\server\c2o\bin and add the following two lines to the end of the file:

```
wrapper.java.additional.XX=-Djavax.net.ssl.truststore=
"<jdk_home>\jdk.1.6.0.0N\jre\lib\security\jssecacerts"
wrapper.java.additional.XX=-Djavax.net.ssl.trustStorePassword="changeit"
```

– **XX**
  Specifies the next available wrapper.java.additional number in the file. For example, if wrapper.java.additional.10 is the last number that is defined in the file, use 11 and 12 for these entries.
– **jdk_home**
  Specifies the location of the JDK that the CA Process Automation server is running. The *N* variable in the JDK version is the version indicator.

5. Keep changeit as the password and restart the CA Process Automation server.
   CA Process Automation is configured to connect to BMC Remedy using SSL.

Configure and Test BMC Remedy Connection

This section describes how to define the BMC Remedy Mid-Tier server to the CA Process Automation system and test the connection between the products. You define separate properties for the ITIL and non-ITIL versions of BMC Remedy.

Configure Non-ITIL BMC Remedy Server Settings

You configure the non-ITIL BMC Remedy server settings within CA Process Automation to enable the connection between the two products.

Follow these steps:

1. Start the CA Process Automation Management Console.
2. Click User Requests.
   The User Requests pane opens.

   NOTE
   If the pane does not open, recycle the console. Changes may not have been applied from a previous operation.
3. Expand CA Remedy Gateway, right-click RemedyHPDConfiguration, and select Start Request.
   The Remedy Parameters pane opens on the right. This pane contains fields that are mandatory for incident creation.
4. Enter values for the following fields as described. All fields are required, but some provide default values.

   WARNING
   Some fields such as Category, Item, and Type only accept specific values. For example, the Priority field can only be Urgent, High, Medium, or Low. If you enter an invalid value, the incident is not created. For more information about valid entries, see the BMC Remedy AR System documentation.

   NOTE
   An asterisk at the beginning of the name indicates that you can input an expression (in JavaScript). Static strings are double-quoted.

   – **Remedy Mid-Tier Server**

     Specifies the BMC Remedy Mid-Tier server host name or IP address and port in the following format: *server:port*. Use the following URL to test this value: http://*server:port*/arsys/home.

     If you omit the port number, the port number defaults to 80.
   – **Remedy Mid-Tier Server is configured to use SSL**

     Specifies whether the BMC Remedy Mid-Tier server is configured to use SSL.

NOTE
If the server is configured to use SSL, ensure that you have installed the SSL certificate on the CA Process Automation server. For more information about installing the SSL certificate, see Install SSL Certificate for Remedy Mid-Tier Server.

- **Remedy Server Name**
  Specifies the BMC Remedy AR System server name. Use the following URL to test this value: http://<midtier_server>:<port>/arsys/WSDL/public/<ar_server>/ITPAM_HelpDesk_Modify_Service.
- **Remedy Username**
  Specifies the user name to log in to BMC Remedy.
- **Remedy Password**
  Specifies a valid password for the BMC Remedy user name.
- **Maximum concurrent connections to Remedy**
  Defines the number of simultaneous BMC Remedy connections allowed.
  Default: 10
- ***Remedy Requester Login**
  Specifies the BMC Remedy incident requester login. This value is used as the login value in the generated BMC Remedy incident.
- ***Remedy Requester Name**
  Specifies the full name of the BMC Remedy requester. This value is used as the Requester Name in the generated BMC Remedy incident. The name must exist in BMC Remedy as a valid name.
- ***Remedy ticket category**
  Specifies the BMC Remedy ticket category.
  Default: Default
- ***Remedy ticket item**
  Specifies the BMC Remedy ticket item.
  Default: Default
- ***Remedy ticket type**
  Specifies the BMC Remedy ticket type.
  Default: Default
- ***Source in Remedy for ticket creation**
  Specifies the BMC Remedy ticket source.
  Default: NMP (Network Management Program)
- **Auto close ticket when alarm cleared**

  Specifies whether a BMC Remedy ticket is automatically closed when the related CA SOI alert is cleared.
- **Email address to send failure messages**
  Specifies a valid email address.
- **Max retries for Remedy operations**
  Specifies the number retry attempts when a failure occurs.
  Default: 3

5. Click Next.
   The Remedy Optional Parameters page opens.
   Note: If the page does not open, ensure that you have entered values for all fields.
6. Enter values in the fields as necessary.
   These fields correspond to the default fields in the BMC Remedy HPD:HelpDesk form and are not mandatory by default for incident creation. However, if the BMC Remedy are customized, a field may be required.

   NOTE
   The Ticket priority and urgency fields have a preset list of values. You can change the values in this list by editing the values in the Common Dataset. For more information about customizing BMC Remedy ticket fields, see How to Customize BMC Remedy Ticket Fields with CA SOI Alert Parameters.

7. Click Next.
   The Start Request page opens.
8. Click Finish.
   The request displays on the Start Request Instances pane.
9. Click Refresh.
   A State value of Completed indicates a successful BMC Remedy connection.

Configure ITIL BMC Remedy Server Settings

You configure the ITIL BMC Remedy server settings within CA Process Automation to enable the connection between the two products.

Follow these steps:

1. Start the CA Process Automation Management Console.
2. Click User Requests.
   The User Requests pane opens.

   NOTE
   If the pane does not open, recycle the console. Changes may not have been applied from a previous operation.
3. Expand CA Remedy Gateway, right-click RemedyITSMConfiguration, and select Start Request.
   The Remedy Parameters pane opens on the right. This pane contains fields that are mandatory for incident creation.
4. Enter values for the following fields as described. The fields that are marked as optional use a default value if left blank.

   WARNING
   Some fields such as Category, Item, and Type only accept specific values. For example, the Priority field can only be Urgent, High, Medium, or Low. If you enter an invalid value, the incident is not created. For more information about valid entries, see the BMC Remedy AR System documentation.

   NOTE
   An asterisk at the beginning of the name indicates that you can input an expression (in JavaScript). Static strings are double-quoted.

   Some of the parameters have drop-down lists that correspond to default BMC Remedy values. To customize the values in these lists, you can modify the Common Dataset. For more information about customizing BMC Remedy ticket fields, see How to Customize BMC Remedy Ticket Fields with CA SOI Alert Parameters.

   – **MidTier Hostname/IP Address and port**
     Specifies the BMC Remedy Mid-Tier server host name or IP address and port in the following format: *server:port*.
     Use the following URL to test this value: http://<server>:<port>/arsys/home.
     If you omit the port number, it defaults to 80.
   – **Remedy Mid-Tier Server is configured to use SSL**

     Specifies whether the BMC Remedy Mid-Tier server is configured to use SSL.

     NOTE
     If the server is configured to use SSL, ensure that you have installed the SSL certificate on the CA Process Automation server. For more information about installing the SSL certificate, see Install SSL Certificate for Remedy Mid-Tier Server.
   – **Remedy AR Server Name**
     Specifies the BMC Remedy AR System server name. Use the following URL to test this value: http://<midtier_server>:<port>/arsys/WSDL/public/<ar_server>/ITPAM_HelpDesk_Modify_Service.
   – **Username to create, query, and modify Incidents**

Specifies the user name to log in to BMC Remedy.

- **Password for the user**
  Specifies a valid password for the BMC Remedy user name.
- **Maximum concurrent connections to Remedy**
  Defines the number of simultaneous BMC Remedy connections allowed.
  Default: 10
- ***Customer first name**
  Specifies the BMC Remedy customer first name. This value is used as the Requester Name in the generated BMC Remedy incident. The name must exist in BMC Remedy as a valid name.
- ***Customer last name**
  Specifies the BMC Remedy customer last name. This value is used as the Requester Name in the generated BMC Remedy incident. The name must exist in BMC Remedy as a valid name.
- ***Reported source for incident creation**
  Specifies the source that created the ticket. You can map the source to the CA SOI event_source property. For more information, see How to Customize BMC Remedy Ticket Fields with CA SOI Alert Parameters.
- ***Incident service type**
  Specifies the service type that the incident requires. Select a value from the drop-down list.
- ***Impact**
  Specifies the incident impact. You can map this value to the CA SOI impact value. For more information, see How to Customize BMC Remedy Ticket Fields with CA SOI Alert Parameters.
- ***Urgency**
  Specifies the incident urgency. You can map this value to any of the exposed CA SOI values. For more information, see How to Customize BMC Remedy Ticket Fields with CA SOI Alert Parameters.
- ***Categorization tier 1**
  (Optional) Specifies the first categorization tier.
  Default: Default
- ***Categorization tier 2**
  (Optional) Specifies the second categorization tier.
  Default: Default
- ***Categorization tier 3**
  (Optional) Specifies the third categorization tier.
  Default: Default
- ***Assignee**
  (Optional) Specifies the incident assignee.
- ***Assigned group**
  (Optional) Specifies the assigned group name.
- ***Assigned group shift name**
  (Optional) Specifies the assigned group shift name.
- ***Assigned support company**
  (Optional) Specifies the support company name.
- ***Assigned support organization**
  (Optional) Specifies the support organization name.
- ***Resolution reason when alert is cleared**
  Specifies a resolution reason. Select a value from the drop-down list. This field is required if the Auto close ticket when alert cleared check box is selected.
- **Auto close ticket when alarm cleared**

  Specifies whether a BMC Remedy ticket is automatically closed when the related CA SOI alert is cleared.
- **Auto clear alert when ticket closes**
  Specifies whether a CA SOI alert is cleared when the associated BMC Remedy ticket is closed.
- **Email to report Remedy related errors**

Specifies a valid email address.

– **Retries before reporting error**
  Specifies the number retry attempts when a failure occurs.
  Default: 3

The Remedy WorkInfo Parameters page opens. The fields on this page are not mandatory by default for incident creation.

5. Enter values in the fields as necessary and click Next.
   The Remedy Additional Parameters page opens. The fields on this page are not mandatory by default for incident creation.

   NOTE
   If the page does not open, ensure that you have entered values for all fields on the Remedy Parameters page.

6. Enter values in the fields as necessary and click Next.
   The Remedy Custom Parameters page opens. For more information about this page, see How to Customize BMC Remedy Ticket Fields with CA SOI Alert Parameters.
7. Click Next.
   The Start Request page opens.
8. Click Finish.
   The request displays on the Start Request Instances pane.
9. Click Refresh.
   A State value of Completed indicates a successful BMC Remedy connection.

Test BMC Remedy Connection

You can test the BMC Remedy connection after configuring the server settings in CA Process Automation.

Follow these steps:

1. Start the CA Process Automation Management Console.
2. Click User Requests.
   The User Requests pane opens.
3. Expand CA Remedy Gateway, right-click TestRemedyServerConnection, and select Start Request.
   The Start Request page opens on the right.
4. Click Finish.
   The request displays on the Start Request Instances pane.
5. Click Refresh.
   A State value of Completed indicates a successful BMC Remedy connection.

Configure and Test CA SOI Connection

This section describes how to define the CA SOI server to the CA Process Automation system and test the connection between the products.

Configure CA SOI Server Settings

You configure the CA SOI server settings in CA Process Automation to enable the connection between the two products.

Follow these steps:

1. Start the CA Process Automation Management Console.
2. Click User Requests.
   The User Requests pane opens.

NOTE
If the pane does not open, recycle the console. Changes may not have been applied from a previous operation.

3. Expand CA Remedy Gateway, right-click SSAServerConfiguration, and select Start Request.
   The SAM Parameters page opens.
4. Complete the following fields and click Next.
   - **SSA Web Services Hostname**
     Defines the SA Manager host name.
   - **SSA Web Services Port**
     Defines the SA Manager and web services port. This port is 7090 by default.
   - **SSA Username**
     Defines a valid user name to access CA SOI.
   - **SSA Password**
     Defines the password for the user that is defined in the SSA Username field.

   The Start Request page opens.
5. Click Finish.
   The request displays on the Start Request Instances pane.
6. Click Refresh.
   A State value of Completed indicates a successful CA SOI configuration.

Test CA SOI Connection

You can test the CA SOI connection after configuring the server settings in CA Process Automation.

Follow these steps:

1. Start the CA Process Automation Management Console.
2. Click User Requests.
   The User Requests pane opens.

   NOTE
   If the pane does not open, recycle the console. Changes may not have been applied from a previous operation.
3. Expand CA Remedy Gateway, right-click TestSSAServersConnection, and select Start Request.
   The Start Request page opens.
4. Click Finish.
   The request displays on the Start Request Instances pane.
5. Click Refresh.
   A State value of Completed indicates a successful CA SOI configuration.

Test BMC Remedy Integration

After you have completed all previous steps, you can test the functionality of the CA Process Automation Remedy connector and configured CA Remedy Gateway files. This procedure describes how to create manually a test BMC Remedy ticket from CA Process Automation to test the configuration.

NOTE
For more information about using the integration to create incidents in CA SOI, see Using the BMC Remedy Integration.

Follow these steps:

1. Right-click CreateRemedyTestTicket in the User Requests pane of the CA Process Automation Management Console and select Start Request.

The Create Ticket page opens on the right.
2. Enter a description for the ticket and click Next.
   The Start Request page opens.
3. Click Finish.
   The process starts and the request displays on the Start Request Instances pane.
4. Click Refresh.
   A State value of Completed indicates a successful BMC Remedy connection and incident creation.
5. Click Default Process Watch in the left pane and click Ended Instances.
   A list of currently ended processes displays in the Ended Instances pane.
6. Right-click the process that has the same start time as when you ran the test process and select Process Dataset.
   A pane opens on the right with a list of parameters.
   If the request failed, scroll to the errorMessage field for an explanation of the request failure.

   NOTE
   One potential reason for failure is if you have customized your BMC Remedy form to add custom fields or make previously optional fields required. You apply these changes in CA Process Automation for the integration to work. For more information, see How to Customize BMC Remedy Ticket Fields with CA SOI Alert Parameters.
7. Scroll to the bottom of the pane.
   The troubleTicket field contains a valid BMC Remedy ticket ID, and the ticketWebURL field contains a URL.
8. Copy the URL into a web browser.
   A BMC Remedy Login page opens.
9. Enter valid credentials and click Log In.
   The created test ticket Incident Request page opens.

If you cannot view the test ticket using the generated URL, see Verify Web Context URL Link.

Verify Web Context URL Link

If your BMC Remedy installation does not use the standard URL, you change it in your CA Process Automation configuration for the ticket links to work properly. This procedure describes how to change the value that the CA Remedy Gateway uses to invoke BMC Remedy web services from within a CA SOI alert.

This procedure is only required if the Web Context URL link in the CA SOI Alert Details tab fails to start BMC Remedy. For more information about testing the link, see Test BMC Remedy Integration.

Follow these steps:

1. Start the CA Process Automation Management Console and click ITPAM Client in the upper right.
   The CA Process Automation Client opens.
2. Click Configuration Browser, expand Default Environment in the Browser pane, and double-click Orchestrator.
   The Folders pane opens.
3. Select CA Remedy Gateway.
   The Remedy Gateway integration processes display in the right pane.
4. Double-click CARemedyGatewayProcessWatch.
   The CARemedyGatewayProcessWatch pane displays.
5. Select DS - Configuration Dataset in the Filters pane.
   The Remedy Server Configuration page opens.
6. Select the RemedyITSM or RemedyHelpDesk field, depending on your BMC Remedy version.
   The Remedy Parameters dialog opens.
7. Modify the value in the URI field to reflect the correct URI to access the BMC Remedy web services, and click OK. The default URI format is as follows:

```
http://<mid-tierhost>/arsys/services/ARService?server=<arserver>&webService=
```

The Remedy Parameters dialog closes.

8. Click Apply on the Remedy Server Configuration page.
9. Right-click Gateway Processes Restart All on the left and select Start.
   A pane displays for specifying a start time.
10. Click Finish.
    The process runs.
11. Double-click one of the following options, depending on your BMC Remedy version:
    – Create HelpDesk Incident for non-ITIL BMC Remedy
    – Query Incident to get Entry ID for ITIL BMC Remedy
12. Click Execution Settings on the left pane.
    The Execution Settings pane opens.
13. Select Post-execution code.
    The Post-execution dialog for the process you selected displays.
14. Modify Process.ticketWebURL to reflect the correct URI to access the BMC Remedy web services and click OK.
    The Post-execution dialog closes.
15. Click Apply and close the Properties dialog.
16. Click the Check In icon.
    The Versions page opens.
17. Click OK.
    The process is checked in, and the new BMC Remedy web services URI value is in effect.

Using the BMC Remedy Integration

You can create a BMC Remedy incident automatically from any CA SOI alert displayed in the Operations Console using alert escalation policy or manually. Both methods generate the same incident.

Creating a BMC Remedy incident from a CA SOI alert results in one of the following actions:

- CA SOI creates a help desk case of type Incident in the BMC Remedy system that is tied to the CA SOI alert. This incident is assigned a BMC Remedy-generated identifier, which appears as the incident Case ID number in BMC Remedy. This value is then updated in the alert Ticket ID field in CA SOI.
- CA SOI populates the incident with CA SOI alert data such as the creation date, model name, network address, root cause, and alert description.
- CA SOI populates BMC Remedy-only incident fields using system-specified default values such as the requester name, source, category, item, type, priority, and urgency. Set these values in CA Process Automation.

NOTE
For more information, see Configure and Test BMC Remedy Connection.

NOTE
SOI can automatically close any ticket that is opened after an alert has cleared. To enable this option, see Review Alert and Ticket Closure Considerations.

You can display the incident in BMC Remedy directly from the Operations Console after you create the incident.

For information about generating, managing, and viewing BMC Remedy incidents from SOI, see How to Work with a Configured Help Desk Integration.

WARNING
If you added custom fields to your BMC Remedy Help Desk Case form or fields on your form are required that are not on the Remedy Parameters or Remedy Optional Parameters page in CA Process Automation, configuration is required for the integration to work. You map all custom or required fields to CA Process Automation before using the integration. For more information, see How to Customize BMC Remedy Ticket Fields with CA SOI Alert Parameters.

How to Customize BMC Remedy Ticket Fields with CA SOI Alert Parameters

When a BMC Remedy incident is created based on a CA SOI alert, the CA Process Automation process CreateRemedyTicket creates the ticket using the configured RemedyHPDConfiguration or RemedyITSMConfiguration form.

> **NOTE**
> For more information about configuring these forms, see Configure and Test BMC Remedy Connection.

You can customize the BMC Remedy Help Desk Case form. In addition to the default BMC Remedy form fields, you create fields to support your unique business requirements. Some fields require a value when the ticket is created, while others are optional. The CA Remedy Gateway processes provide a set of predefined BMC Remedy parameters. However, you can define additional parameters for your customized HelpDesk form.

If either of the following situations applies to you, you configure support for custom or mandatory fields:

- Custom fields on your BMC Remedy HelpDesk form
- Standard BMC Remedy fields that are not on the Remedy Parameters or Remedy Optional Parameters configuration pages in CA Process Automation, are required to create a ticket and are not system-generated

You map these fields in CA Process Automation; otherwise, the ticket creation fails.

1. Customize the HPD:HelpDesk (non-ITIL) or HPD:IncidentInterface_Create (ITIL) form at your site.
   You customize this form within BMC Remedy. For more information about customizing forms in BMC Remedy, see the BMC Remedy documentation.
2. Configure BMC Remedy web services to support your custom or unlisted mandatory fields.
   Define an XML element by which to communicate the value between CA Process Automation and BMC Remedy. You must also indicate whether the parameter is for a mandatory field.
3. Configure CA Process Automation to support your custom fields.
   Specify in CA Process Automation how to populate custom or mandatory fields when generating a ticket from CA SOI. You can use both static and dynamic values.

Configure BMC Remedy Web Services XML Elements

To support custom or mandatory fields when creating a BMC Remedy ticket using the CA Process Automation Remedy connector and CA Remedy Gateway processes, you must define a parameter by which to pass information from CA SOI and the connector to BMC Remedy for each added field. If the custom or mandatory field does not appear on the Remedy Parameters or Remedy Optional Parameters pages in CA Process Automation, a parameter has not been defined and associated with the BMC Remedy field in the CA Remedy Gateway processes by default.

To declare a new parameter, you define an XML element in BMC Remedy web services. The XML element is added to the ITPAM_HelpDesk_Submit_Service (non-ITIL versions) or ITPAM_HPD_IncidentInterface_Create_WS (ITIL versions) web services definition, which is used for communication between CA Process Automation and BMC Remedy.

> **NOTE**
> This procedure assumes that the custom fields have already been defined and are present on the HPD:HelpDesk (non-ITIL) or HPD:IncidentInterface_Create form (ITIL).

Follow these steps:

1. Select one of the following from the Start menu on the AR System Server, depending on your BMC Remedy version:
   - Versions below 7.6: Programs, Action Request System, BMC Remedy Administrator on the BMC Remedy server.
   - Version 7.6: Programs, BMC Software, AR System 1, BMC Remedy Developer Studio.

   The login page opens.
2. Enter valid credentials and click OK.
   The BMC Remedy Administrator home page or BMC Remedy Developer Studio page opens.
3. Do one of the following depending on your BMC Remedy version:

- Versions below 7.6: Expand the folder for the Remedy Mid-Tier server under the Servers node and select Web Services.
- Version 7.6: Expand the folder for the Remedy Mid-Tier server, expand All Objects, and double-click Web Services.

A list of available web services displays.

4. Open ITPAM_HelpDesk_Submit_Service (non-ITIL versions) or ITPAM_HPD_IncidentInterface_Create_WS (ITIL versions).
   The Modify Web Service opens.
5. Access the Input Mapping section.
   The fields on the left are those available on the Help Desk Case form and the items on the right correspond to the element used by the CA Remedy Gateway. By default, the last active field is Create Time.

   WARNING
   The sequence of the fields and elements cannot be changed. BMC Remedy expects the parameters to come in sequence. You must add any custom or mandatory fields after the Create Time field and associated element.
6. Scroll to the top of the XML Data Type list, right-click the ROOT element, and select New, Element, Field.
   An element named New Field Element displays at the end of the list.
7. Select the new element, give it a meaningful name that corresponds to the form field, and click the button to the right of the name.
   For example, if the Solution Summary field (which is a default field on the Help Desk Case form) is mandatory at your site, you must define an associated XML element for this field such as Solution_Sum.
   The XML Properties dialog opens.
8. Verify in the Object Properties dialog that the Type (data type) is the same as the field it will be mapped to.
   To make the field Optional, set MinOccurs to 0 (it is 1 by default).
9. Select the field in the left list that corresponds to the created XML element so that the field and element are both highlighted and click Map.
   The association is made, the Map button is disabled, and the Mapping Summary information displays.

   NOTE
   The data types need to match when mapping the elements. For example, if you want to populate the dateTime field in the form, you have to create an XML element of dateTime type. CA SOI sends the creation_date and event_occurred parameters to CA Process Automation for this dateTime field.
10. Click OK.
    The Mapping - CreateInputMap dialog closes.
11. Select File, Save Web Service from the BMC Remedy Administrator browser.
    Your changes are saved.
12. Select one of the following from the Start menu:
    - Versions below 7.6: Programs, Action Request System, BMC Remedy Mid Tier, Configure ARSYSTEM
    - Version 7.6: Programs, BMC Software, AR System 1, BMC Remedy Mid Tier, Configure Mid Tier on Localhost

    A login page opens.
13. Enter valid credentials and click Login.
    The BMC Remedy Mid-tier Configuration Tool displays.
14. Select Cache Settings.
    The Cache Settings pane opens.
15. Click Flush Cache.
    The cache is updated, which ensures that the updated web services are used.

Configure CA Process Automation Custom or Mandatory Fields

After you define an XML element in BMC Remedy, you define a corresponding field in the CA Remedy Gateway processes. You can define custom or mandatory fields in the CA Remedy Gateway processes in *one* of the following ways:

- Map custom BMC Remedy fields
  The Add Custom Remedy Fields Mapping option lets you populate fields with static text or an available CA SOI alert attribute value.
- Add Custom JavaScript
  This more complex approach lets you use custom JavaScript code to generate a value for your custom or mandatory field.

  WARNING
  Do not attempt to combine these two methods when adding a custom or mandatory field.

When using either method, you use predefined CA Process Automation Remedy connector variables when indicating how to generate a value for a field. For more information about available variables, see CA Process Automation Remedy Connector Variables.

CA Process Automation Remedy Connector Variables

When the CA Process Automation Remedy connector creates a BMC Remedy ticket, a user can populate the fields by the following methods:

- Using static text or default values that the CA Process Automation or the BMC Remedy settings specify. Settings include Requester Name or Login.
- Using CA SOI alert information. For example, the BMC Remedy ticket Description value is generated using the CA SOI Description field from the incident-generating alert. Information that CA SOI generates comes from the Alert attribute value using the CA SOI Action Editor

Regardless of the field population method, a mandatory field on the BMC Remedy form must have a value for BMC Remedy to create the incident.

When specifying how fields are populated during ticket creation, use static values or CA Process Automation-provided variables. The following variables are available:

- **CA SOI alert attribute variables**
  Describes alert details.
- **BMC Remedy parameters**
  Passes information to BMC Remedy when generating an incident.

This section describes the available CA SOI alert variables, BMC Remedy parameters, and the internal arrays that CA Process Automation uses to store user-specified custom and mandatory fields.

CA SOI Alert Attributes

CA SOI alert attribute variables describe alert details and you can use them to populate BMC Remedy form fields. When using the custom BMC Remedy fields mapping method, the attributes are available from a drop-down list. When writing custom JavaScript source code, the attribute variables are accessible using the format Process.*variablename*.

The following CA SOI alert attribute variables are available:

NOTE
The variable name is in parentheses.

- Alert Description (description)
- Type of Ticket (ticketType)
- Acknowledged (acknowledged)
- Assignee (assignee)
- Maximum priority of all the services that the alert impacts (max_priority)
- Maximum impact of all the services that the alert impacts (max_impact)
- Severity of the alert (severity)
- IP Address (IP_address)
- Model Family (model_family)
- Model Name (model_name)
- Model Class (model_class)
- Model Description (model_description)
- Root Cause (rootcause)
- Connector name (connector_name)
- Creation date of alert (creation_date)
- Detail of alert (detail)
- Event source (event_source)
- Event occurred (event_occurred)
- Event source ID (event_source_id)
- Vendor name (vendor_name)

View BMC Remedy Parameters

The CA Process Automation Remedy connector uses the BMC Remedy parameters to send data to BMC Remedy to create an incident. When defining custom fields, you use the RemedyHelpDesk (non-ITIL) or RemedyITSM (ITIL) variable, which is a copy of the BMC Remedy parameters.

This procedure describes how to display a list of these parameters from CA Process Automation.

Follow these steps:

1. Click ITPAM Client at the upper right of the CA Process Automation Management Console.
   The ITPAM Client opens in the Configuration Browser page.
2. Expand Default Environment in the Browser pane and double-click Orchestrator.
   The Folders pane opens.
3. Select the CA Remedy Gateway folder.
   The Remedy Gateway integration processes display on the right pane.
4. Double-click Configuration Dataset.
   The Value Definition pane opens. This pane contains parameters for the different BMC Remedy installations.
5. Expand Remedy Server Configuration, RemedyITSM or RemedyHelpDesk, and Remedy Parameters.
   A list of BMC Remedy parameters displays. A copy of these parameters is passed to the CreateRemedyTicket process during runtime. For more information, see Remedy* Variable.
   The BMC Remedy parameters display in the right pane with any currently defined values. The values are based on what your Remedy*Configuration.
6. Select a BMC Remedy parameter in the Value Definition pane to display attributes for that parameter in the right pane.

Remedy* Variable

The CA Process Automation Remedy connector uses the Remedy* variable to send data to BMC Remedy for creating a ticket. When defining custom fields, you use the Remedy* variable, which is a copy of the BMC Remedy parameters. The CreateRemedyTicket process creates its own copy of the Remedy* variable when the process is triggered to pass

information to BMC Remedy. The asterisk in the variable corresponds to HelpDesk or ITSM, depending on the version of BMC Remedy that you are using.

> **NOTE**
> The Remedy* variable is only accessible when using the CreateRemedyTicket process. For more information, including source code examples, see Add Custom JavaScript.

The Remedy* variable is a ValueMap data type variable that consists of a group of variables, each describing a BMC Remedy ticket attribute. You access each element in the ValueMap using the dot (.) operator. When used in source code, the elements of a Remedy* variable are accessible using the format Remedy*.*variableName*. For example, to reference the (non-ITIL) BMC Remedy HelpDesk site, you would use RemedyHelpDesk.Site.

> **NOTE**
> The fully qualified reference is Process.RemedyHelpDesk.*variableName*, but RemedyHelpDesk.*variableName* is acceptable.

The following BMC Remedy ticket attributes are available. Each item represents variableName in Remedy*.*variableName*:

- MidTier
- IsSSL
- ARServer
- Username
- Password
- RequesterLogin
- RequesterName
- URI
- Source
- HDModifyService
- HDQueryService
- HDSubmitService
- Category
- Type
- Item
- ContactEmail
- ErrorRetries
- Priority
- Urgency
- OrgSubmit
- PhoneNumber
- Escalated
- HotList
- Region
- Site
- Office
- Department
- Pending
- WorkLog

CA Process Automation also provides a listing of these values. For more information about displaying these values, see View BMC Remedy Parameters and How to Display a CA SOI Alert Instance.

For more information about using the Remedy* variable in custom source code, see Add Custom JavaScript. For more information about CA Process Automation development practices, see the *CA Process Automation User Guide*.

Tag and Value Arrays

You can populate custom or mandatory fields that are not presented in the CA Process Automation Remedy Parameters or Remedy Optional Parameters pages. Populate the fields in *one* of the following ways:

- Using the Add Custom Remedy Fields Mapping option during BMC Remedy server configuration.
  This method lets you populate a field with a static value or one of the CA SOI alert attributes using a drop-down list. When you add fields using this method, the CA Process Automation Remedy connector stores the fields and values in two arrays:
  - **Keys**
    Stores the field names.
  - **Values**
    Stores the values that are used to populate the custom and mandatory fields.
- Using the CreateRemedyTicket Add Custom Code operation to create customized arrays of tags and values.
  This method lets you populate the fields with static or dynamic values.

When adding fields to the Keys array, the fields must be in the exact sequence as the XML elements in the BMC Remedy web services. For more information about defining XML elements, see Configure BMC Remedy Web Services XML Elements.

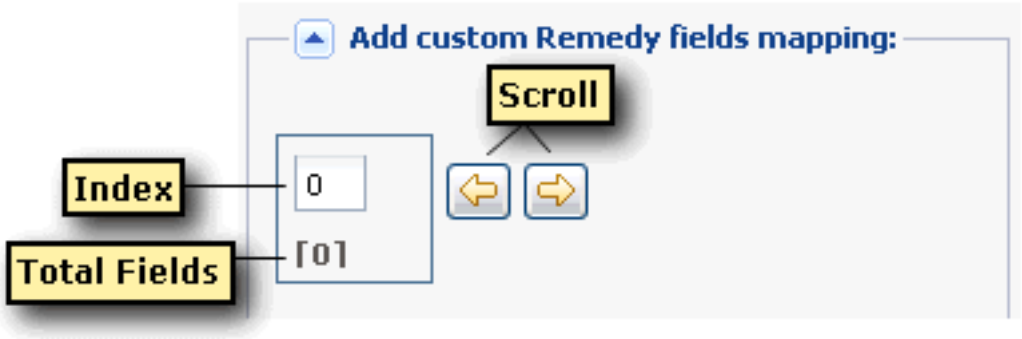
Map Custom Remedy Fields

This procedure describes how to define XML elements to CA Process Automation using the Add custom Remedy fields mapping component of the Remedy Optional Parameters page in CA Process Automation. You can use this approach when the field is populated with static text or an available CA SOI alert attribute.

> **WARNING**
> Use only one of the methods for adding fields. Do not attempt to combine mapping fields and adding custom code.

Follow these steps:

1. Follow the steps for configuring BMC Remedy server settings in Configure Non-ITIL BMC Remedy Server Settings or Configure ITIL BMC Remedy Server Settings. Stop once you reach the Remedy Optional Parameters (non-ITIL) or Remedy Additional Parameters (ITIL) page.
2. (Optional) Click Next on the Remedy Additional Parameters page (ITIL configuration only).
   The Remedy Custom Parameters page opens.
3. Click the Add custom Remedy fields mapping expansion button.
   The following displays:



4. Place your cursor in the Index field.
   The + and x icons display.
5. Click + to add a field.

The Add Custom Remedy Fields pane appears. The total number of fields increases by one. Because the index starts at zero, an index value of zero represents the first element.

6. Complete the following fields:
   - **Remedy field to populate**
     Specifies the name of the XML element that you created in BMC Remedy web services. Enter the fields in the same sequence as they exist in BMC Remedy web services. This value is stored in the Tag Array named Keys.
   - **Value for the fields above**
     Specifies the static text for this field, or the CA SOI alert attribute value. Enter a static value or select the CA SOI alert attribute value from the drop-down list. This value is stored in the ValueArray.

     NOTE
     Currently, the only dateTime attribute available is creation_date. The BMC Remedy field for dateTime has the following format: yyyy-MM-ddThh:mm:ss-tz:tz. The tz:tz values are the hh:mm of time zone information.
7. (Optional) Click + to add an element, and
   click to
   advance to the added element.
   The value for the total number of fields and the index value increase by one.
8. (Optional) Enter the name of the additional element and its value in the appropriate fields.

   WARNING
   The fields must exist in the same sequence as they exist in BMC Remedy.
9. Click Next after you have created all necessary fields.
   The Start Request pane opens.
10. Click Finish.
    The request displays on the Start Request Instances pane.
11. Click Refresh.
    A State value of Completed indicates a successful connection.

Add Custom JavaScript

Custom fields can require more than static text or a direct mapping of a field value. Advanced custom fields that need more flexibility can require that you enter custom code.

WARNING
Use only one of the methods for adding fields. Do not attempt to combine mapping fields and adding custom code.

Perform *one* of the following actions to create custom fields using custom JavaScript code:

- Access and check out the CreateITSMTicket process.
- Add custom code to the CreateITSMTicket process.
- Access and modify variables in the CreateITSMTicket process.
- Check in the modified CreateITSMTicket process.

Follow these steps:

1. Click the following tab:
   - CA Process Automation r3.1: Configuration Browser tab.
   - CA Process Automationr4.0/4.1: Library tab.
2. Expand Default Environment and double-click Orchestrator.
   The Folders pane opens.
3. Select CA Remedy Gateway and double-click the CreateITSMTicket process in the right pane.

The CreateITSMTicket page opens.
For more information about the pane, see Configure Processes for BMC Remedy Integration.

NOTE
If the icon labels do not display, select Edit User Preferences, select Show icon information on the Configuration tab and click OK.

4. Click or the toolbar to check out the process.
   The Check Out icon is disabled and the process is checked out.
5. Double-click the icon named Add custom JavaScript code here to send values to Remedy * Incident.
   The Properties of UserAddedCustom*Code pane opens.
6. Click in the Source code field.
   The Source code page opens.
7. Enter the source code in JavaScript to create mappings that are based on alert attributes and using BMC Remedy parameters.

   NOTE
   For available variables, see CA Process Automation Remedy Connector Variables. For source code examples, see Code Examples.
8. Click OK.
   The Source code pane closes and the code is saved.
9. Click Apply to accept any changes and close the Properties pane.
   The custom code is added to the process.
10. (Optional) Double-click the icon named Create * Incident to access variables.
    The Properties of Create*Incident pane opens.
11. (Optional) Expand the HelpDesk Create Case Parameters or ITSM Incident Parameters and HelpDesk Create Case Other Parameters or ITSM Incident Optional Parameters panes. Modify the variables that are used when generating a BMC Remedy incident.
12. (Optional) Expand the Web Service User Parameters pane to view custom field definitions.
    Custom fields are configured as Keys and Value arrays on the BMC Remedy operator. You can add or further customize fields and values using JavaScript.
13. (Optional) Click Apply to save any variable changes and close the Properties pane.
14. Click the icon on the toolbar to check in the modified process.
    The Versions page opens.
15. Click OK.
    The modified process is checked in.

    NOTE
    For more information about datasets, see the *CA Process Automation User Guide*.

Display a CA SOI Alert Instance

Each time the CreateRemedyTicket process runs, you can display the instance of the process to view the parameters values that are passed to BMC Remedy. Use this procedure to verify that any custom or mandatory fields that you added are applied when a BMC Remedy incident is created.

Follow these steps:

1. Select CA Remedy Gateway from the Folders pane in the CA Process Automation Client application.
   The Remedy Gateway integration processes display.
2. Double-click CARemedyGatewayProcessWatch.
   A list of filters displays in the left pane.
3. Expand and select the CreateRemedyTicket filter.
   A list of available instance filters appears in the Filters pane and a list of instances that satisfy the filter criteria display in the right pane.
4. Double-click an instance.
   The CreateRemedyTicket <*ProcessID*> pane displays. The initial view is the Main Editor, which provides a graphical representation of the process.
5. Select the Dataset tab.
   The Local_Dataset pane on the left displays the parameters used in the process in the following two categories:
   – **System**
     Lists CA Process Automation system-generated variables, including the parameters that are set during configuration.
   – **Input Parameters**
     Lists parameters that are received from CA SOI.
6. Select Remedy* (depending on your BMC Remedy version) within the System list.
   The Remedy* values used in this instance of the CreateRemedyTicket process are displayed.
7. Select Input Parameters.
   A list of input parameters displays. These parameters represent the alert attributes from CA SOI used in this instance of the CreateRemedyTicket process.

Code Examples

This section lists examples of using custom code to populate fields with dynamic values as part of the CreateRemedyTicket process. The examples use alert attributes and BMC Remedy attributes to create unique values for BMC Remedy fields that are based on CA SOI alerts.

Alert attribute values are in the format Process.*variableName* and BMC Remedy values are in the format Remedy*.*variableName*. Substitute HelpDesk or ITSM in place of the asterisk. The Remedy* variable is a ValueMap data type variable, so you must reference its elements as Remedy*.*variableName* for the code to work.

Example: Map CA SOI Service Priority to Remedy Urgency

The following example maps the CA SOI service priority field to the BMC Remedy Urgency field:

```
if (Process.priority != undefined) {
       if (Process.priority == 10)
              RemedyITSM.Urgency = "1-Critical";
       else if ( Process.priority == 9)
              RemedyITSM.Urgency = "2-High";
       else if ( Process.priority == 8)
              RemedyITSM.Urgency = "3-Medium";
      else if ( Process.priority == 7)
              RemedyITSM.Urgency = "4-Low";
      else
```

```
   RemedyITSM.Urgency = "3-Medium";
}

}
```

Example: Customization using Datasets Array

The following example replaces the RemedyITSM.Impact variable with a customized Impact field (Process.CustomImpact) by importing the values from the Mapping Dataset to the current process:

```
Process.ImpactMap = Datasets["Mapping Dataset"].Impact;

if (Process.ImpactMap[Process.impact] != undefined)
{
       Process.CustomImpact = Process.ImpactMap[Process.impact];
}
else
{
       Process.CustomImpact =RemedyITSM.Impact;
}
```

The syntax for using a dataset is as follows:

```
Dataset["Dataset Name"].variableName
```

The Impact variable is an Array type, so the first value is accessible using an index of 0. View the Mapping Dataset used in this example by double-clicking the MappingDataset process from a listing of CA Remedy Gateway integration processes.

Example: Customization using Datasets ValueMap

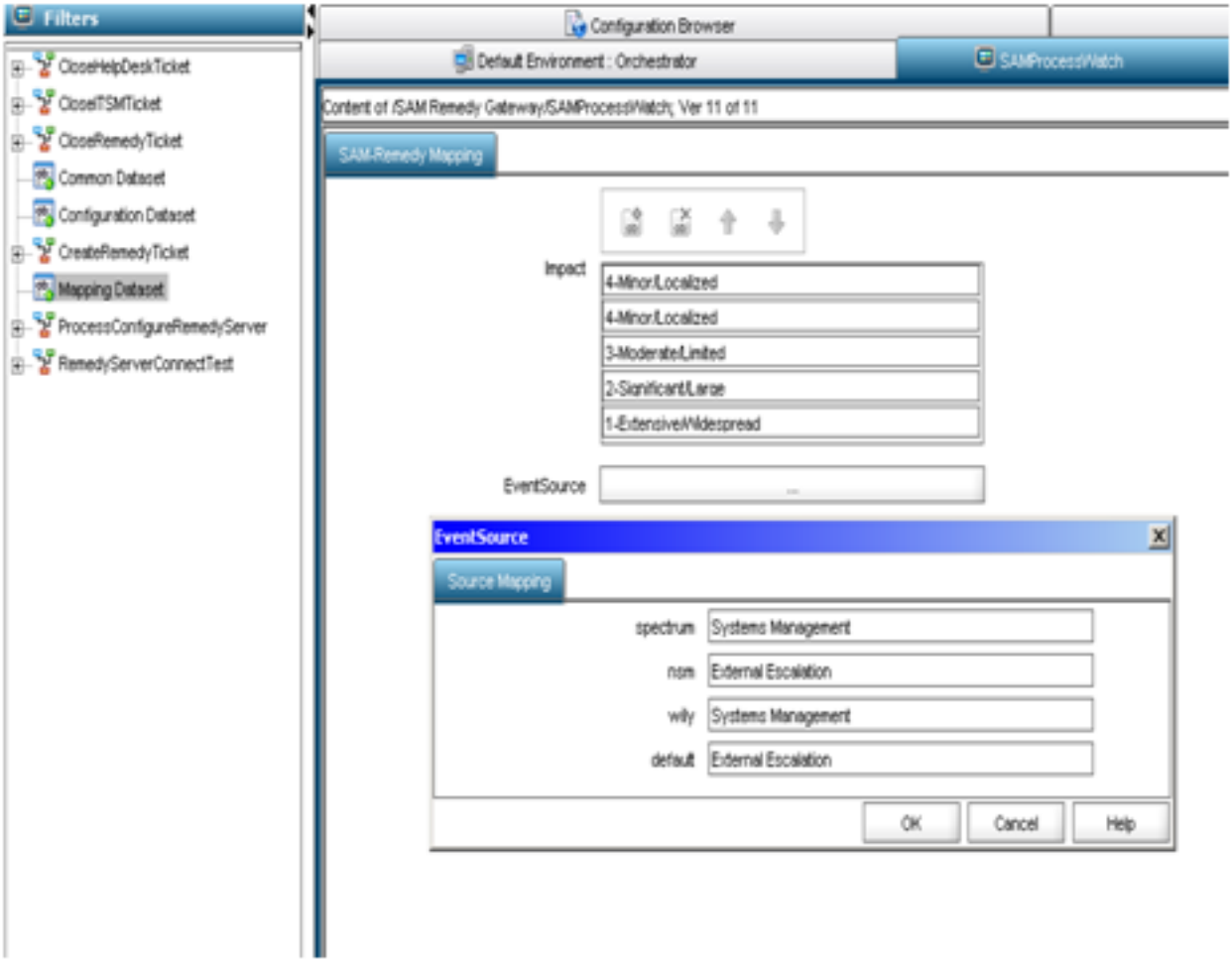
The following example replaces the RemedyITSM.ReportedSource variable with the customized source field Process.CustomSource by importing the EventSource value from the Mapping Dataset to the current process:

```
Process.SourceMap = Datasets["Mapping Dataset"].EventSource;

If (Process.event_source != undefined)
{
       If (Process.SourceMap[Process.event_source] != undefined)
       {              /* event_source value from SAM */
              Process.CustomSource =
              Process.SourceMap[Process.event_source];
       }
       else
              Process.CustomSource = Process.SourceMap.default;

}
else
{
       Process.CustomSource = RemedyITSM.ReportedSource;
}
```

Process.SourceMap[Process.event_source] sets Process.CustomSource as Systems Management if Process.event_source is spectrum:



To access the EventSource dialog, click the EventSource field in the Mapping Dataset dialog.

Example: Populating custom BMC Remedy fields dynamically

The following example replaces the Process.Keys and Process.Values fields with the customized arrays CustomTags and CustomValues:

```
// create tags and value arrays of same size
Process.CustomTags = new Array();
Process.CustomTags.length = 2;
Process.CustomValues = new Array();
Process.CustomValues.length = 2;

// create tag element names
Process.CustomTags[0] = "Root_Cause";
Process.CustomTags[1] = "Acknowledged";

// assign valid values to the tags from SAM alert values
// make sure values being sent to Remedy are values Remedy will accept for these fields
Process.CustomValues[0] = Process.rootcause;
Process.CustomValues[1] = Process.acknowledged;
```

Are you enter the code, rename the Process.Keys array to Process.CustomTags and the Process.Values array to Process.CustomValues in the Web Service User Parameters pane of the CreateRemedyTicket process. For more information, see Add Custom JavaScript.

How to Configure a CA Service Desk Integration

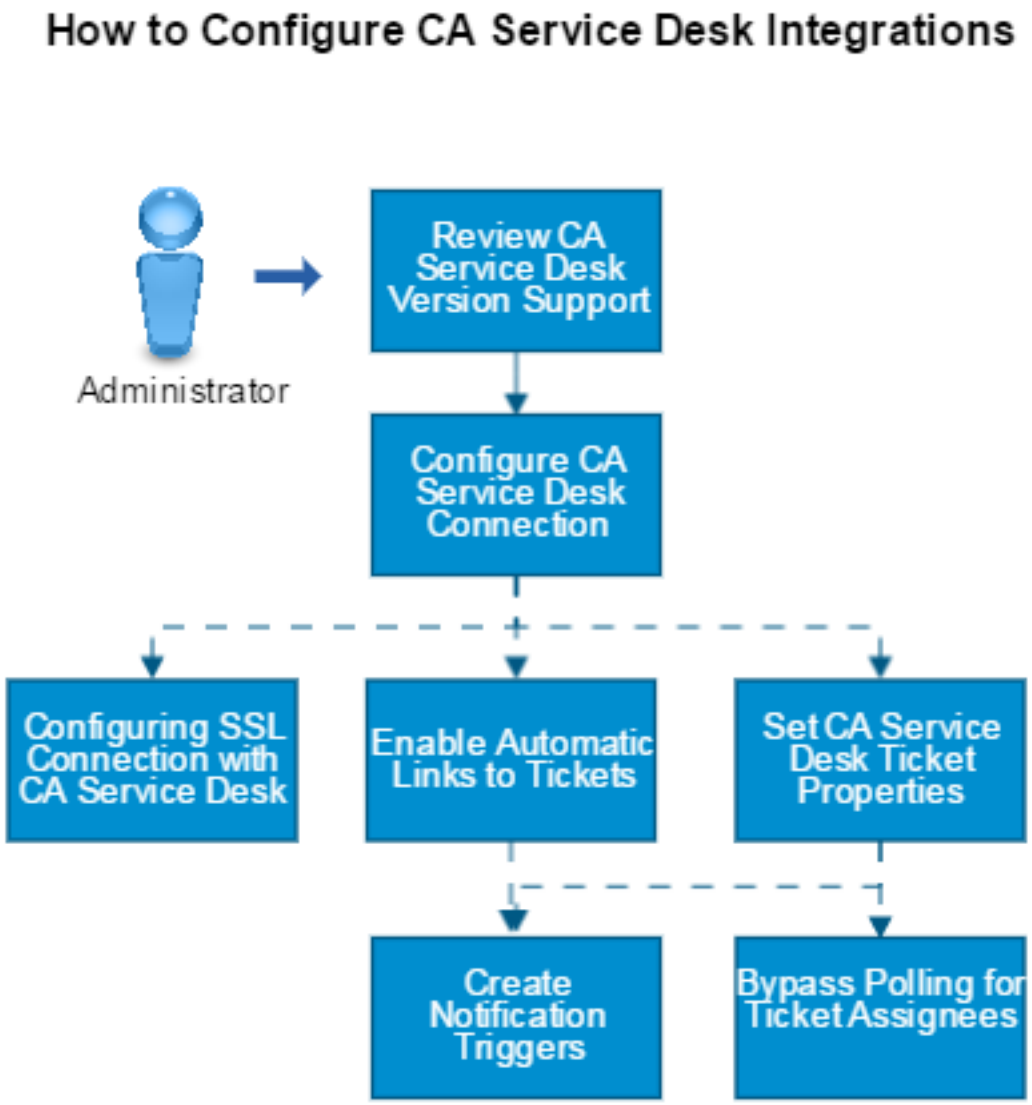
As an administrator, you configure CA Service Desk integration to integrate CA SOI alert escalation with CA Service Desk incident management.

To configure CA Service Desk integration, you configure a connection with CA Service Desk.

> **WARNING**
> The CA Service Desk integration described in this section differs from the CA Catalyst connector for CA Service Desk. The described integration is specific to CA SOI, and lets you generate CA Service Desk tickets on CA SOI alerts manually or through escalation policy. The CA Catalyst connector for CA Service Desk imports CA Service Desk information, such as users, incidents, and so on, into CA SOI for management as CIs. For more information about the CA Catalyst connector for CA Service Desk, see the *CA Service Desk Connector Guide* provided with the connector package. You can download the connector from CA Support.

Use this scenario to guide you through the process:

Figure 14: Configure Service Desk Integrations



1. Review CA Service Desk version support.
2. Configure CA Service Desk connection.
3. (Optional) Configure SSL connection with CA Service Desk by exporting the CA Service Desk SSL certificate then add the SSL certificate to the shared CA SOI/CA Catalyst trust store.
4. (Optional) Enable automatic links to tickets.
5. (Optional) Set CA Service Desk ticket properties.
6. (Optional) Create notification triggers in CA Service Desk.
7. (Optional) Bypass CA SOI polling for ticket assignees.
8. (Optional) Review CA Spectrum and CA Service Desk ticket synchronization considerations.
9. (Optional) Review CA Service Desk troubleshooting information.
10. Work with the configured help desk integration.

Review CA Service Desk Version Support

See Software Support for current CA Service Desk Support.

Configure CA Service Desk Connection

Configure the connection to CA Service Desk in CA SOI to integrate alert and incident management between CA SOI and CA Service Desk.

Follow these steps:

1. Click the Administration tab on the Dashboard.
2. Expand CA Service Operations Insight Manager Configuration and the server name and click Help Desk Configuration.
3. Select the appropriate version of CA Service Desk in the Help Desk Type drop-down list.

   NOTE
   The Service Desk Web Services component must be installed for the version that you select.
4. Enter CA Service Desk server properties. Select the SSL check box if CA Service Desk is SSL-enabled.

   NOTE
   For more information about enabling integration with SSL-enabled CA Service Desk, see Export the CA Service Desk SSL Certificate.

   The user credentials must either have administrator privileges or belong to an Access Type that has rights to CA Service Desk Web Services and APIs. In addition to the Administrator role, the Level 2 Analyst role has the appropriate rights.
5. Click Test.
   A successful connection message displays if you entered valid CA Service Desk server settings.
6. Click Save.

   NOTE
   If you changed an existing Help Desk Configuration setting (for example, a BMC Remedy or different CA Service Desk server integration), restart the CA SOI Application Server service for the change to take effect.

Export CA Service Desk SSL Certificate

Contents

CA SOI can integrate with a CA Service Desk installation that uses SSL with a self-signed certificate.

You configure SSL without a fully qualified domain name (FQDN). The URL generated in a CA SOI alert to link to a trouble ticket does not use an FQDN.

You export the SSL certificate that CA Service Desk uses so that you can import it into CA SOI.

Follow these steps:

1. Enter the URL to access the SSL-enabled CA Service Desk in a web browser.
   A security alert page opens with certificate information.
2. Click Continue to this website (not recommended).
3. Click Certificate Error next to the web URL.
4. Click View Certificate.
   The Certificate page opens.
5. Select the Details tab and click Copy to File.
   The Certificate Export Wizard page opens.

6. Click Next.
   The Export File Format page displays.
7. Retain the defaults and click Next.
   The File to Export page opens.
8. Click Browse, navigate to the location you want to save the certificate, enter a certificate name, and click Save.
   The certificate location and name display on the File to Export page.
9. Click Next.
   The Completing the Certificate Export Wizard page opens.
10. Click Finish.
    A confirmation page opens. The CA Service Desk SSL certificate is exported.

Import SSL Certificate into CA SOI

After you export the CA Service Desk SSL certificate, import the certificate into CA SOI so that it can use the certificate to access CA Service Desk.

Follow these steps:

1. Access the CA SOI Console Administration page by entering the following URL in a web browser:
   `http://<servername>:<port>/sam/admin`
   - **servername**
     Specifies the SA Manager server name.
   - **port**
     Specifies the SA Manager Tomcat server port.
     Default: 7090

   The CA SOI Console Administration page opens.
2. Click SSL Certificates.
   The SSL Certificates page opens.
3. Click Browse next to the File with Certificate field.
   A file page opens.
4. Select the certificate that you exported from CA Service Desk and click Open.
   The certificate path and file name display in the File with Certificate field.
5. Give the certificate a descriptive name in the Alias Name field and click Save.
   A restart page opens. Restart the CA SOI server for the certificate to import.
6. Click OK.
   A confirmation message displays at the bottom of the page and the certificate is imported.
7. Access the Help Desk Configuration page of the Administration UI and verify that the CA Service Desk server information is entered correctly.

   NOTE
   For more information, see Configure CA Service Desk Connection.
8. Select the SSL check box and click Test.
   A successful connection message displays.
9. Click Save.
   CA SOI is configured to connect to CA Service Desk through SSL.

Enable Automatic Links to Tickets

Click the Ticket ID link for an alert in CA SOI to open CA Service Desk in context to see the associated ticket details.

When you click the link, you are prompted for a user name and password to log in to CA Service Desk. If you set up external authorization with Apache Tomcat on the CA Service Desk system, you can proceed directly to the ticket detail without logging in.

Perform this procedure on the CA Service Desk server.

Follow these steps:

1. Go to the web site http://jcifs.samba.org.
2. Download the latest version of the jcifs.jar file and copy it to NX_ROOT\bopcfg\www\CATALINA_BASE\webapps\CAisd\WEB-INF\lib on the CA Service Desk system.
3. Open the web.xml file that is located at NX_ROOT\bopcfg\www\CATALINA_BASE\webapps\CAisd\WEB-INF.
4. Add the following code after the *Add filter-mapping here* section:

```
<!-- Add filter-mapping here -->
<filter>
    <filter-name>NtlmHttpFilter</filter-name>
    <filter-class>jcifs.http.NtlmHttpFilter</filter-class>
    <init-param>
       <param-name>jcifs.smb.client.domain</param-namex>
       <param-value>domainname</param-value>
    </init-param>
    <init-param>
       <param-name>jcifs.netbios.wins</param-name>
       <param-value>WINS IP</param-value>
    </init-param>
</filter>
<filter-mapping>
    <filter-name>NtlmHttpFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
<!-- Context Listener -->
```

5. Restart Tomcat by entering the following commands on the command line:

```
pdm_tomcat_nxd -c stop
pdm_tomcat_nxd -c start
```

6. Log on to the CA Service Desk web interface and go to the Administration tab, Security, Access Types. Click the default access type.
   The Access Type Detail page opens.
   For more information about configuring CA Service Desk using the web interface, see the *CA Service Desk Administration Guide*.
7. Click Edit.
   The Update Access Type page opens.
8. Go to the Web Authentication tab and enter a check mark in the Allow External Authentication check box.
9. Create a contact with your domain login as the System Login.
10. Run the Tomcat URL.
    You can now log in to CA Service Desk directly.

Set CA Service Desk Ticket Properties

You can populate the properties of a CA Service Desk ticket that CA SOI created based on an alert from the Operations Console in either of the following ways:

- By adding one of the provided ticket properties to a create ticket or create announcement escalation action
- By creating a custom ticket or announcement property and adding it to an escalation action

For more information about setting ticket property values, adding provided properties to escalation actions, and creating custom properties, see How to Create Escalation Policy.

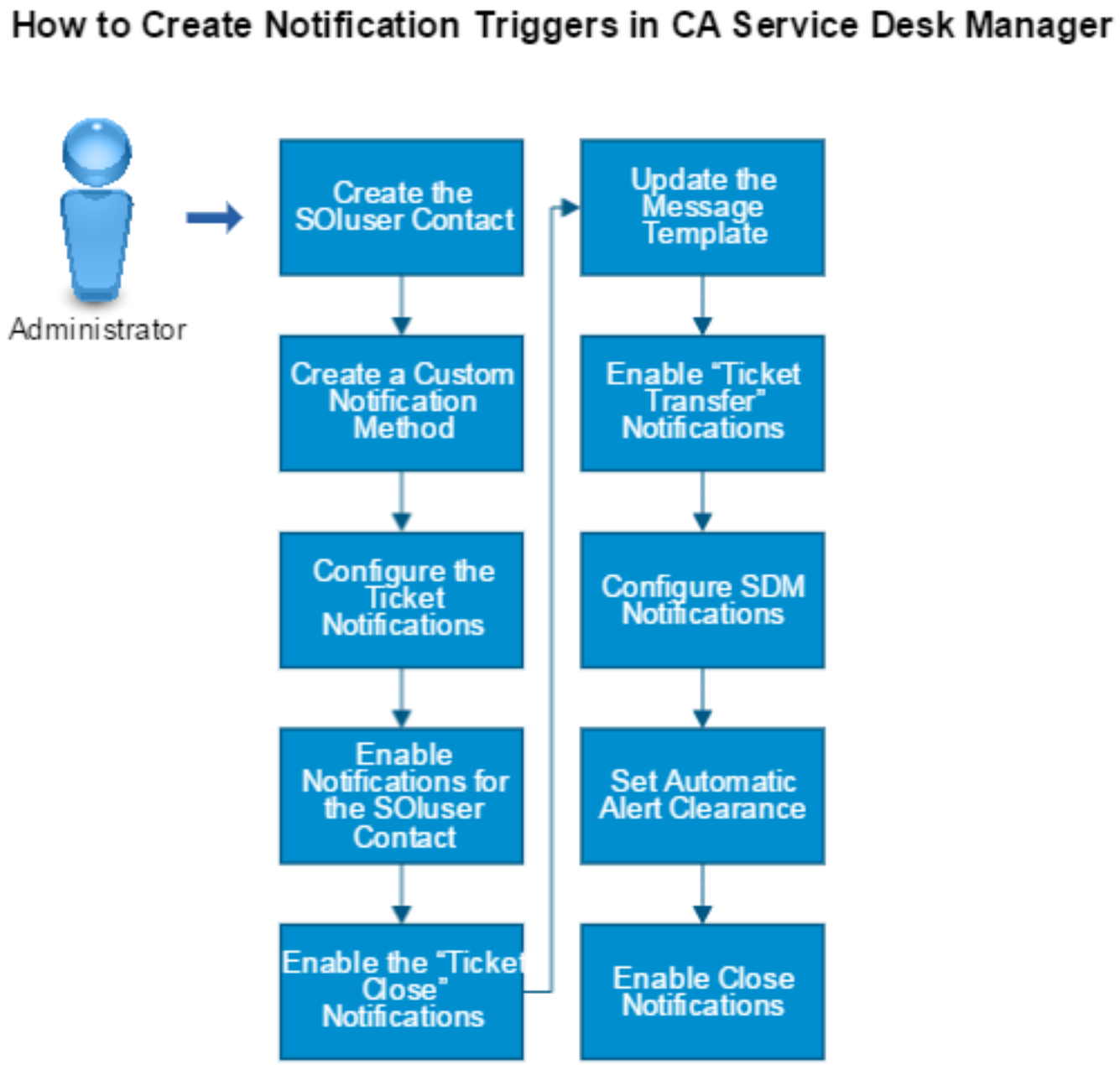
How to Create Notification Triggers in CA Service Desk

Contents

As an administrator, you can replace the CA Service Desk Manager polling functionality with notification triggers. These triggers allow CA Service Desk Manager to notify CA SOI when specific ticket-change events occur in CA Service Desk. The notification triggers reduce the previous polling load on the CA Service Desk Manager and increase performance.

Use this scenario to guide you through the process:

Figure 15: Create Notification Triggers in SDM



1. Create the SOluser contact on the CA Service Desk server.
2. Create a custom CA Service Desk notification method.
3. Configure the CA Service Desk ticket notifications that CA SOI receives:
   a. Enable the notifications for the SOluser contact.
   b. Create the SOluser condition macro.
   c. Enable the CA Service Desk notifications for the Ticket Close actions.
   d. Update the message template to send the status.

   e. Enable the CA Service Desk notifications for the Ticket Transfer actions.
4. Configure the CA Service Desk notifications.
5. Set automatic alert clearance in CA SOI.
6. Enable the close notifications.

Create the SOIuser Contact on the CA Service Desk Server

You create a CA SOI contact that is named SOIuser on the CA SOI server to communicate with the CA Service Desk server.

Follow these steps:

1. Navigate to your CA Service Desk Manager server home page with the following URL:
   `http://<service_desk_server>/CAisd/pdmweb.exe`
2. Click the Service Desk tab.
3. Click File, New Contact.
   The Create New Contact window opens.
4. Depending on the CA Service Desk version, enter SOIuser in both the Last Name and User ID (or System Login) fields.
5. Select Active from the Status drop-down list.
6. Select the appropriate Access Type from the Access Type drop-down list. This setting allows tickets to be assigned to the user.
7. Click Save.

Create a Custom CA Service Desk Notification Method

You create a custom notification method for CA Service Desk to send notifications to CA SOI. This method communicates CA Service Desk ticket changes to CA SOI.

Follow these steps:

1. From the CA Service Desk Manager server home page, click the Administration tab.
2. Expand the Notifications folder and click Notification Methods.
   The Notification Method List opens.
3. Click Create New.
   The Create New Notification Method window opens.
4. Enter the following information:
   a. Enter SOI_Notification in the Symbol field.
   b. (Optional) Enter a description for the notification method.
   c. Specify the notification method as follows:
      - For Windows: Enter NotifySOI.bat.
      - For Solaris/Linux: Enter the full path to the NotifySOI script, such as
        `/opt/CA/ServiceDesk/bin/NotifySOI.sh`

        NOTE
        Verify that the NotifySOI file is in the Service_Desk_Home/bin folder. Also, do not select the Write to File check box because the integration uses web services instead of reading the information from a file.
   d. Select Active from the Record Status drop-down list.
   e. Click Save.
      The notification method is created.

Configure the CA Service Desk Ticket Notifications that CA SOI Receives

When a ticket associated with a CA SOI alert changes, CA Service Desk can send CA SOI notifications. The notifications update the tickets associated alert in CA SOI to reflect changes to the ticket. For example, you can configure the CA Service Desk integration so that a ticket closure triggers an automatic ticket closed notification. This notification triggers CA SOI to clear the associated alert. Similarly, when a ticket is transferred, a ticket transfer notification triggers CA SOI to update the troubleshooter information for the associated alarm.

These notifications use the CA Service Desk keywords that must match the keywords that you set in the CA SOI integration. By default, the keyword for the close action is Closed. The keyword for the transfer action is Transfer (in both CA Service Desk and the CA SOI integration configuration).

> **NOTE**
> The keywords are case-sensitive.

Enable Notifications for the SOIuser Contact

You enable the CA Service Desk notifications for the CA SOI SOIuser contact on the CA Service Desk server.

Follow these steps:

1. From the CA Service Desk Manager server home page, click the Service Desk tab.
2. Select Search, Contacts.
3. Enter SOIuser in the Last Name field and click Search.
4. Click the SOIuser contact under the Name column.
   The Detail window for the SOIuser contact opens.
5. Click Edit and click the Notification tab.
6. For each of the Notifications drop-down list types (Low, Normal, High, and Emergency), select the SOI_Notification from the Method drop-down list.
7. Click Save.

Create the SOIuser Condition Macro

You create a macro that verifies if an affected user is the SOIuser you created in the previous procedure. You will reuse this macro as a condition when you define the CA Service Desk notifications in subsequent procedures.

Follow these steps:

1. From the CA Service Desk Manager server home page, click the Administration tab.
2. Expand the Events and Macros folder and click Macros.
   The Macro List page opens.
3. Click Create New.
4. Complete the following fields and click Continue:
   - Symbol: Enter Check if SOIuser.
   - Macro Type: Select Site-defined Condition from the drop-down list.
   - Object Type: Select Request/Incident/Problem from the drop-down list.
5. Enter the following Macro Description: Check if the affected user is SOIuser.
6. Click Save.
   The Check if SOIuser Macro Detail window opens.
7. Click Add condition in the Conditions tab.
8. Perform the following actions:
   a. Complete the following fields:

- Sequence: Enter 10.
- Select an Attribute: Click the link, navigate to, and select Affected End User.
- Select Attribute or Data Value: Select Data Value from the drop-down list.

b. Click Data Value.
The Contact Search window opens.
c. Enter SOIuser in the Last Name column and click Search.
d. Click SOIuser.
e. SOIuser appears in the Data Value field.

9. Click Save.

Enable CA Service Desk Notifications for the "Ticket Close" Actions

You can configure CA Service Desk to send a notification to CA SOI when a CA SOI alert ticket closes. This notification causes CA SOI to clear the associated alert. You assign the "Check if SOIuser" macro you created in the previous step as a condition.

Follow these steps:

1. From the CA Service Desk Manager server home page, click the Administration tab.
2. Expand the Notifications folder then click Notification Rules.
3. Click Create New.
4. Complete the following fields:
   – Symbol: Enter SOI Close.
   – Object Type: Select Request/Incident/Problem from the drop-down list.
5. Click Save & Continue.
   The Update Notification Rule page opens.
6. Enter the following Description: Notify CA SOI on close if affected user is SOIuser.
7. Add the "Check if SOIuser" condition:
   a. Click Condition.
   The Macro list window opens.
   b. Locate the "Check if SOIuser" macro or click Show Filter and enter %Check if SOIuser% in the Symbol field to search for it.
   c. Click Check if SOIuser.
8. Update the Message Template:
   a. Click Message Template.
   b. Locate and click Default Close Requested message template for request/incident/problem.

      NOTE
      You can also click Show Filter and enter %Default Close Requested% in the Symbol Field to narrow the list.
9. Update the Contacts List:
   a. Click the Contacts tab and click Update Contacts.
   b. Enter SOIuser in the Last Name field and click Search.
   c. The Notification Recipients - Update page opens.
   d. Add the SOIuser contact from the Contacts Match list to the Notification Recipients list and click OK.
10. Click Save.

    NOTE
    The SOI Close Notification Rule now displays until you close the window and then click the Close Activity Notification again.

    Your completed SOI Close Notification Rule Detail window looks like the following graphic:

SOI Close Notification Rule Detail

Symbol	Object Type	Record Status
SOI Close	Request/Incident/Problem	Active

Description

Notify SOI on Close if Affected End user is SOIuser

Condition	Message Template
Check if SOIuser	Default Close Requested message template for request/incident/problem

Last Modified Date	Last Modified By
01/18/2013 02:24 pm	ServiceDesk

1. Object Contacts | 2. Contacts | 3. Contact Types | 4. Related Activity Notifications

Contact List — Search | Show Filter | Clear Filter | Update Contacts | Export($)

1 contact found

Name	Contact Type	Access Type	Contact ID	User ID	Telephone Number	Status
SOIuser	Analyst	Administration		SOIuser		Active

1 contact found

11. Click Close Window.
12. Click the Administration tab and expand the Notifications folder.
13. Click Activity Notifications.
14. Locate and click the activity notification named Close or search for it.
15. Click the Notification Rules tab and click Update Notification Rules.
16. Enter SOI Close in the Symbol field and click Search.
17. Move SOI Close from the Notification Rules Available list to the Notification Rules Selected list.
18. Click OK.
    Your completed Close Activity Notification Detail window looks like the following graphic:

Close Activity Notification Detail

Symbol	Code	Internal	Record Status
Close	CL	No	Active

Description

close request/incident/problem/change/issue

Related Ticket Activity

No

Activity Valid for

Requests/Incidents/Problems	Change Orders	Issues	Managed Surveys
YES	YES	YES	NO
Knowledge Documents	**Knowledge Document Comments**	**Knowledge Report Card**	**Assistance Session**
NO	NO	NO	NO

Contacts	Configuration Items
NO	NO

Object Type	Last Modified Date
Requests/Incidents/Problems	07/26/2012 04:29 pm

Requests/Incidents/Problems Tabs

1. Notification Rules | 2. Survey | 3. Events

Notification Rules List — Search | Show Filter | Clear Filter | Update Notification Rules | Export

1-2 of 2

Symbol	Description	Auto Notification	Condition	Active
Default Close Notification Rule for request/incident/problem	Always notifies the attached Object Contacts, Contacts and Contact Types.	Yes		Active
SOI Close	Notify SOI on Close if Affected End user is SOIuser	Yes	Check if SOIuser	Active

1-2 of 2

Consider the following situations:

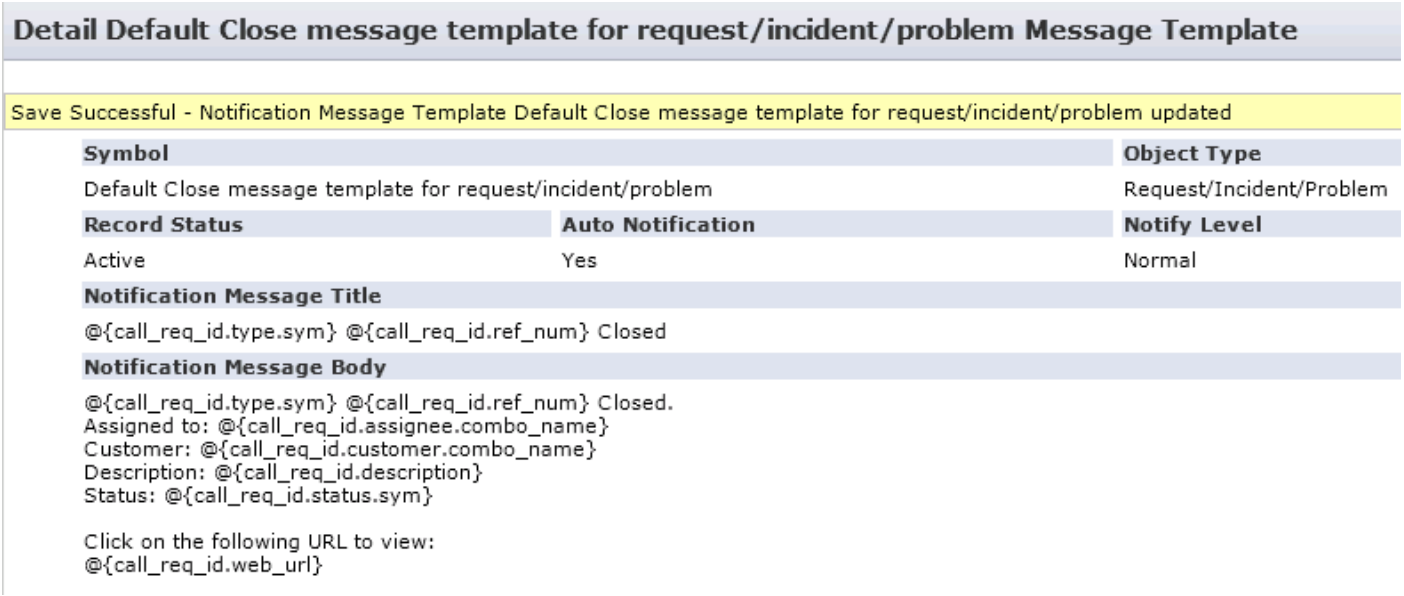
- Depending on the notification requirements, you can have one or more Close Notifications that are defined for the Close Activity.
- For every notification rule set in the Rules List, the SOIuser is notified. If you do not want this behavior, add the following condition: If the affected user is the SOIuser, then do not send the notification for every rule that is associated with the Close Activity.
- If you use the default rule, then you cannot add a condition to it. If you want to add a condition, create a rule that is identical to the default except attach the Check if SOIuser condition.

Update the Message Template to Send the Status

You now update the Default Close message template.

Follow these steps:

1. From the CA Service Desk Manager server home page, click the Administration tab.
2. Expand the Notifications folder and click the Message Templates folder.
3. Click Show Filter.
4. Enter the following text in the Symbol Field and click Search:
   ```
   Default Close%
   ```
5. Select the Default Close message template for request/incident/problem.
   The Detail window opens.
6. Click Edit.
7. In the Notification Message Body field, locate the following entry:
   ```
   Description: @{call_req_id.description}
   ```
8. Add the following text after the Description entry:
   ```
   Status: @{call_req_id.status.sym}
   ```



9. Click Save.
10. If you are using the Closed_Requested status code, then perform Steps 1 -9 for the Default Close Requested message template for request/incident/problem.

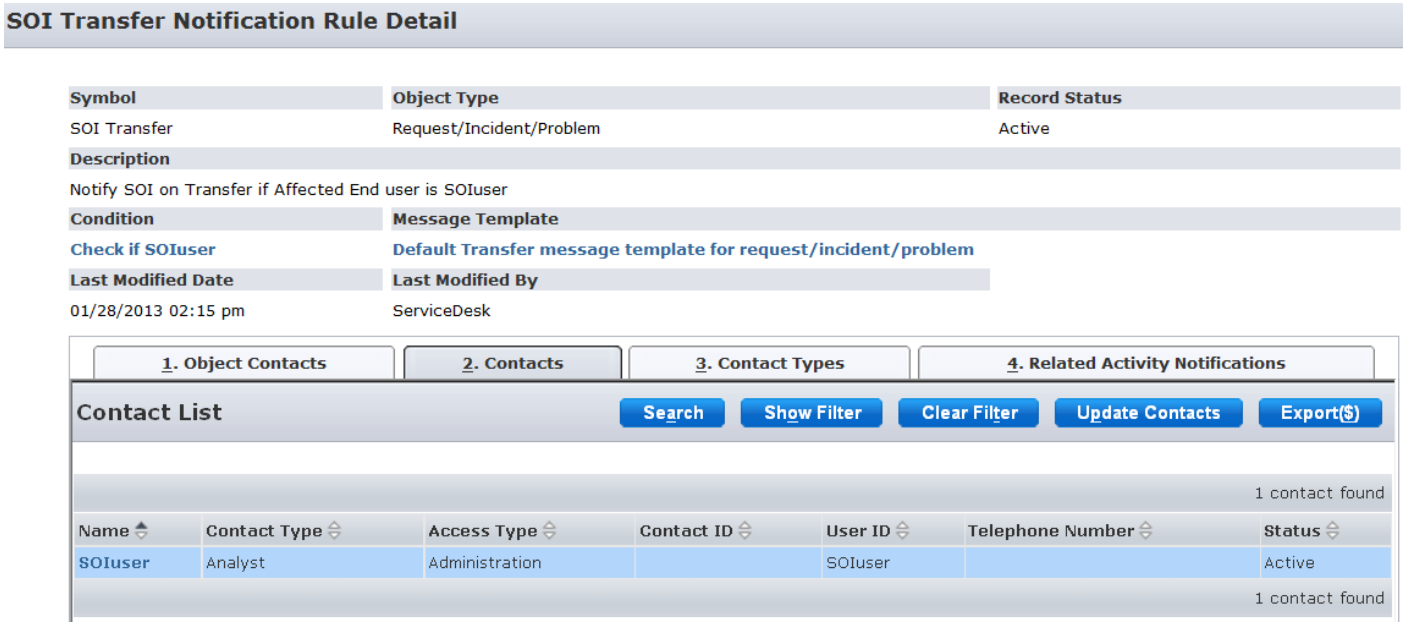
Enable CA Service Desk Notifications for the "Ticket Transfer" Actions

You can configure CA Service Desk to send a notification to CA SOI when a CA SOI alert ticket is transferred to a new Assignee. A ticket transfer notification causes CA SOI to set the owner of the ticket to the new assignee.

Follow these steps:

1. From the CA Service Desk Manager server home page, click the Administration tab.
2. Expand the Notifications folder and click Notification Rules.
3. Click Create New.
4. Complete the following fields and click Save & Continue:
   a. Symbol: Enter SOI Transfer.
   b. Object Type: Select Request/Incident/Problem.
5. Enter the following Description: Notify SOI on transfer if the affected user is SOIuser
6. Add the Check if SOIuser condition:
   a. Click Condition.
   b. Locate the Check if SOIuser macro or click Show Filter and enter %Check if SOIuser% in the Symbol field to search for it.
   c. Click Check if SOIuser.
   The Condition field populates with Check if SOIuser.
7. Update the Message Template:
   a. Click Message Template.
   b. Locate and click Default Transfer message template for request/incident/problem

   NOTE
   You can also click Show Filter and enter %Default Transfer% in the Symbol Field to narrow the list.
8. Update the Contacts List:
   a. Click the Contacts tab and click Update Contacts.
   b. Enter SOIuser in the Last Name field and click Search.
   c. Add the SOIuser contact from the Contacts Match list to the Notification Recipients list and click OK.
9. Click Save.
   Your completed SOI Transfer Notification Rule Detail window looks like the following graphic:

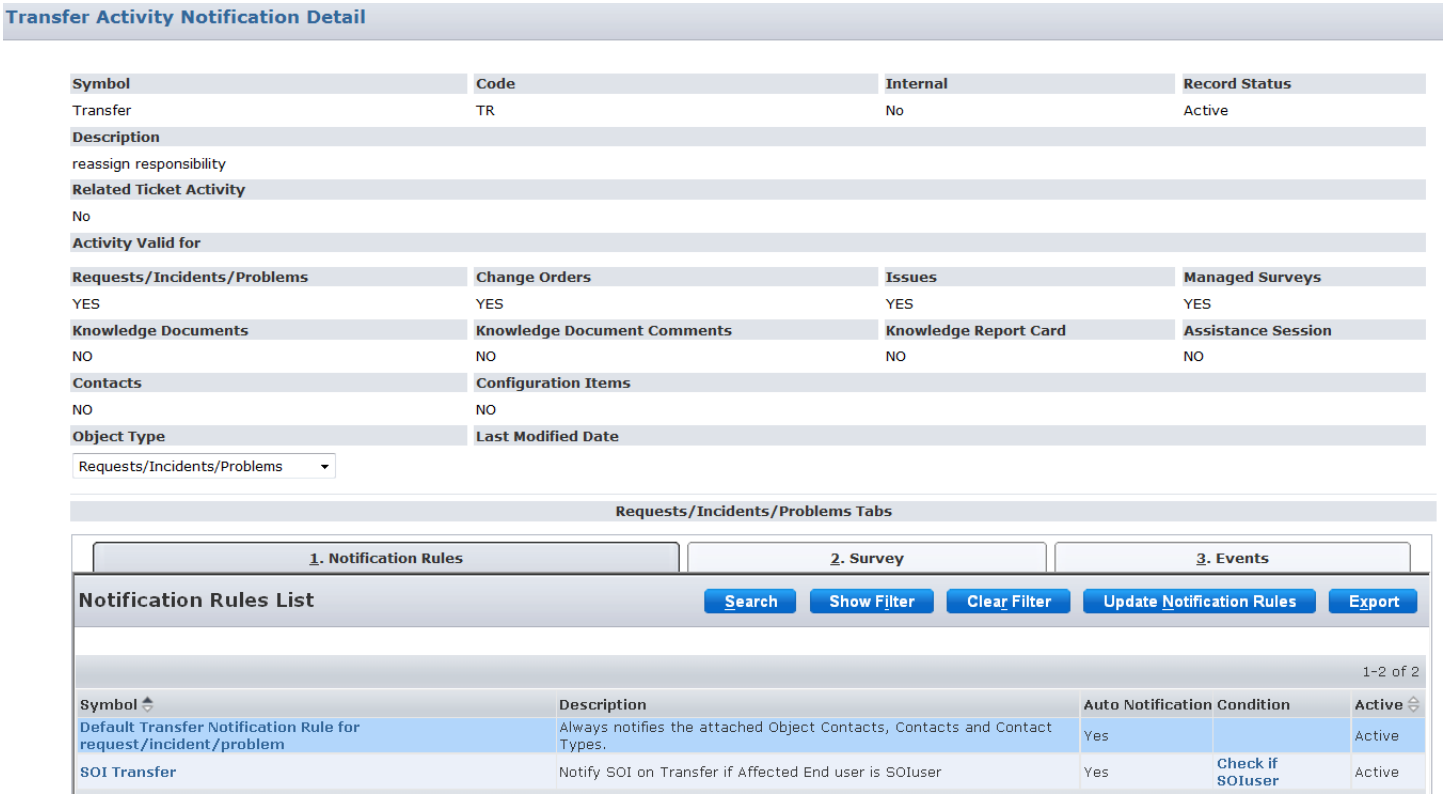


10. Click Close Window.

11. Click the Administration tab and expand the Notifications folder.
12. Click Activity Notifications.
13. Locate and click the activity notification named Transfer or use the search.
14. Click the Notification Rules tab and click Update Notification Rules.
15. Enter SOI Transfer in the Symbol field and click Search.
16. Move SOI Transfer from the Notification Rules Available list to the Notification Rules Selected list.
17. Click OK.

NOTE
The SOI Transfer Notification Rule does not display until you click Close Window and then click the Transfer Activity Notification again.

Your completed Transfer Activity Notification Detail window looks like the following graphic:



Consider the following situations:

- Depending on the notification requirements, you may have one or more Transfer Notifications that are defined for the Transfer Activity.
- For every notification rule set in the Rules List, the SOIuser is notified. If you do not want this behavior, add the following condition: If the affected user is the SOIuser, then do not send the notification for every rule that is associated with the Transfer Activity.
- If you use the default rule, then you cannot add a condition to it. If you need to add a condition, create a rule that is identical to the default except attach the Check if SOIuser condition.

Configure CA Service Desk Notifications

When the assignee value on a CA Service Desk ticket changes, the CA Service Desk Manager notification system notifies CA SOI of the change by calling the INSTALL_DIR\bin\NotifySOI.bat script. The script then calls CA SOI through the CA SOI Web Services to set the CA SOI assignee on the alert to the same value as the CA Service Desk ticket.

A CA SOI-created ticket can cause CA SOI to clear an alert when the ticket status changes to the configured (on the SOI Console Help Desk Configuration dialog) status. CA SOI clears alerts only if the Auto Clear Alert check box is selected.

Follow these steps:

1. On the CA SOI installation image, copy the contents from the Disk1\Integrations\SOI-CAServiceDesk folder to the CA Service Desk Manager server in the SDM_HOME\bin directory.
2. Edit the ConfigSOI.bat file and modify the following lines:
   a. SDM_HOME: If the path is different from the default path.
   b. JAVA_HOME: If the path is different from the default path.
3. Locate and run the SDM_HOME\bin and execute the ConfigSOI.bat utility.
4. Perform the following actions:
   a. Set the installation folder to SDM_HOME\bin.
   b. Set the Java home folder, but do not include the bin folder.
   c. Click Continue.
   d. Complete the CA SOI configuration information.
   e. Use the SA Manager port for the SOI Port number.
5. Click Save.
   The utility generates a file named soi_env.bat in the SDM_HOME folder.
6. Click Test to the test the connection to CA SOI from the CA Service Desk Manager server.

   NOTE
   If the test fails, view your settings and verify that the SDM server can access the SA Manager server.
7. Exit the utility.

Set Automatic Alert Clearance in CA SOI

The CA SOI Auto clear alert option is available on the Operations Console. Select Tools, Help Desk Configuration. In the General tab, you see the Auto clear alert option.

By default, CA SOI automatically clears alerts by polling CA Service Desk Manager. You can now automatically clear alerts by setting up CA Service Desk Manager Notifications. If you use SDM Notifications, disable the polling (by unchecking the Enable polling checkbox. Using SDM Notifications instead of polling improves the performance and reduces the SDM server load.

You have two ways of letting CA SOI accomplish the auto clear function and assignee synchronization:

1. **Not recommended:** Polling, which can negatively affect the CA Service Desk Manager performance.
2. **Recommended:** CA Service Desk Manager notifications. By default, CA SOI keeps the SOI Alert Asignee synchronized with the CA Service Desk ticket assignee. The notification method does not affect the CA Service Desk Manager performance. Disable the polling as described in the following procedure.

   NOTE
   When submitting a ticket from CA SOI Operations Console, set the affected end user to SOIuser.

Follow these steps:

1. On the Operations Console, select Tools, Help Desk Configuration.
2. In the General tab, select the Auto clear alert option and select one or more status conditions using the arrow buttons.
3. (Optional) Select the "Auto change trouble ticket status when alert is cleared" option and select a ticket status.

NOTE
If you shut down or remove a connector, CA SOI automatically changes the cleared alerts to the selected status. You cannot undo this operation.

4. Select or clear the Enable Polling option.
5. Click Save.

Enable Close Notifications

CA SOI supports all CA Service Desk Manager statuses. If CA SOI is configured to use the Problem-Fixed, Resolved, or Close-Requested close statuses, then disable the Make Active attribute.

Follow these steps:

1. In CA Service Desk Manager, click the Administration tab.
2. Expand the following folders: Service Desk, Requests/Incidents/Problems.
3. Click the Status folder.
4. Click the Resolved link.
   The Request/Incident/Problem Status Detail page opens.
5. Click Edit.
6. Clear the Make Active check box.
7. Click Save.
8. Repeat Steps 3 - 7 for all the statuses you selected in the topic Set Automatic Alert Clearance in CA SOI.

NOTE
If you clear the polling option in the Set Automatic Alert Clearance in CA SOI procedure, for each status that you want to trigger, configure the status in CA Service Desk Manager.

Bypass CA SOI Polling for Ticket Assignees

If you experience a degraded CA Service Desk performance, you can bypass the CA SOI polling for ticket assignees to reduce the number of web service calls into CA Service Desk.

Follow these steps:

1. On the Dashboard Administration tab, edit the Help Desk configuration page for the SA Manager and click Save.
2. On the SA Manager node, edit the following file:
   <SOI_HOME>\tomcat\custom\svcdesk-config.xml
   This file now has the following entry:
   ```
   <svcdsk-poll-assignee>true</svcdsk-poll-assignee>
   ```
3. Set the parameter to "false" instead of "true".
4. Restart the CA SAM Application Server service.

Review CA Spectrum and CA Service Desk Ticket Synchronization

The CA Spectrum connector sends information about the CA Service Desk tickets that are associated with CA Spectrum alerts and root cause CIs to CA SOI. CA SOI processes CA Spectrum ticket information as follows:

- CA SOI can automatically close any ticket that is opened after an alert has cleared. To enable this option, see Review Alert and Ticket Closure Considerations.
- If no CA SOI ticket exists, CA SOI uses and displays the ticket that is associated with the CA Spectrum alert.
- If a CA SOI escalation has created a ticket for the alert, CA SOI treats a ticket from CA Spectrum as follows:

  - The CA Spectrum ticket is updated in the CA Service Desk comment log to indicate that it is a duplicate of the ticket created in CA SOI.
  - The CA SOI ticket is also updated in the comment log to note the duplicate ticket in CA Spectrum.
  - The duplicate tickets are not updated when other fields require updating. Only the main ticket is updated.
  - If an automatic ticket status change occurs as a result of the alert closure, a comment is logged in the duplicate ticket. The comment indicates the CA SOI ticket status change.

This functionality applies to all CA Spectrum versions that the CA Spectrum connectors support.

Review CA Service Desk Integration Troubleshooting

If you cannot connect to CA Service Desk or create CA Service Desk tickets in CA SOI, do the following to troubleshoot CA Service Desk connection problems:

- Verify that the connection settings are correct on the Help Desk Configuration page of the Administration UI. Click Test to test the connection.
- If CA Service Desk is configured for SSL, verify that you selected the SSL check box. The integration does not work if this check box is not synchronized with the CA Service Desk SSL configuration.
  For more information, see Export CA Service Desk SSL Certificate.
- Try to access the following CA Service Desk URL independent of CA SOI:

  ```
  http://<ServiceDeskServer>:<ServiceDeskPort>/axis/services/
  USD_R11_WebService?wsdl
  ```

  NOTE
  The default CA Service Desk port is 8080.

  If you cannot reach this URL, do the following:
  - Verify that the Service Desk Web Services component is installed on the CA Service Desk system. This component must be installed for the integration to work.
  - See the CA Service Desk documentation.
- If you are using CA Service Desk r12.1 that has been upgraded from a previous release, clear the CA Service Desk browser cache.

How to Configure a ServiceNow Integration

As an administrator, you integrate CA SOI with ServiceNow for alert and incident management. CA Process Automation is a workflow management product with enterprise automation capabilities. The ServiceNow connector provides CA Process Automation with the ServiceNow operators and workflows that use the CA Catalyst integration technology to connect with ServiceNow.

After an installation and configuration, the CA Process Automation ServiceNow Gateway performs the following functions for CA SOI:

- ServiceNow incidents are automatically created based on the CA SOI alert escalation policy or manually submitted from the Operations Console.
- When CA SOI sends an alert data through the connector, the ServiceNow system generates an incident with an identifier.
- The identifier is updated in the CA SOI alert and associates the incident with the alert. The alert data provides a description of the problem for the incident.
- After the associated problem is resolved, the CA SOI alert and ServiceNow incident can be optionally cleared automatically.
- When the CA SOI alert is cleared, the ServiceNow incident can be optionally closed.
- An alert annotation that you create for a CA SOI alert is reflected on a ServiceNow incident. Similarly, ServiceNow incident work notes update entry is reflected in the CA SOI alert annotation table.

This section describes how to install and configure the CA Process Automation ServiceNow Gateway and use it to integrate CA SOI with ServiceNow.

For detailed information about how to implement this integration, see the *Integration Guide* packaged with the CA Connector for ServiceNow. Use the CA SOI process definition files packaged with the connector when establishing the integration, not the files packaged with CA SOI.

> **NOTE**
>
> For the most current information about the connector, help desk, and CA Process Automation versions that CA SOI supports, see Software Support.

How to Configure Other Help Desk Product Integrations

Contents

As an administrator, you can configure a custom integration with any third-party help desk product using the Universal Help Desk API. The API provides a Java source template file in which you can code callback routines that CA SOI calls to acquire information from the help desk. The Universal Help Desk API can interface with help desk products through web services or local calls.

Use this scenario to guide you through the process:

1. Program the UniversalHelpDesk.jar file.
2. Configure the Universal Help Desk connection.
3. Work with the configured help desk integration.

Program the UniversalHelpDesk.jar File

For custom third-party help desk integrations, CA SOI acquires help desk information from the UniversalHelpDesk.jar file at SOI_HOME\tomcat\lib. Create a custom-compiled UniversalHelpDesk.jar file and place it at SOI_HOME\tomcat\lib. When complete, the file must contain a class that is called UniversalHelpDesk. The UniversalHelpDesk class contains the routines listed in this section. The package name is com.external.UniversalHelpDesk.

After you complete coding and creating your UniversalHelpDesk.jar, add it to the SOI_HOME\tomcat\lib and restart the CA SAM Application Server service. Then configure the Universal Help Desk connection.

int HDInitialize Routine

This routine is called when the SA Manager starts and should contain any initialization required such as binding to a web service through the SOAP API and initializing memory. It should return a zero if the initialization was successful and non-zero if it was unsuccessful. If the routine returns a non-zero value, the HDGetError routine is called to obtain the error text.

Code this routine as follows:

```
int HDInitialize (Map<CONNECTINFO, String> connectInfo)
```

CONNECTINFO is the following enum:

```
public static enum CONNECTINFO
{
       TYPE, SERVER, PORT, USER, PASSWORD,
       SSL, CLEARTICKET, CLEARALARM, POLL, POLLINTERVAL,
       TICKETSTATUS
}
```

Void HDUninitialize Routine

This routine contains any cleanup that is required when the SA Manager shuts down.

Code this routine as follows:

```
Void HDUninitialize()
```

int HDLogin Routine

This routine is called to log in to the help desk. The user name and password configured in the Administration UI Help Desk Configuration page is passed to the routine.

The routine returns the identifier for the help desk session. The identifier is cached and passed to all other callback routines. If the connection fails, the routine returns a value of -1, in which case the HDGetError routine is called.

Code this routine as follows:

```
Int HDLogin(String username, String password)
```

- *username*
  Specifies the user name to log in to the help desk, which you specify in the Administration UI Help Desk Configuration page.
- *password*
  Specifies the password to log in to the help desk, which you specify in the Administration UI Help Desk Configuration page.

Void HDLogout Routine

This routine logs out from the help desk session.

Code this routine as follows:

```
Void HDLogout()
```

Boolean HDIsConnected Routine

This routine returns whether the help desk connection is valid. If the help desk connection is valid, the routine returns a value of true. Otherwise, the routine returns a value of false.

Code this routine as follows:

```
Boolean HDIsConnected()
```

String HDCreateTicket Routine

This routine creates the help desk ticket when a CA SOI user manually generates an alert ticket or when a create ticket escalation policy action runs.

Code this routine as follows:

```
String[] HDCreateTicket(Map<String, String> properties)
```

Param Properties:
Collection of name value pairs of properties to set. The syntax of the keys in the collection is as follows:

- **param#**_name for the names
- **param#**_value for the corresponding values
  where # is a number from 0 to the max number of name / value pairs.

Return Values:

- **String[5]**
  Contains the ticket information in a string array.
- **String[0]**
  Contains the ticket handle (if applicable).
- **String[1]**
  Contains the ticket number.
- **String[2]**
  Contains the ticket type (Request (R), Problem (P), Incident (I)).
- **String[3]**
  Contains the URL used to start directly to the ticket.
- **String[4]**
  Contains the error text explaining the failure if the ticket number is empty.

The properties parameter lists the name value pairs representing fields and properties CA SOI uses to populate the help desk ticket. The list contains properties from the following areas:

- CA SOI exposed runtime parameters (such as $ASSIGNED)
- Attributes the administrator assigns in the CA SOI Action Editor dialog.

HDTicketProperties is defined as the following Java enum:

```
public static enum HDTicketProperties
{
       // Common Help Desk properties
       NUM_OF_PROPS, // Number of properties in the collection
       TICKET_TYPE,  // Type of ticket: Request, Incident, Problem etc.
       ASSIGNEE,     // To whom the ticket is assigned
       DESCRIPTION,  // Ticket description
       GROUP,        // Group that the ticket belongs to
       IMPACT,       // Impact of alert on asset
       PRIORITY,     // Priority of alert
       SEVERITY,     // Severity of alert
       SUMMARY,      // Ticket summary
       URGENCY,      // Urgency of the alert

       // SSA 2.0 exposed properties
       ACKNOWLEDGED,      // True if the SSA Alert has been acknowledged
       ALARM_ID,          // SSA alert identifier
       CONNECTOR_NAME,    // Name of the connector from which the SSA alert
                             originated
       CREATION_DATE,     // Creation date of the SSA alert (in XML Schema
                             format) e.g. 2009-08-10T12:00:00-05:00
       DETAIL,            // Detail from the SSA alert
```

```
        EVENT_OCCURRED,    // Date (in XML Schema format) of the event occurrence
        EVENT_SOURCE,      // Source of the event (Domain manager name)
        IP_ADDRESS,        // IP Address of the affected CI
        MODEL_NAME,        // Name of CI/Service in SSA
        MODEL_CLASS,       // Class of CI/Service in SSA
        MODEL_DESCRIPTION, // Description associated with SSA CI/Service
        MODEL_FAMILY,      // Family of SSA CI/Service
        ROOT_CAUSE,        // SSA Alert which has the greatest impact on the SSA
                              CI/Service
        VENDOR_NAME,       // Name of vendor of CI (if any)
}
```

String HDCreateAnnouncement Routine

This routine creates the help desk announcement when a create announcement escalation policy action runs. Implement this routine only if the third-party help desk does supports announcements.

Code this routine as follows:

```
String[] HDCreateAnnouncement(Map<HDTicketProperties, String> properties)
```

Return Values:

- **String[5]**
  Contains the announcement information in a string array.
- **String[0]**
  Contains the announcement handle (if applicable).
- **String[1]**
  Contains the announcement number.
- **String[2]**
  Contains the announcement type (Request (R), Problem (P), Incident (I)).
- **String[3]**
  Contains the URL used to start directly to the announcement.
- **String[4]**
  Contains the error text explaining the failure if the announcement number is empty.

The properties parameter lists the name value pairs representing fields and properties that CA SOI used to populate the help desk announcement. The list contains properties from the following areas:

- CA SOI exposed runtime parameters (such as $ASSIGNED)
- Attributes the administrator assigned in the CA SOI Action Editor dialog.

int HDCloseTicket Routine

If you configure the option to close a ticket automatically when its associated alert clears, then this routine closes the help desk ticket.

If the ticket is successfully closed, the routine returns a value of zero. Otherwise, the routine returns a non-zero value. If the routine returns a non-zero value, the HDGetError routine is called to obtain the error text.

Code this routine as follows:

```
int HDCloseTicket(String ticketHandle, String ticketNum)
```

- *ticketHandle*
  Specifies the handle of the ticket to be closed, if applicable.
- *ticketNum*

Specifies the number of the ticket to be closed.

int HDUpdateTicket Routine

This routine is called when CA SOI alert information changes and the associated help desk ticket requires updating with the most recent values.

If the ticket updates successfully, the routine returns a value of zero. Otherwise, the routine returns a non-zero value and the HDGetError routine is called to obtain the error text.

Code this routine as follows:

```
int HDUpdateTicket(String ticketHandle, String ticketNum, String property, String value)
```

- *ticketHandle*
  Specifies the handle of the ticket to update, if applicable.
- *ticketNum*
  Specifies the number of the ticket to update.
- *property*
  Specifies the name of the property that has a value change.
- *value*
  Specifies the new value of the property.

int HDIsTicketClosed Routine

This routine is called during the polling cycle to query if the specified help desk ticket has been closed. If a true value is returned and the Auto clear alert check box is selected on the Help Desk Configuration page, the associated alert is cleared in CA SOI.

Code this routine as follows:

```
int HDIsTicketClosed(String ticketHandle, String ticketNum)
```

- *ticketHandle*
  Specifies the handle of the ticket to query, if applicable.
- *ticketNum*
  Specifies the number of the ticket to query.

Object HDGetTicketProperty Routine

This routine is called during the polling cycle to query the value of a particular property (such as assigned). If the help desk ticket value is different from the CA SOI alert value, the CA SOI alert value is changed to match the ticket value.

The only property that is currently supported is assigned.

The routine returns the property value as an object. A null object is returned if an error occurred.

Code this routine as follows:

```
Object HDGetTicketProperty(String ticketHandle, String ticketNum, String property)
```

- *ticketHandle*
  Specifies the handle of the ticket to query, if applicable.
- *ticketNum*
  Specifies the number of the ticket to query.
- *property*
  Specifies the name of the property value to obtain.

String HDGetError Routine

This routine is called when a non-zero value is returned from any of the specified callback routines. The routine returns the error text of the last error that occurred.

Code this routine as follows:

```
String HDGetError(int errorCode)
```

- *errorCode*
  Specifies the error code that is returned from one of the callback routines.

Configure Universal Help Desk Connection

You configure a connection to the help desk product in CA SOI.

Follow these steps:

1. Access the CA SOI Administration UI and click the Administration tab.
   The Administration Pages pane opens.
2. Expand CA Service Operations Insight Manager Configuration and the server name and click Help Desk Configuration.
   The Help Desk Configuration page opens.
3. Select Universal Help Desk in the Help Desk Type drop-down list and enter the following properties:
   - **Server**
     Specifies the name of the help desk host server.
   - **Port**
     (Optional) Specifies the port number that the desk host server uses.
   - **User**
     Specifies the user account with which to access the help desk.
   - **Password**
     Specifies the password that corresponds with the help desk user account.
   - **SSL**
     (Optional) Specifies whether to use SSL to communicate with the selected help desk application.
4. Click Test.
   A successful connection message displays if you entered valid help desk settings.
5. Click Save.
   The help desk connection is configured. CA SOI begins interacting with the help desk application through escalation policy and manual ticket generation.

How to Work with Configured Help Desk Integrations

Contents

As an operator, you can perform help desk ticket operations with an integrated help desk product.

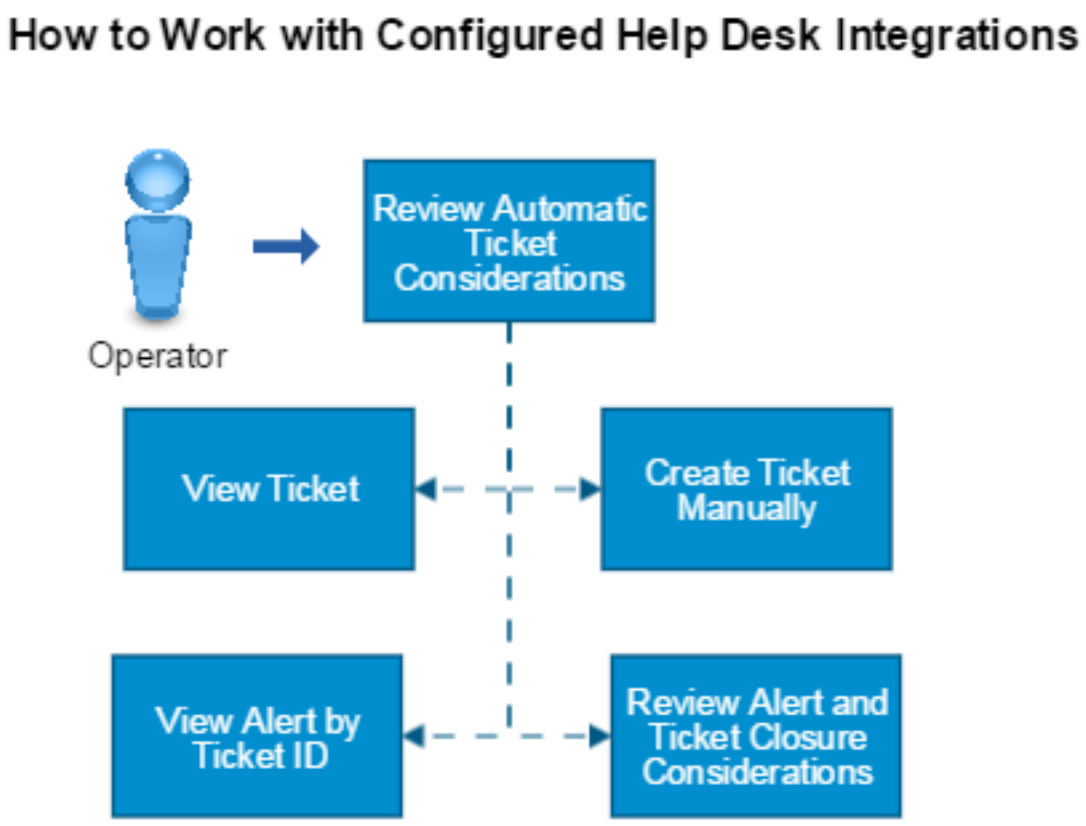
An administrator configures the connection between CA SOI and a help desk product, the integration works as follows:

> **NOTE**
> Help desk products use different terms such as a *ticket*, *incident*, or *problem* to refer to the object that the application creates to indicate a problem. This section uses the term *ticket* to refer to the created object in any integrated help desk product.

- Help desk tickets are automatically created based on CA SOI alert escalation policy or manually submitted from the Operations Console.
- The ticket number is added to the CA SOI alert and associates the ticket with the alert. The alert data provides a description of the problem for the ticket.
- (CA Service Desk only) You can synchronize the CA SOI assignee alert property with the CA Service Desk ticket Assigned property. If one property is set, the other property sets to the same value.
- Clicking the ticket number within CA SOI lets you start the help desk product in context to view ticket details.
- After the associated problem is resolved, the CA SOI alert and the help desk ticket can be synchronized. If one is cleared, the other clears automatically.
- (BMC Remedy only) An alert annotation that is created for a CA SOI alert is reflected on a BMC Remedy incident. Also, a BMC Remedy incident worklog entry is reflected in the CA SOI alert annotation table.
- (HP Service Manager only) An alert annotation that is created for a CA SOI alert is reflected on an HP Service Manager incident. Similarly, an HP Service Manager incident journal update entry is reflected in the CA SOI alert annotation table.

Use this scenario to guide you through the process:

Figure 16: how to work with configured help desk integrations



1. Configure automatic ticket creation through escalation policies.
2. Create a ticket manually.
3. View a ticket.
4. View an alert by the associated ticket ID.
5. Review alert and ticket closure considerations.

Review Automatic Ticket Creation Considerations

You can configure automatic ticket creation when a CA SOI alert is generated. Configure automatic ticket creation in any of the following ways:

- Configure global alert escalation policy with a create ticket action.
- Configure nonglobal alert escalation policy for a service or alert queue with a create ticket action.
- Update existing policy to add a create ticket action.

> **NOTE**
> For more information about creating and implementing alert escalation policy, see How to Create Escalation Policy.

When automatic ticket creation is correctly configured, a ticket is automatically created after an alert is generated that matches the escalation policy criteria. Monitor the Alerts tab of the Operations Console Contents pane for new alerts and automatic ticket generation.

> **NOTE**
> It may take up to a minute after an alert is generated for the ticket to be created and reflected in CA SOI.

After the ticket is created, the following displays in the CA SOI alert:

- The Ticket ID field is populated with the new ticket number.
- The Ticket ID field is also populated with a context-sensitive link to start the integrated help desk product and display the created ticket for the alert.

(CA Service Desk only) You can also automatically create CA Service Desk announcements in the same ways that you create tickets with a create announcement action.

Create a Ticket Manually

You can create a ticket for any CA SOI alert manually under either of the following conditions:

- You are not using automatic incident creation.
- The alert does not satisfy the escalation policy but warrants incident creation.

Follow these steps:

1. Select the alert for which you want to generate a ticket in the Alerts tab of the Operations Console.
   Alert information displays in the Component Detail pane. An empty Ticket ID field indicates that a ticket has not yet been created.
2. Right-click the alert and select Take Action.

   > **NOTE**
   > If the root cause of the alerts is the same and the situation is resolvable by one fix, then you can assign the same incident to multiple alerts

   The Take Action dialog opens.
3. Do *one* of the following:
   - Select an existing escalation policy action to submit the ticket.
   - Click Create and create a Create Ticket action in the Escalation Action Editor dialog.
4. (Optional) Select the 'Use this selection as default and do not show this dialog again' check box to hide this dialog in the future and use the default action. You can toggle this option in the Preferences dialog.
5. Click OK.
   A ticket is opened in the integrated help desk product that contains details about the alert. The ticket uses the properties specified in the create ticket action that you selected. A confirmation dialog displays the ticket ID.

   > **NOTE**
   > It may take up to a minute for the ticket to be created and reflected in CA SOI.

For more information about creating alert escalation policy, see How to Create Escalation Policy.

View a Ticket

You can start the integrated help desk product directly from the Operations Console using the context-sensitive link in the Alert Details tab to view the ticket that is associated with an alert.

Follow these steps:

1. 1. Select the alert for which you want to view an associated ticket in the Alerts tab of the Operations Console. Alert information displays in the Alert Details tab of the Component Detail pane. A populated Ticket ID field indicates that a ticket has been created for the alert.
   2. Click the link in the Ticket ID field of the Alert Details tab.

      NOTE
      If a link does not appear for the Ticket ID, check the help desk server to ensure that it is running.

      A login page opens for the help desk product.

      NOTE
      (CA Service Desk only) You can configure CA Service Desk to go directly to trouble ticket detail without prompting for the user name and password first. For more information, see Enable Automatic Links to Tickets.
   3. Enter valid credentials and click Log In.
      The help desk product displays details about the ticket that is generated for the alert. From this page, you can update and save changes directly to the ticket.
      The following fields contain CA SOI-specific information by default:
      CA Service Desk
   4. **Summary**
      Contains a subset of the Description field from the CA SOI alert.
   5. **Description**
      Contains the entire Description field from the CA SOI alert.
   6. **Priority**
      Contains the maximum priority of all of the impacted services of the alert.
   7. **Severity**
      Contains the alert severity.
   8. **Impact**
      Contains the maximum impact on all impacted services of the alert.
   9. **Root Cause**
      Contains the Root Cause of the CA SOI alert.
   10. **Configuration Item**
      Contains the instanceID of the CA SOI alert.

BMC Remedy

1. 1. **Summary**
      Contains a subset of the Description field from the CA SOI alert.
   2. **Description**
      Contains the entire Description field from the CA SOI alert.

HP Service Manager

1. **Title**
   Contains a subset of the Description field from the CA SOI alert.
2. **Description**
   Contains the entire Description field from the CA SOI alert.

View an Alert by the Ticket ID

After ticket creation, you can reference the CA SOI alert that is associated with the ticket using the Ticket ID number.

Follow these steps:

1. Note the identifier for the ticket in the help desk product.
   The following values correspond to the Ticket ID in CA SOI:
   – CA Service Desk: Ticket number
   – BMC Remedy and HP Service Manager: Incident or Case ID
2. Enter the identifier in the Filter field in the Alerts tab of the Operations Console Contents pane.
   The alert whose Ticket ID value matches the entered value displays. If the alert does not display, it may have already been cleared.

Review Alert and Ticket Closure Considerations

CA SOI alerts and help desk tickets behave as follows when an alert or ticket is cleared, or other updates occur:

- You can configure a ticket to close automatically when its associated alert is cleared in CA SOI. By default, this option is not selected and tickets are not closed when their associated alerts are cleared. Configure automatic ticket closure as follows:

  NOTE
  For more information about clearing alerts, see How to Create Escalation Policy or How to Assign and Update Alerts.

  – CA Service Desk: Select the 'Auto change trouble ticket when alert is cleared' check box on the Help Desk Configuration dialog of the Operations Console.

    NOTE
    For more information about editing the help desk configuration, see Configure Help Desk Integration.

  – **BMC Remedy**: Select the 'Auto close ticket when alert cleared' check box on the Remedy Parameters page of the RemedyHPDConfiguration or RemedyITSMConfiguration form within CA Process Automation.
  – **HP Service Manager**: Select the DoAutoClose check box on the HPSM Parameters page of the HPSMConfiguration form within CA Process Automation.
- You can configure a CA SOI alert to clear automatically when its associated ticket closes. By default, this option is not selected and alerts are not cleared automatically when their associated tickets are closed. Configure automatic alert closure as follows:
  – **CA Service Desk**: Select the 'Auto clear alert' check box on the Help Desk Configuration dialog of the Operations Console.
  – **BMC Remedy**: Select the 'Auto clear alert' check box on the Remedy Parameters page of the RemedyHPDConfiguration or RemedyITSMConfiguration form within CA Process Automation.
  – **HP Service Manager**: Select the DoAutoClearAlert check box on the HPSM Parameters page of the HPSMConfiguration form within CA Process Automation.
- The CA Service Desk integration lets you specify a CA Service Desk Status to use for the automatic ticket and alert closure. Therefore, you can assign any status to a ticket after its associated alert clears, and you can clear an alert when its associated ticket attains a specific status.
- After ticket creation, clearing an alert is the only change that is communicated to the ticket except for the following changes:

  - The Assigned field changes in CA Service Desk if you assign the alert to a CA SOI user after ticket generation. Other changes to an alert such as acknowledgements are not reflected in CA Service Desk.
  - Alert annotations are synchronized with BMC Remedy incident worklog entries and HP Service Manager Journal Update entries.
- Do not delete a ticket manually in the integrated help desk product that is associated with an alert unless it is necessary for repair, because it affects synchronization between CA SOI and the help desk. The following issues occur when you manually delete a ticket:
  - If you delete a ticket, the change is not reflected in CA SOI. Manually reset the alert Ticket ID value to keep the systems synchronized.
  - If you delete the Ticket ID for an alert, the associated ticket is not automatically deleted. Manually delete or close the ticket in the help desk product to keep the systems synchronized.

High Availability Implementation

This section describes how to implement CA SOI in a high availability environment.

Implementation in a Microsoft Cluster Server Environment

This section describes how to install CA SOI in a cluster environment for high availability:

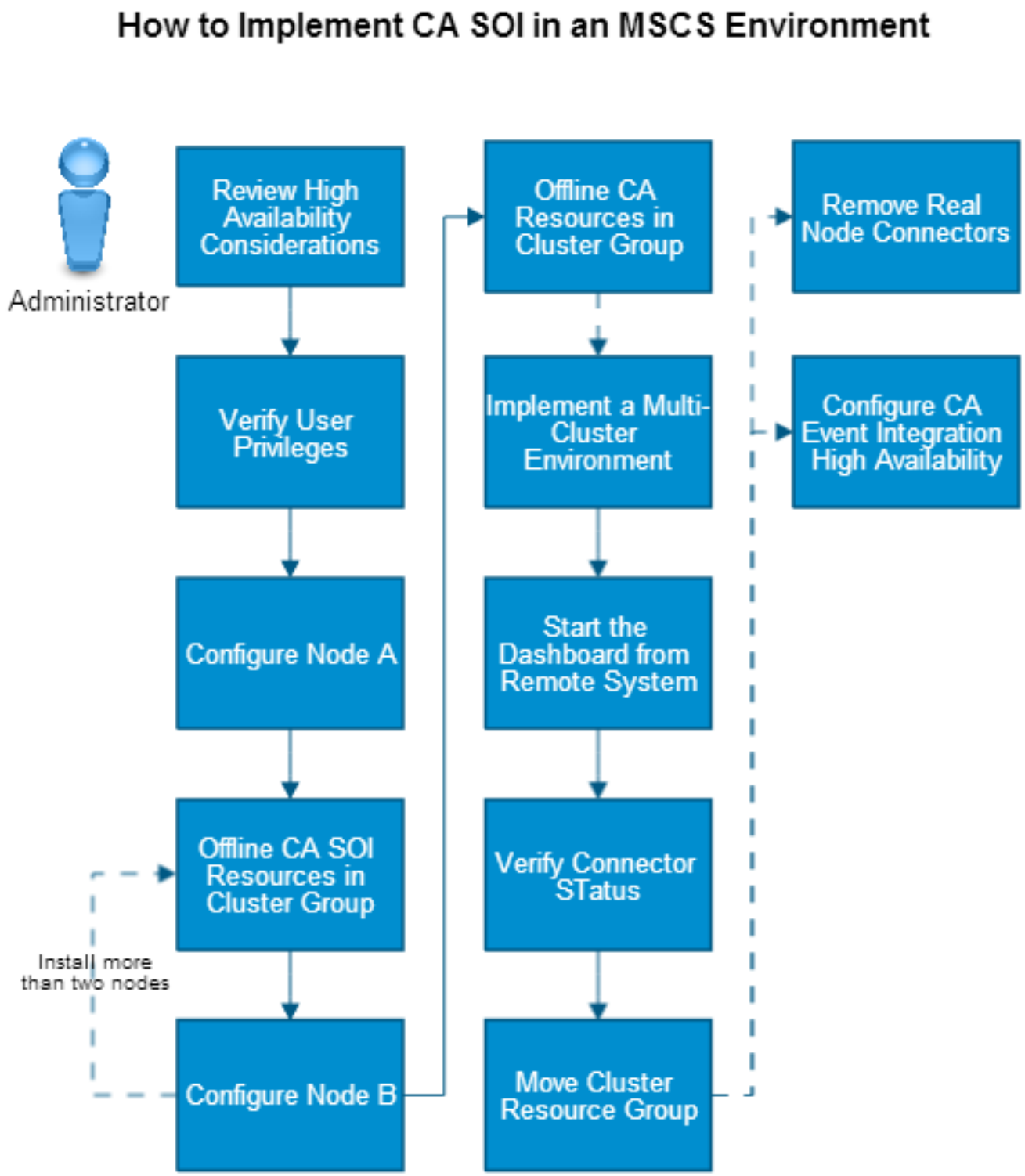
How to Implement CA SOI in an MSCS Environment

As an administrator, you can implement high availability in an MSCS environment.

High availability, which is also known as *fault tolerance* or *failover*, is a common architectural requirement. High availability is based on a Microsoft Cluster Server (MSCS). High availability helps to ensure a business continuity during an IT resource interruption. The main objective of implementing a high availability solution is no downtime for IT resources. High availability support for CA SOI provides failover capabilities and a solution for applying maintenance while avoiding product downtime.

Use this scenario to guide you through the process:

Figure 17: how to implement ca soi in an mscs environment



1. Review the high availability implementation considerations

2. Verify user privileges
3. To enable multi-cluster on MQ Server, Install MQ Server.
4. Configure Node A
5. Offline all CA SOI resources in the cluster group
6. Configure Node B
7. (Optional) Repeat steps 4-5 to install on more than two nodes in the cluster, if necessary.
8. Offline all CA SOI resources in the cluster group
9. If you installed the necessary CA SOI components on the cluster nodes, skip to Step 10. For a multi-cluster deployment, continue with Step 9.
10. (Optional) Implement a multi-cluster deployment
11. Start the Dashboard from a remote system
12. Verify the status of all connectors
13. Move the cluster resource group in each cluster to the node that you want to remain active.
14. (Optional) Remove any real node connectors from the Administration tab if the appropriate virtual node connectors exist.

High Availability Implementation Considerations

Contents

Review the following considerations before starting your high availability implementation.

Platform Support

See Software Support.

Assumptions and Prerequisites

To implement CA SOI in an MSCS environment, you need a working understanding of MSCS. You are also familiar with how MSCS is deployed and maintained in your environment. Basic knowledge of defining cluster resources and customizing CA SOI files is not required, but it can be helpful.

Your environment meets the following requirements:

- A Microsoft SQL Server instance is required. The instance can exist on the cluster that is configured for high availability or outside of the cluster.
- You can move the Windows 2008/2012/2016 cluster "Service or Application" that is used for CA SOI failover between cluster nodes. Have relevant details regarding that group available.
- A shared drive is available and is not a QUORUM disk.
- Each node in the cluster must have at least two network adapters. One adapter is used for the client public network and the second one is used for internal cluster communication.
  If the order of adapters and bindings is wrong, connectors may not work properly. The adapter for public network must be listed first in the Adapters and Bindings list in Windows Network Connections Advanced Settings.

The procedures and examples in this section are based on a cluster (or multiple clusters) consisting of two nodes. However, you can use the procedures to apply CA SOI on clusters with more than two nodes.

Failover Considerations

When a failover occurs, active user interface sessions can be lost for a short time period. The interface indicates that it lost the connection. If you shut down and quickly restart the CA SOI services, you may not notice interruptions in the user interface. However, if you experience an interruption, refresh the browser on the Dashboard to connect again.

When the Dashboard loses the connection and requires a refresh, the Dashboard displays the message:

```
Warning: Service Assurance UI Server is down or the host is currently unavailable.
```

When the Operations Console failover is in progress, the Operations Console displays a red square around the entire console. The status displays the following message:

```
Lost connection to web server.
```

The Operations Console automatically reconnects when the failover completes, but CA SOI can prompt you to resupply valid user credentials.

Failover of ActiveMQ Server

The following table describes the failover of MQ Server on Microsoft Cluster and Non-Cluster environment.

	Microsoft Cluster	Non-Cluster
MQ Server and SA Manager on different machine	Not Supported	Not Supported
MQ Server and SA Manager on the same machine	Supported with full system restart and sync.	Not Supported

NOTE

Installation of MQ Sever on a separate cluster node is not supported.

Service Shutdown

The time to shut down the CA SOI services can delay the failover process. To minimize this issue, the resource kit changes the shut down wait time from 5 minutes to 1 minute. The change takes effect when the services are restarted.

In some cases, the CA SAM Integration Services service can take up to 200 seconds to shut down. The resource kit changes this default setting to 45 seconds. This option specifies the number of seconds to allow between the time the wrapper asks the JVM to shut down and the time that the JVM side of the wrapper responds that it is stopping. To change this setting, open the <SOI_HOME>\jsw\conf\SAM-IntegrationServices.conf file and update the wrapper.jvm_exit.timeout option.

Failover of Remote SA Store Database

If the SA Store database is remote and the connection is lost due to a restart of the remote database server, network problems, or other issues, the lost connection can affect newly created alerts. It may also prevent CA SOI from performing certain user interface tasks that require updates to the SA Store. For example, clearing an alert generates an error message while the connection is lost.

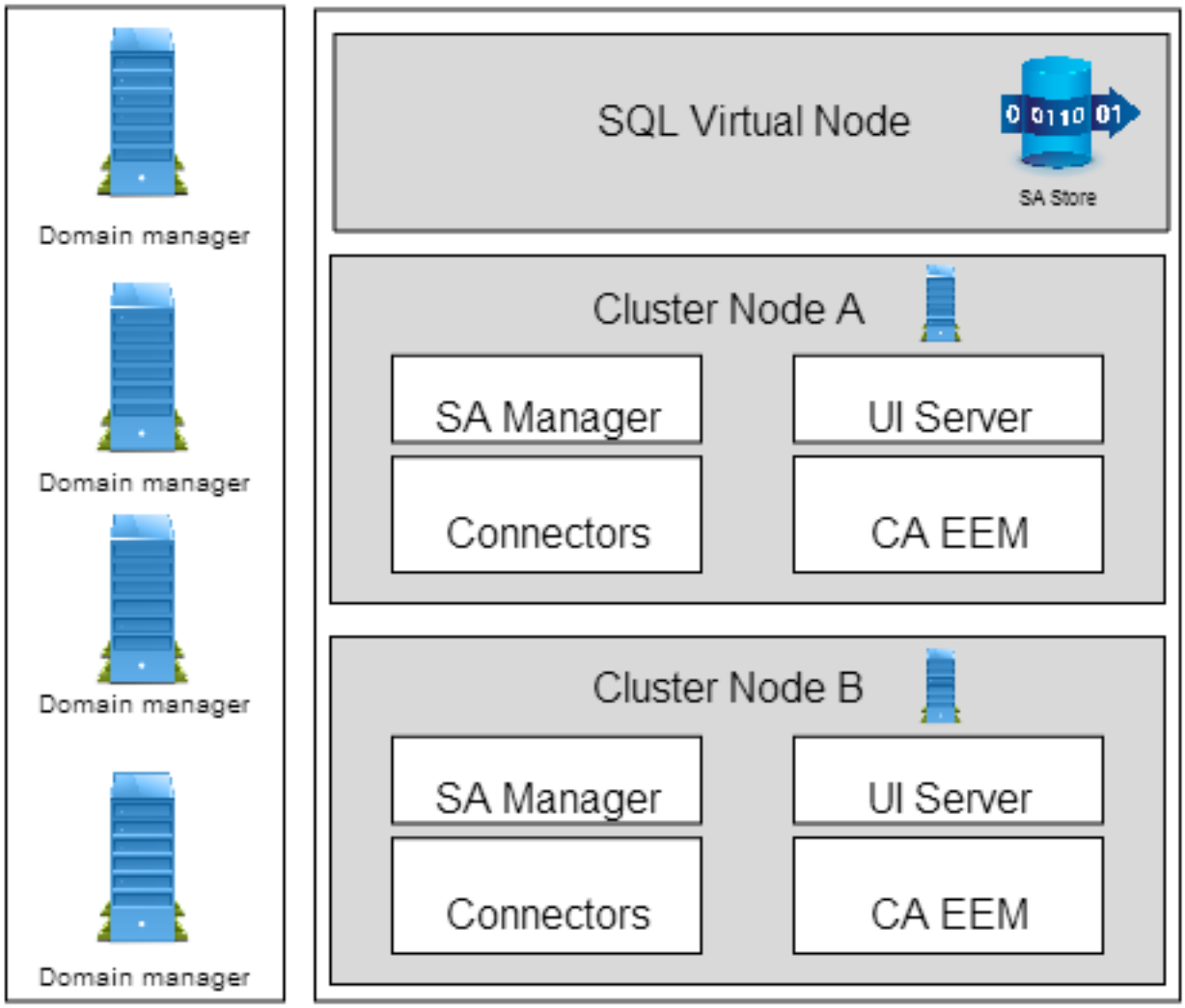
When the database connection is restored, it continues to function, but there is no queuing mechanism. If the IFW restarts, it resynchronizes the alerts from domain managers except for the Universal connector alerts. You can reissue any clear alert commands that generated an error after the remote database server becomes available.

CA SOI High Availability Architecture

You have several architectural options for a high availability implementation. The diagrams in this topic show common configurations.

SA Store database, SA Manager, and UI Server

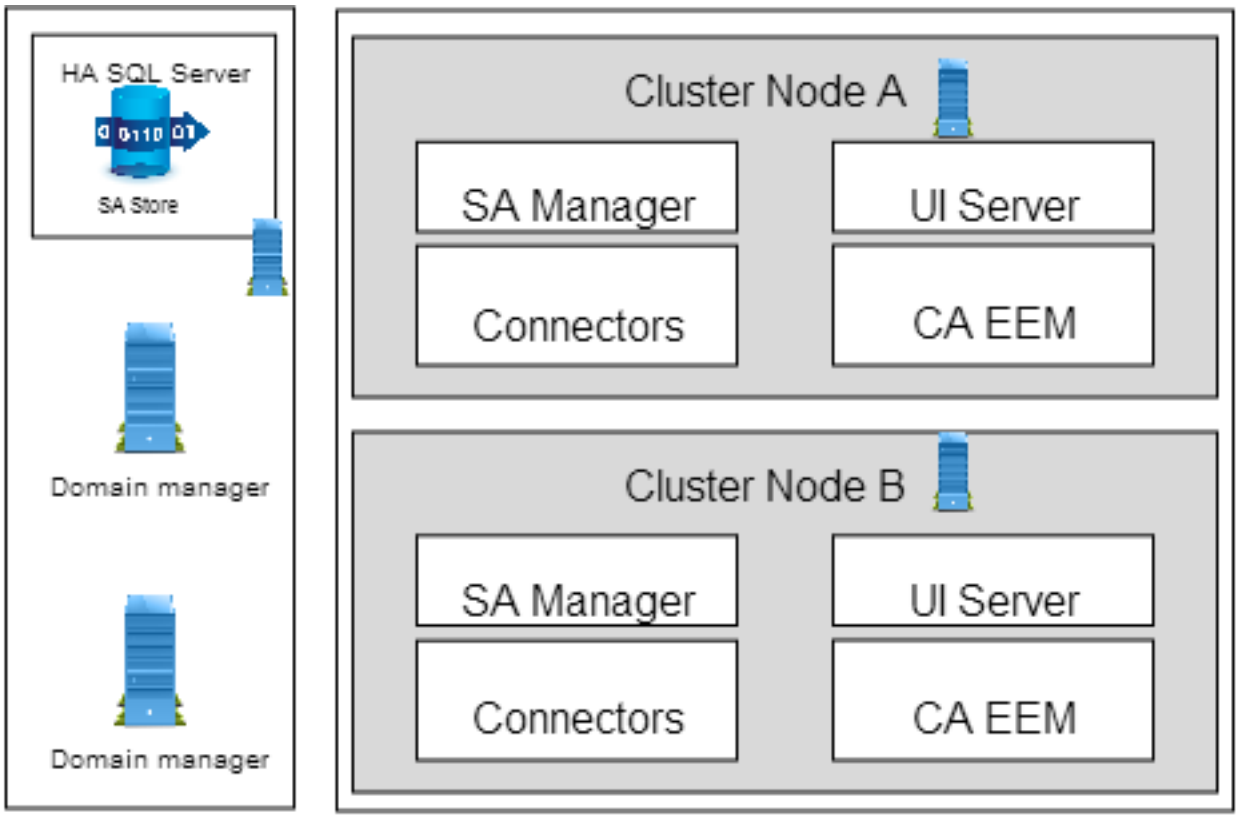
Figure 18: HA Architecture 1



Note that in this configuration, the clustered components use a single virtual Microsoft SQL Server node within the cluster for the SA Store database. All CA SOI components are installed on a single cluster node and a single MQ Server fails over to a secondary single cluster node. The connectors installed on the SA Manager and CA EEM are also replicated across nodes.

SA Manager, UI Server, and remote SA Store database

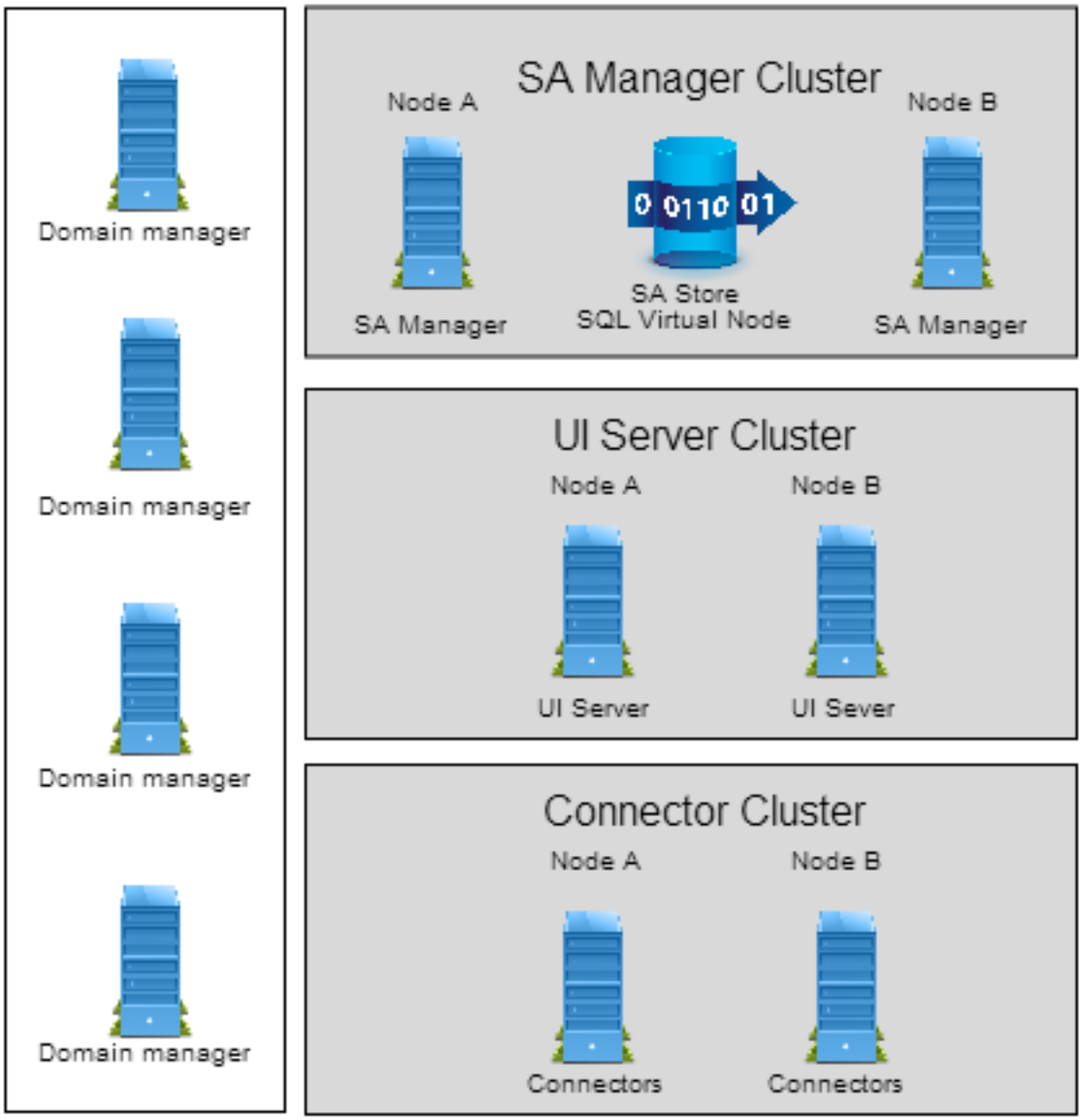
Figure 19: HA architecture 2



In this configuration, the SA Manager and UI Server on both cluster nodes report to a remote highly available database. If the remote database is not available, alerts are not created and the user interface cannot make updates such as clearing alerts. Therefore, we recommend that the SA Store database is local to the SA Manager when possible.

Multi-server HA Deployment

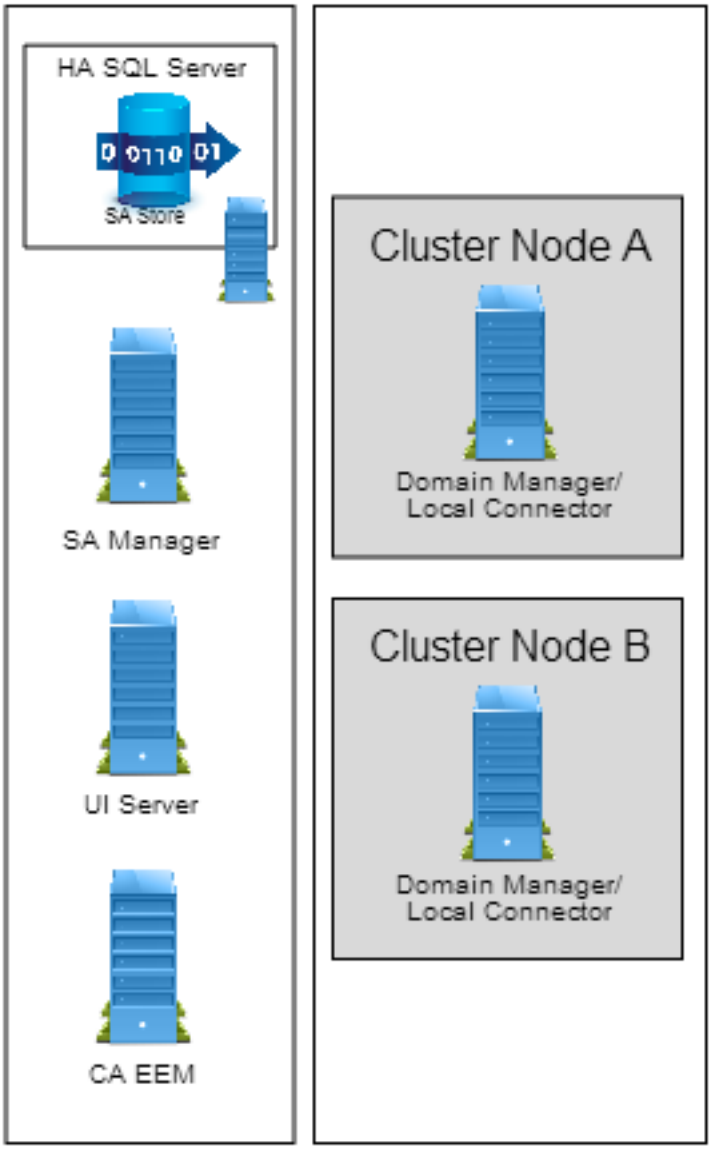
Figure 20: multi-server ha deployment



This configuration represents a typical CA SOI deployment with components installed on multiple servers. To implement a multi-server HA deployment, you must deploy each component on cluster nodes in separate clusters. This example uses a SQL virtual node on the SA Manager cluster for the SA Store database. CA EEM is not pictured but can be installed with the SA Manager or remotely. Depending on the number of connectors and their installation requirements, multiple connector servers on multiple clusters could be necessary.

Domain Manager and Connectors

Figure 21: domain manager and connectors



The domain manager is clustered in this configuration, so the associated connector is required on both cluster nodes. The connectors can report to the highly available SA Manager.

Prerequisite Software Considerations

Other software required in a CA SOI deployment is not a part of the CA SOI high availability solution. The software includes CA EEM, CA Business Intelligence, and the domain managers that are integrated through connectors. For information about the high availability configurations for these products, see the relevant product documentation.

CA EEM Considerations

The SA Manager requires an installation of CA EEM; however, you do not have to install CA EEM on a cluster node. If you do install on a cluster node, install CA EEM on each cluster node independently. This installation is similar to a non-cluster node installation.

For more information, see Configure CA EEM Data Store Replication.

Java Runtime Considerations

If CA EEM is installed on a cluster node, install the Java Runtime there also. As with CA EEM, Java Runtime must be installed on each cluster independently and with the following requirements:

- Use real node names
- Install on local drive
- No sharing of data

CA Business Intelligence Considerations

CA Business Intelligence should be installed in a non-clustered environment. However, if failover is required for CA Business Intelligence, review the BusinessObjects best practice in the CA Business Intelligence documentation.

Domain Manager Considerations

If a domain manager is running in a cluster environment, install the connector for this domain manager in the cluster environment. For more information, see Install Connectors with HA Domain Manager.

Tiered Deployment Considerations

A tiered CA SOI deployment uses CA SOI Domain connectors to send service information from lower-tier (or remote) SA Managers to an Enterprise SA Manager. When you configure a tiered deployment for high availability, each tier requires its own typical high availability implementation. To coordinate communication across tiers when failover occurs, the CA SOI Domain connectors must be able to detect failovers from the enterprise and remote tiers.

For more information, see How to Implement a Tiered CA SOI Deployment in an MSCS Environment.

Cluster Requirements

When you install the SA Manager on a cluster node, some of the SA Manager data files are moved to a shared disk. The files do not require a large amount of space on the disk. The shared disk must meet the following requirements:

- It must be defined in the cluster group selected for CA SOI failover.
- It cannot be the QUORUM disk.
- It must be an MBR partitioned disk. GPT partitioned disks are not supported.

> **NOTE**
> If the cluster has multiple NICs, additional steps may be required.

Make note of the drive letter used for the shared disk and its physical disk cluster resource name, because these values are required for customization.

If SQL Server is installed on a cluster node, you must select the shared disk associated with the SQL cluster resource group. If SQL Server is not installed on a cluster node, identify a cluster resource group that can be used for CA SOI failover. The resource group must have the following cluster resource types defined:

- Network Name
- IP Address
- Physical Disk

If you select a SQL Server resource group, you must select the shared disk used by Microsoft SQL Server on the group.

One cluster is sufficient for a single server CA SOI installation, in which the SA Manager, UI Server, and connectors exists on the same server, or cluster node. However, a typical multi-server installation of CA SOI cannot use multiple nodes within the same cluster to distribute component installations. Instead, you deploy components such as the UI Server and connectors remotely on separate clusters, if necessary. Each cluster must have the same CA SOI components installed on each necessary node (for example, SA Managers on three nodes, or SA Manager and UI Server on two nodes). Therefore, a typical deployment could include three or more clusters, with each component installed on the necessary nodes in each cluster.

> **NOTE**
> From this point forward, a multi-server deployment is also referred to as a multi-cluster deployment.

High Availability Toolkit User Privileges

The High Availability Toolkit interrogates the cluster setup and creates the required CA SOI cluster resources. To do this, you must have the appropriate user privileges to administer the cluster.

The following list is a summary of the required user privileges. The requirements can vary depending on the environments lockdown requirements.

- **Windows 2008/2012/2016 Cluster**
- You must be a member of the Local Administrators group.
- You must have Domain Admin privileges or be a local Administrator user.
  If corporate policy prohibits granting you Domain Admin privileges, you can log in as the local Administrator user. As a local Administrator user, you may not be able to launch Failover Cluster Management because privileges may require you be a Domain user or Domain Admin user. The High Availability Toolkit should allow you to define the required resources as a Domain user or Domain Admin user.
- You may have to run in Elevated User mode, which you Run as Administrator from a command prompt.

In some cases, access can be denied when multiple remote connections exist. In a Windows 2008 cluster, some namespaces are configured with encryption (RequiresEncryption), which requires special permissions to view the content. To accommodate access, the High Availability Toolkit attempts to determine if permission is denied and if denied, prompts to be run in Elevated User mode by selecting Run as Administrator from a command prompt.

Remote Connector Considerations

There are no special considerations for installing remote connectors reporting to a highly available SA Manager. You specify the virtual node name when prompted for the MQ Server host. Similarly, you specify the virtual node name on the Integration Services Configuration page in the Virtual hostname field after you select the Identify connectors with a virtual hostname check box. All other fields are not affected by a highly available manager.

High Availability Connector Considerations

High availability (also known as fault tolerance or failover) is a common architectural requirement based on Microsoft Cluster Server (MSCS) that focuses on ensuring business continuity in the event of an interruption of IT resource availability. The main objective of implementing a high availability solution is zero downtime for IT resources.

You can install connectors in a high availability environment. High availability support provides failover capabilities and a solution for applying maintenance while avoiding service management downtime. For more information about installing connectors to work in a high availability CA SOI environment, see High Availability Implementation.

To install connectors to operate with a high availability domain manager, you must follow the process described in How to Implement CA SOI in an MSCS Environment, with the following differences:

- Install only the IFW and the connector on the domain manager nodes.
- Only connector-specific CA SOI resources are added to the cluster group after you run the resource kit on each node. No data files are moved to the shared disk when running the resource kit on a connector-only system.

  NOTE

  This procedure does not apply for installing the CA Spectrum connector when CA Spectrum is installed with Primary and Secondary SpectroSERVERSs (Distributed SpectroSERVER environment).

Verify the status of all connectors and remove any connectors reporting to the real node.

Verify User Privileges

Verify that you have the privileges to administer the cluster. In most cases, unless further lockdown is required, Domain Admin rights and additional privileges are required to run the High Availability toolkit in Windows 2008, Windows 2012, or Windows 2016.

Follow these steps:

1. Open Failover Cluster Management in the Windows 2008, Windows 2012, or Windows 2016.
   If you are unable to open the program or you do not have the necessary privileges, then contact your system administrator.
2. Add a dummy cluster resource.
   If you are able to add the dummy cluster resource, you *do* have required privileges.

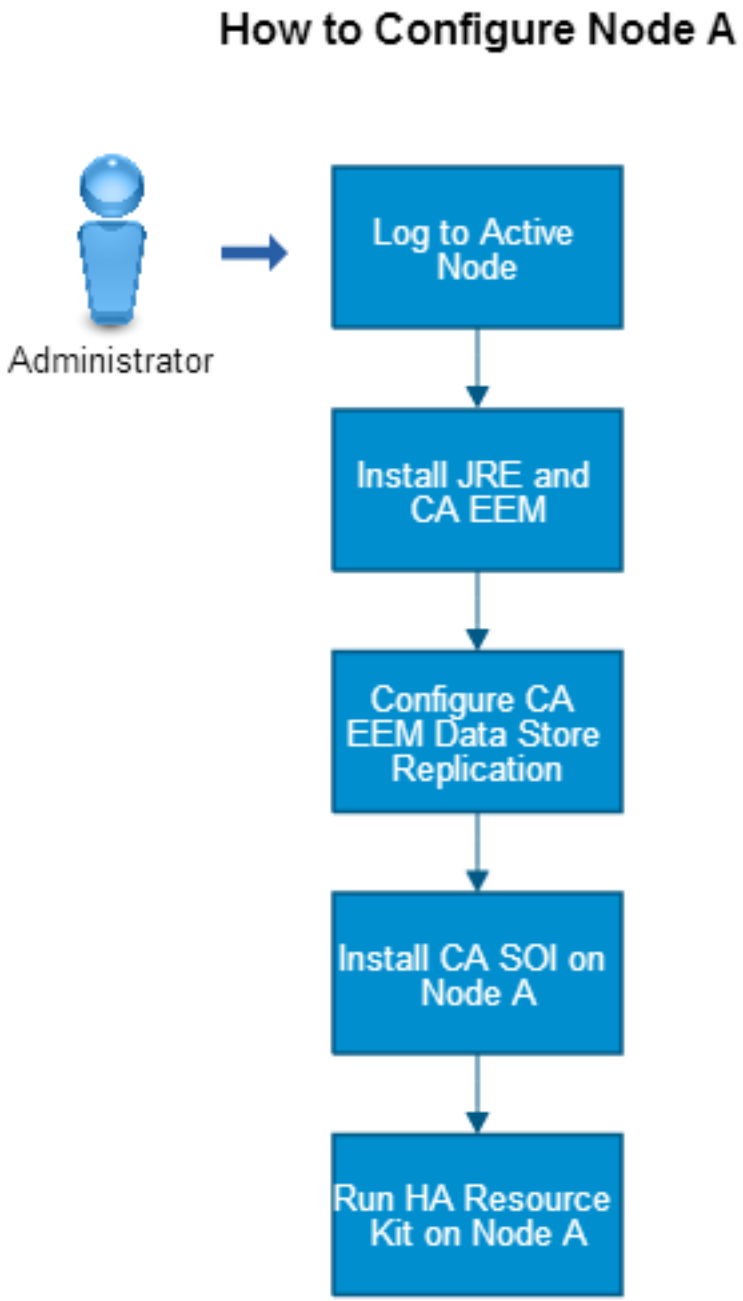
   NOTE
   Delete the dummy cluster resource after adding it successfully.
3. Verify that the domain user is a part of the Local Administrators group.
4. To update user privileges, either by promoting a domain user to a domain admin user or granting special privileges, log out and log in again.

How to Configure Node A

Contents

We refer to the active node in this section as Node A.

Figure 22: how to configure node A



Use the following process to configure Node A.

1. Log on to the active node.
2. (Optional) If required, Install Java Runtime and CA EEM.
3. Configure CA EEM data store replication.
4. Install CA SOI on Node A.

5. Run the High Availability Resource Kit utility on Node A.

Log on to Active Node

Log on to the active node (referred to as Node A in this section) of the selected cluster group.

Install Java Runtime and CA EEM

This is not required if you are using CA EEM installed on a non-cluster node.

If you have not done so already, review the CA EEM considerations and the Java Runtime considerations.

For more information about installing the Java Runtime or CA EEM, refer to their respective product documentation.

If CA EEM is installed, enable CA EEM data store replication.

Configure CA EEM Data Store Replication

The CA EEM release that is shipped with CA SOI uses CA Directory as its data store. This component provides built-in support for data store replication. When CA EEM is installed in a cluster environment, data store replication ensures that customization of the Operations Console such as preferences, users, and group configuration is consistent regardless of the active node. We recommend data store replication to eliminate the need to repeat configuration and preferences on each node.

When CA EEM is installed on two cluster nodes, Node A reports to CA EEM on Node A when active. Similarly, Node B reports to CA EEM on Node B when active. CA EEM has its own data store on each cluster node. Each CA EEM is configured to extract LDAP data individually, because this data is not replicated.

Complete the following process to configure CA EEM data store replication:

> **NOTE**
> The following process only applies if CA EEM is installed on the same cluster.

1. Install CA EEM on each cluster node of the CA SOI cluster resource group. In a multi-cluster CA SOI deployment, install CA EEM on the same cluster as the SA Manager.
2. Install CA SOI on each cluster node and provide the following information about the EEM Server Settings page:
   – Specify the virtual node name in the EEM Server field.
   – Specify SOI-<VIRTUALNODE> in the Application Name field.

   > **NOTE**
   > If you change the format of this field, ensure that the Application Name is the same on all cluster nodes.
3. When you run the HA Resource Kit on each cluster node, a page opens asking to configure CA EEM data store replication if CA EEM is installed on the cluster node. Click Yes on this dialog.
   The Resource Kit customizes the required files for multi-write, which automatically synchronizes the application data in the CA EEM data store. This data includes Operations Console privileges, groups, and preferences, but not external users and groups extracted from external sources such as Active Directory.
   A confirmation page opens.

   > **NOTE**
   > Click Yes to enable and configure CA EEM data store replication if CA EEM is or will be installed on all nodes of the cluster.
4. Click OK.
   The CA Directory services are restarted. Data store replication is enabled after you run the Resource Kit on all cluster nodes with CA EEM installed.

Configure CA EEM on each cluster node regardless of data store replication. If CA EEM is configured to use an external directory, configure each node, because external directory data is not synchronized.

The communication on cluster nodes is already trusted, so CA EEM data store replication does not use certificates in a cluster environment.

Install CA SOI on Node A

Install CA SOI on the first active node (Node A) before running the resource kit or installing on other cluster nodes. Specific values are required to prepare for high availability implementation.

Single and multi-cluster deployments are supported. Multi-cluster deployments contain separate component installation distributed across clusters.

Consider the following items:

- In a multi-cluster deployment, you complete the high availability setup, including running the High Availability Toolkit and configuring the cluster resources, before installing the next component and configuring high availability for the component in a separate cluster.
- Clear the Start Services option in the installer to prevent the services from starting. You can install all CA SOI components on this node (including connectors), or only the SA Manager in a multi-cluster installation.

Follow these steps:

1. Run soi-installer.exe from the Disk1\SOI folder of the CA SOI installation image.
   The License Agreement page opens.
2. Accept the license agreement, then accept the third-party license agreements, and click Next.
   The Choose Install Folder screen opens.

   NOTE
   If CA SOI components are already installed on the system, the Choose Install Folder page does not appear. Any new CA SOI components that you install are installed to the existing CA SOI directory because all components must reside in the same directory.
3. Accept, enter, or choose the installation folder.

   NOTE
   The maximum installation path length is 150 characters. The installation blocks paths with more than 150 characters.

   The Choose Install Set screen opens.
4. Proceed through the installation pages. Do the following to prepare for high availability implementation:
   - Select Manager and MQ Server for a combined installation, or MQ Server only for a multi-cluster deployment.
   - The administrator user name and password on the Administrator page must be the same on all cluster nodes.
   - The port numbers on the Manager Configuration page must be the same on all cluster nodes.
   - The UCF Broker port number on the CA Catalyst Logic Server must be the same all cluster nodes.
   - If the SQL Server is installed on the cluster node, ensure you specify the virtual node name for the JDBC Host on the SAMStore DB Configuration page. Using the real node name, in this case, does not work.
   - If the User Interface Server is selected for installation, the port numbers on the User Interface Server Configuration page must be the same on all cluster nodes.
   - When installing the UI Server separately from the SA Manager in a multi-cluster deployment, connect to the SA Manager during installation using its virtual node name.
   - If CA EEM is installed on the cluster node, ensure that you enter the virtual node name in the EEM Server Settings page for the EEM Server and Application Name fields. For more information about virtual nodes, see How to Configure CA EEM Data Store Replication.
   - The TCP port number on the MQ Server Configuration page must be the same on all cluster nodes.
   - The installer automatically selects the real node name for the connector name on the Integration Services Configuration page. Update the Connector Name to the virtual node name.

NOTE
The cluster may have multiple virtual nodes. You must specify the virtual node that is associated with the shared disk that you intend to use. For more information about clusters on multiple virtual nodes, see Cluster Requirements.

- Remember to clear Start Services on the Start Services page.
- You can also install connectors on the same node, or on a separate node in a multi-server deployment. During each connector installation, update the connector name to use the virtual node name. See Step 5 for multi-server deployment requirements.

5. Click Finish on the Summary page.
   The installation begins and the Install Complete page opens when finished. CA SOI is installed on the active node.
6. (Optional) Install connectors on the same node using the standalone connector installers if you want to deploy all components in a single cluster. Follow the same conventions (where applicable) listed in Step 5. Ensure that you change the connector name to use the virtual node name.
   Follow the applicable conventions in Step 5 when you install the UI Server or connectors in separate clusters in a multi-cluster deployment after completing the high availability setup in previous clusters.
7. Start and stop the CA SAM Application Server service.
   Required configuration files are created.

Run the High Availability Resource Kit on Node A

The High Availability Resource Kit defines CA SOI cluster resources and customizes required files. Run the kit on the active node to configure CA SOI for an MSCS environment. This utility also enables CA EEM data store replication.

Follow these steps:

1. Copy the \Disk1\SSAHA folder of the installation image on Node A. Run the kit from the shared drive to use for CA SOI shared data files.
2. Open the SSAHA folder on the shared drive and run setupMscs.hta.
3. Click CA Service Operations Insight HA Setup.
   The HA Setup dialog opens with introductory information.
4. Click Yes.
   A dialog appears for selecting the shared drive.

   NOTE

   If you are not running the utility on a cluster node, a failure dialog appears, and the utility closes.
5. Select the drive letter for the shared disk on which the CA SOI data files will reside and click OK.

   NOTE

   The shared disks that are online on the cluster node are listed. The local disks are not included.

   If SQL Server is installed on the same cluster, select the shared disk that is associated with the SQL Server cluster resource group.

   NOTE

   The drive letter that you select is where the CA SOI resource kit is installed. Otherwise, the resource kit does not let you continue.
6. Click Yes when asked to enable CA EEM data store replication if CA EEM is installed on the cluster node.
   A dialog appears stating the data store replication is enabled.
7. Click OK.
   The CA EEM directory services are restarted, and the resource kit completes the high availability setup.
8. Open the Cluster Administrator and select the cluster group on which you ran the kit.
   Add the following resources to the cluster group:

- CA SOI MQ Server
- CA SOI Application Server
- CA SOI Event Management
- CA SOI Integration Services
- CA SOI User Interface Server
- CA SOI Store Indexer
- CA SOI UCF Broker

NOTE

Some of the resources may not exist if the relevant CA SOI component is not installed.

9. Offline the CA SOI cluster resources.
   Some CA SOI resources can take up to 5 minutes to shut down. The installation sets the service shutdown timeout value to 5 minutes, but the HA Resource Kit changes this value to one minute. The change takes effect after the services are restarted.
   The high availability resource kit has run on Node A.
10. Run a Move Group operation to the second cluster node (Node B).

    NOTE

    The Move Group operation fails if the resources are online.

    Node B is now the active node.
    In a multi-cluster deployment, repeat this procedure on each cluster after completing the full high availability setup on previous clusters, according to the defined sequence.

Offline all CA SOI Resources in Cluster Group

Perform this task before configuring the next node; otherwise, the installation attempts to start the resources during the move group and fails. The failure is due to components not being installed on the new active node yet.

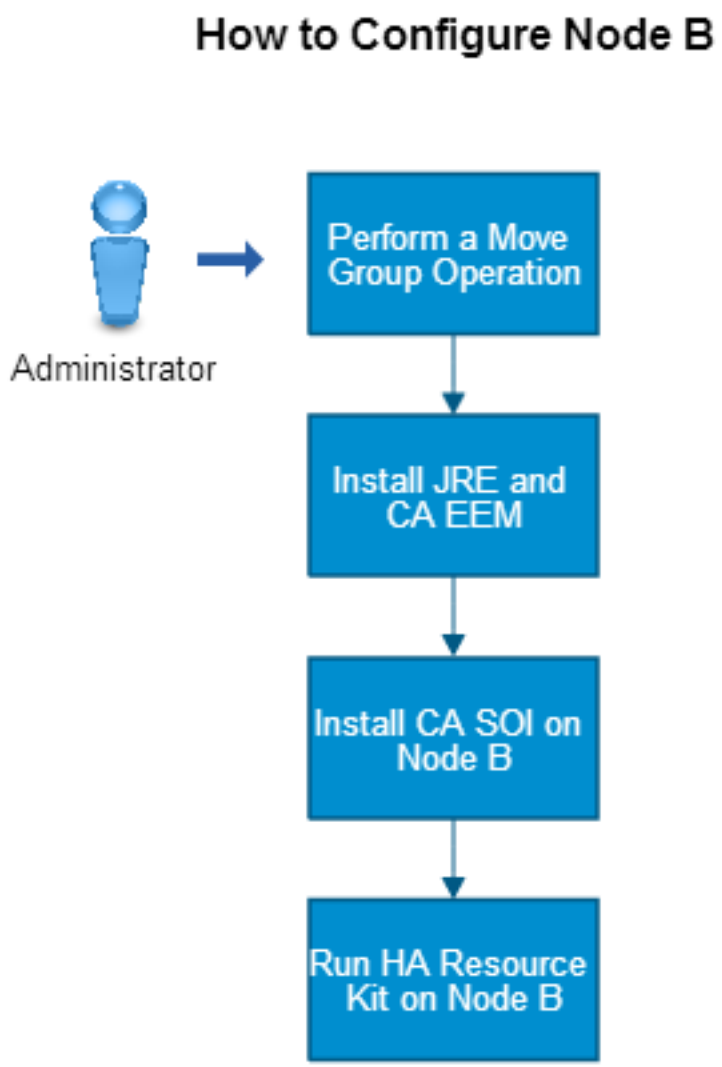
Offline all CA SOI resources in the cluster group. If any CA SOI services are running before running the resource kit to apply customized files, and bring them back online.

How to Configure Node B

Contents

We refer to the active node in this section as Node B.

Figure 23: how to configure node B



1. Perform a move group operation.
2. (Optional) If required, install the Java Runtime and CA EEM on Node B.
3. Install CA SOI on Node B.
4. Run the High Availability Resource Kit on Node B.

Perform a Move Group Operation

Perform a move group operation to the second cluster node, which becomes the active node that we refer to as Node B.

Install Java Runtime and CA EEM

This is not required if you are using CA EEM installed on a non-cluster node.

If you have not done so already, review the CA EEM considerations and the Java Runtime considerations.

For more information about installing the Java Runtime or CA EEM, refer to their respective product documentation.

If CA EEM is installed, enable CA EEM data store replication.

Install CA SOI on Node B

After the Move Group operation, Node B becomes the active cluster node. Install CA SOI on this node to configure a high availability between nodes. Select the same components and use the same options that you selected for Node A. Select the Use Existing Database option because the installation on Node A already created the database.

For multi-cluster deployments, install the same components on Node B that you installed on Node A. For example, if you installed only the SA Manager on Node A, install only the SA Manager on Node B. This convention applies to all clusters.

Follow these steps:

1. Start the CA SOI installation on Node B and follow the same conventions as the installation on Node A, but clear the Start Services option in the installer to prevent the services from starting.
   Several values must be consistent across nodes. For more information, see Install CA SOI on the Active Node.
   When the Database page opens, it detects the created SA Store database on Node A.
2. Select Use Existing Database to use the same database as Node A and click Next.
3. Click Finish on the Summary screen.
   The installation begins and the Install Complete screen appears when finished.
   Follow the same conventions when you install the UI Server or connectors in separate clusters in a multi-cluster deployment as a part of the high availability setup in each cluster.
4. Start and stop the CA SAM Application Server service on Node B after installation.
   Required configuration files are created.

   NOTE

   The log file contains the following message: Could not drop object 'ca_ssa_ci_detail' because it is referenced by a FOREIGN KEY constraint.This message appears as expected. So, the working of CA SOI is not affected.

Run the High Availability Resource Kit on Node B

Run the High Availability Resource Kit on Node B. The kit is installed on the shared drive from Node A and is available for use on Node B.

Follow these steps:

1. Run the Resource Kit following the same conventions in Run the High Availability Resource Kit on Node A.
   The Resource Kit automatically uses the shared disk drive that you provided for Node A. If it requests this information, select the same drive you did for Node A.
   The following resources display on the Node B cluster group after the kit completes:
   – CA SOI Application Server
   – CA SOI MQ Server
   – CA SOI Event Management
   – CA SOI Integration Services
   – CA SOI User Interface Server
   – CA SOI Store Indexer
   – CA SOI UCF Broker

   NOTE
   Some of the resources may not exist if the relevant CA SOI component is not installed.

   The status of the cluster resources that are defined by Node A change to online.
2. Offline and online the SOI Service in the Failover Cluster Manager.

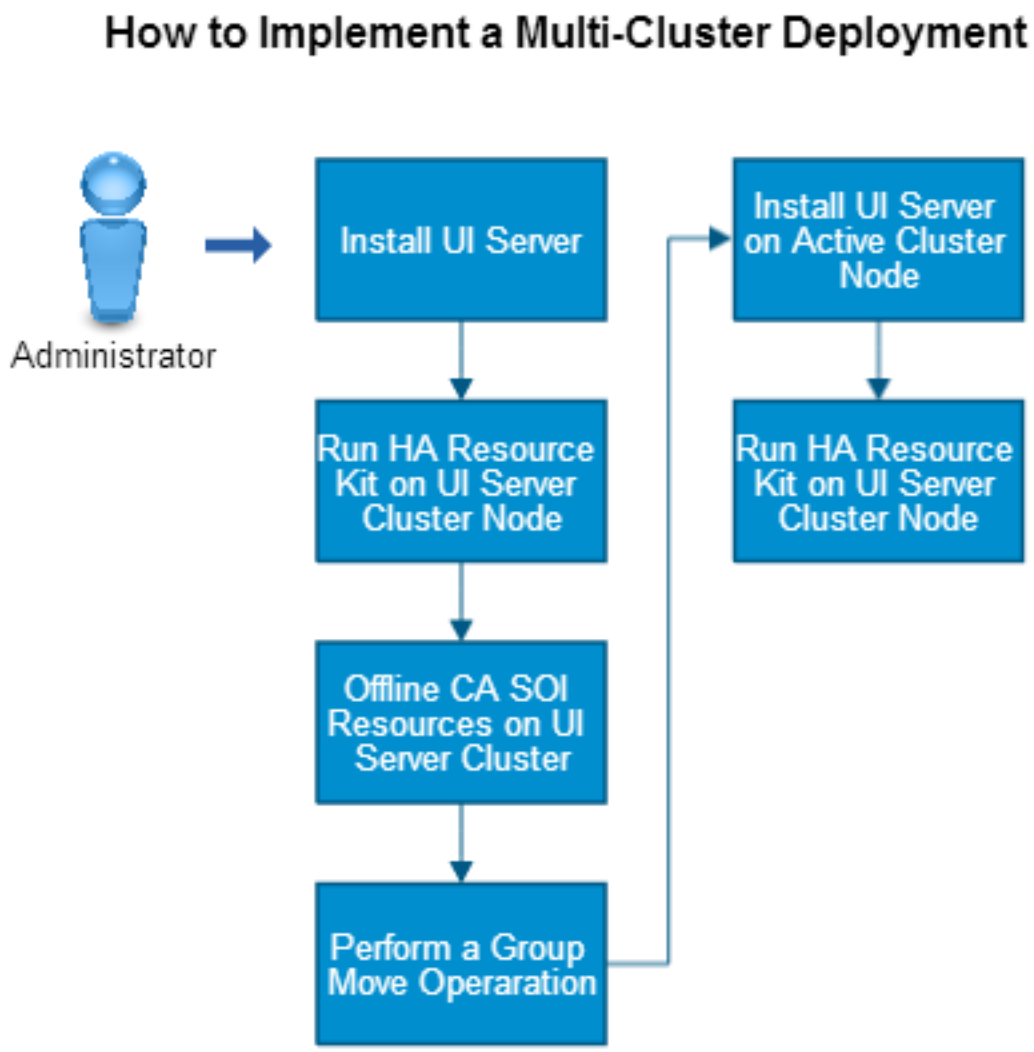
The changes that the Resource Kit made are in effect. Cluster configuration is complete for a single cluster deployment. To verify the deployment, access the dashboard and check connector status.

NOTE
In a multi-cluster deployment, repeat this procedure on each cluster after completing the full high availability setup on previous clusters, according to the defined sequence.

How to Implement Multi-Cluster Deployment

Contents

Figure 24: how to implement a multi-cluster deployment



1. Install the UI Server.
2. Run the High Availability Resource Kit on the active UI Server cluster node.
3. Offline all CA SOI resources in the UI Server cluster group.

4. Perform a move group operation to the second UI Server cluster node.
5. Install the UI Server on the cluster node that is now active.
6. Run the High Availability Resource Kit on the active UI Server cluster node.
7. (Optional) Repeat Steps 3-6 to install the UI Server on more than two nodes in the cluster, if necessary.
8. Repeat Steps 1-7 to install connectors and run the High Availability Resource Kit on the connector cluster nodes. Install the connectors in a different cluster from the SA Manager and UI Server.

Install the UI Server

Install the UI Server on the active node of a different cluster from the one used for the SA Manager if you did not install the UI Server with the SA Manager in Step 3. Clear the Start Services option in the installer.

> **NOTE**
> If installing CA SOI components separately, install each component in a separate cluster, not on separate nodes in the same cluster. Each cluster must have the same CA SOI components that are installed on each necessary node.

Connect to the SA Manager during installation using its virtual node name.

Run the High Availability Resource Kit on Active UI Server Cluster Node

The High Availability Resource Kit defines CA SOI cluster resources and customizes required files. Run the kit on the active node to configure CA SOI for an MSCS environment. This utility also enables CA EEM data store replication.

Follow these steps:

1. Copy the \Disk1\SSAHA folder of the installation image or the r3.3 CU1 patch to the shared disk on Node A. Run the kit from the shared drive to use for CA SOI shared data files.
2. Open the SSAHA folder on the shared drive and run setupMscs.hta.
3. Click CA Service Operations Insight HA Setup.
   The HA Setup dialog opens with introductory information.
4. Click Yes.
   A dialog appears for selecting the shared drive.

   > **NOTE**
   > If you are not running the utility on a cluster node, a failure dialog appears, and the utility closes.
5. Select the drive letter for the shared disk on which the CA SOI data files will reside and click OK.

   > **NOTE**
   > The shared disks that are online on the cluster node are listed. The local disks are not included.

   If SQL Server is installed on the same cluster, select the shared disk that is associated with the SQL Server cluster resource group.

   > **NOTE**
   > The drive letter that you select is where the CA SOI resource kit is installed. Otherwise, the resource kit does not let you continue.
6. Click Yes when asked to enable CA EEM data store replication if CA EEM is installed on the cluster node.
   A dialog appears stating the data store replication is enabled.
7. Click OK.
   The CA EEM directory services are restarted, and the resource kit completes the high availability setup.
8. Open the Cluster Administrator and select the cluster group on which you ran the kit.
   Add the following resources to the cluster group:

- CA SOI Application Server
- CA SOI MQ Server
- CA SOI Event Management
- CA SOI Integration Services
- CA SOI User Interface Server
- CA SOI Store Indexer
- CA SOI UCF Broker

NOTE
Some of the resources may not exist if the relevant CA SOI component is not installed.

9. Offline the CA SOI cluster resources.
   Some CA SOI resources can take up to 5 minutes to shut down. The installation sets the service shutdown timeout value to 5 minutes, but the HA Resource Kit changes this value to one minute. The change takes effect after the services are restarted.
   The high availability resource kit has run on Node A.
10. Run a Move Group operation to the second cluster node (Node B).

    NOTE
    The Move Group operation fails if the resources are online.

    Node B is now the active node.
    In a multi-cluster deployment, repeat this procedure on each cluster after completing the full high availability setup on previous clusters, according to the defined sequence.

Offline all CA SOI Resources in UI Server Cluster Group

Perform this task before configuring the next node; otherwise, the installation attempts to start the resources during the move group and fails. The failure is due to components not being installed on the new active node yet.

Perform a Move Group Operation

Perform a move group operation to the second UI Server cluster node, which becomes the active node.

Install the UI Server on Active Node

You install the UI Server on the cluster node that is now active. Clear the Start Services option in the installer, and connect to the SA Manager during installation using its virtual node name.

After the installation completes, bring the CA SOI resources online in the UI Server cluster.

Install Connectors

Multiple connectors can reside on the same cluster nodes in a cluster, but you cannot install different connectors on separate nodes in the same cluster (for example, a CA eHealth connector on one node and a CA Spectrum connector on a separate node). Repeat the process as many times as necessary in separate clusters if you want to install connectors separately from one another.

While installing each connector, change the connector name in the Integration Services Configuration page to the virtual node name.

Access the Dashboard in a High Availability Environment

Contents

If you configured the UI Server for HA, start the Dashboard from a remote or local server using the virtual node name in the URL.

To start the Dashboard, enter the following URL:

```
http://<virtualnode>:7070/sam/ui
```

- *virtualnode*
  To identify the virtual node name, select the CA SOI cluster resource group in the Cluster Administrator and review the Network Name resource type properties. If SQL Server is installed on the same cluster node, the virtual node name should be the SQL virtual node name.

User Interface Considerations

Any changes that you make to the following settings on the SA Manager are automatically saved to the shared disk. The shared disk is at the following location and occurs on all cluster nodes:

- EEM Configuration (eem-config.xml at <SOI_HOME>\SamUI\webapps\sam)
- Email Configuration (email-config.xml at <SOI_HOME>\tomcat\custom)
- Help Desk Configuration (svcdesk-config.xml at <SOI_HOME>\tomcat\custom)
- CA Process Automation Server Configuration (itpam-config.xml at <SOI_HOME>\tomcat\custom)
- USM Web View Server Configuration (usm_web_view_cfg.xml at <SOI_HOME>\tomcat\custom)
- Mobile Dashboard Server Configuration (mobile_ui_srvr_cfg.xml at <SOI_HOME>\tomcat\custom)
- Escalation action retry setting (action_retry_cfg.xml at <SOI_HOME>\tomcat\custom)

Any changes that you make to the following settings on the UI Server are automatically saved to the shared disk. The shared disk is at the list location and occurs on all cluster nodes:

- EEM Configuration (eem-config.xml at <SOI_HOME>\SamUI\webapps\sam)
- Email Configuration (email-config.xml at <SOI_HOME>\SamUI\custom)
- JNLP Configuration (custom-jnlp-config.xml at <SOI_HOME>\SamUI\custom)

Connector Configuration Files

The High Availability Toolkit automatically updates and then copies the connector configuration files in the <SOI_HOME>\resources directory to the shared disk.

If any manual changes are required to these files, perform them on the shared disk.

Any changes made to connector configuration from the Administration UI automatically update the configuration files on the shared disk.

USM Web View High Availability

The High Availability Toolkit automatically copies the following USM Web View resources to the shared disk:

- Apache SOLR: <SOI_HOME>\SamUI\solr
- Indexer: <SOI_HOME>\Indexer\timestamp.properties
- USM Web View interface: <SOI_HOME>\jsw\bin\bookmarks.xml

Any changes that you make to properties stored in these resources are copied to the shared disk.

For more information about customizing the USM Web View using these shared resources, see Configure USM Web View Integration.

View Connector Status in a High Availability Environment

To verify that connectors operating on the virtual node are online and operating as expected, view the connector status. Remove any connectors that are registered to the real node name, so that only cluster-aware connectors exist.

Connectors appear associated with the virtual nodes. If you did not specify the virtual node during installation, connectors can appear associated with the real nodes. These connectors appear as offline.

Follow these steps:

1. Access the Dashboard in a High Availability Environment and click the Administration tab.
2. Verify the local and remote status of all connectors.
   If you did not specify for connectors to use the virtual node name or specified the wrong virtual node name during installation, you may see two versions of local connectors: one reporting to the virtual node name, and one reporting to the real node name or incorrect virtual node name. All connectors reporting to the virtual node name should be Online. Connectors reporting to the real node name were installed without the virtual node name specification and should appear as Offline. Remove any connectors that are registered with the real node name or with an incorrect virtual node name.
3. (Optional) Remove any connector that is registered with the real node name or incorrect virtual node name.
   All local connectors reporting to the real node name or incorrect virtual node name are removed. All connector data should come from cluster-aware connectors reporting to the virtual node.

How to Implement a Tiered CA SOI Deployment in a MSCS Environment

As an administrator, you can configure a tiered CA SOI deployment. The deployment uses CA SOI Domain connectors to send service information from lower-tier (or remote) SA Managers to an Enterprise SA Manager. You can configure a tiered CA SOI deployment for high availability, so that not only does each individual CA SOI instance fail over appropriately, but the CA SOI Domain connectors can also detect a failover in their remote SA Managers and switch to collecting data from the secondary node when required.

Each CA SOI instance requires its own cluster, and distributing components across systems requires multiple clusters per instance. Each cluster must have the same CA SOI components installed on each necessary node.

Complete the following process to implement a tiered CA SOI deployment in a MSCS environment:

1. Install all remote CA Catalyst and CA SOI instances and configure them for high availability according to the instructions in How to Implement CA SOI in a MSCS Environment.
2. Install the Enterprise CA SOI tier, including CA SOI Domain connectors for each remote CA SOI tier, on nodes in a separate cluster from the other tiers and configure the CA Catalyst and CA SOI enterprise tier for high availability according to the instructions in How to Implement CA SOI in a MSCS Environment.
   Adhere to the following conventions when installing CA SOI Domain connectors as part of the enterprise tier:
   – For detailed installation instructions, see Install the CA SOI Domain Connector.
   – Install each CA SOI Domain connector on the same cluster node as other enterprise tier components or on a separate cluster. Do not install CA SOI Domain connectors on the same cluster node as any remote CA SOI tier.
   – Configure the CA SOI Domain connector to connect to the virtual node of the CA Catalyst Server for the Enterprise SA Manager on the Enterprise Message Bus installer page if the CA Catalyst Server is configured for high availability.
   – Configure the CA SOI Domain connector to connect to the virtual node of the appropriate remote SA Manager on the Domain Connector Configuration installer page.

   The tiered CA SOI deployed is configured for high availability. Each tier can fail over as with a typical high availability implementation, and if a remote tier fails over, the CA SOI Domain connector can collect information from its secondary node for transmittal to the Enterprise SA Manager.

Install Connectors with HA Domain Manager

As an administrator, to install connectors to operate with a highly available domain manager, follow the process described in How to Implement CA SOI in a MSCS Environment, with the following differences:

- Install only the connector on the domain manager nodes.
- Only connector-specific CA SOI resources are added to the cluster group after you run the resource kit on each node.

> **NOTE**
> This procedure does not apply for installing the CA Spectrum connector when CA Spectrum is installed with Primary and Secondary SpectroSERVERs (Distributed SpectroSERVER environment).

Verify the status of all connectors and remove any connectors reporting to the real node or incorrect virtual node.

Update High Availability Environment

Contents

To install more connectors or CA SOI components after you run the High Availability Resource Kit, perform the following steps to update the high availability environment:

1. Offline all CA SOI cluster resources.
2. Install the component or connector on all cluster nodes. Either install the component on cluster nodes that contain existing CA SOI components, or install on cluster nodes in a separate cluster if the component requires a dedicated system.
3. Rerun the kit on all cluster nodes with CA SOI components to apply the changes in the high availability environment.

If you previously configured CA EEM data store replication, a message displays when you rerun the kit stating that data store replication was previously enabled and that the updates are not reapplied.

How to Add Connectors to a High Availability Installation

Connectors should already be installed before you run the resource kit. However, if you want to add connectors to a high availability installation after the resource kit is run, complete the following process:

1. Offline all CA SOI cluster resources.
2. Start and complete the connector installation on the first cluster node, and clear Start Services on the Start Services page.
   Install the connector on a node with existing CA SOI components or on a cluster node in a separate cluster if it requires its own dedicated server. If you install on a node remote to the SA Manager, enter the virtual node name when specifying the SA Manager connection information.
3. Verify that the CA SOI cluster resources are still offline.
4. Start and complete the connector installation on the second cluster node, and clear Start Services on the Start Services page.
5. (Optional) Repeat Steps 3-4 to add the connector to other cluster nodes if necessary.
6. Rerun the High Availability Resource Kit on all cluster nodes with CA SOI components.
7. Online the CA SOI cluster resources.
   The connectors are added to the highly available SA Manager.

Background Images

The High Availability Resource Kit updates the location of the topology background images to the shared drive and copies all background images to the shared drive so that the cluster nodes can share the background images.

To copy more background images, copy them to the shared drive and not the local drive. To find the location of the topology background images folder, see the sam-app-config.xml file at <SOI_HOME>\SamUI\webapps\sam\WEB-INF\sam\config and <SOI_HOME>\tomcat\webapps\sam\WEB-INF\sam\config.

How to Update a High Availability Implementation with Multiple NICs

As an administrator, you must manually correct the virtual node name suffix in all necessary places.

MSCS 2008/2012/2016 typically requires two NIC interfaces: one for the internal heart beat and another for the public network. Additional NICs may exist, such as one for a private network. If you have performed a high availability implementation of CA SOI in an environment with additional NICs that have different domain suffixes, the resource kit automatically uses the suffix from the cluster node instead of the virtual node.

> **WARNING**
> This process is only necessary in the typical use case described when multiple NICs cause a situation where the suffix of the cluster nodes differs from that of the virtual node.

Complete the following process to correct the virtual node name suffix in an environment with multiple NICs:

1. Complete all steps in How to Implement CA SOI in an MSCS Environment.
2. Make a backup copy of the <SOI_HOME>\resources directory on the shared disk.
3. Offline the CA SOI cluster resources.
4. Update all XML files in the <SOI_HOME>\resources directory with the virtual node name defined so that it uses the correct suffix. Files that require updates include the following:
   - eventManagerServerConfig.xml
   - Configurations\SSA_IFW_*servername*.xml
   - Configurations\mtc_*servername*.xml
5. Update the following files to update the suffix in the virtual node name:
   - <SOI_HOME>\apache-activemq\conf\activemq.xml
   - <SOI_HOME>\SamUI\webapps\sam\server-config.xml
   - <SOI_HOME>\tomcat\lib\eventManagerClientConfig.xml
   - <SOI_HOME>\ServiceDiscovery\connectivityContext.xml

   > **NOTE**
   > Some of these files may not exist if the component is not installed on the same cluster or if the specific option was not selected as part of the CA SOI installation.
6. Update the following Registry files located at <SOI_HOME>\tomcat\registry\topology\physical\node0\sor to update the suffix in the virtual node name:
   - restserver.xml
   - sorapp.xml
   - ssaserver.xml
   - wsman.properties

   > **NOTE**
   > These files may not exist if the SA Manager was installed on a different cluster.
7. Run <SOI_HOME>\tomcat\registry\registryloader.bat.
   The updated files are loaded into the Registry.
8. Online the CA SOI cluster resources.
9. Remove any duplicate connectors that use the old virtual node name suffix.
10. (Optional) Repeat this procedure if necessary for an implementation that uses multiple clusters.

Upgrade Issue in Microsoft Cluster Server Environment

Symptom:

The **<SOI_HOME>\log\ifw.log*** files are no longer created after upgrading from Service Operations Insight 4.0 SP2 to Service Operations Insight 4.2 version. This issue occurs only on Service Operations Insight 4.2 Microsoft Cluster Server environment.

This issue occurs because the **<SOI_HOME>\resources\log4j.xml** file resides on the local drives of the physical cluster nodes, but not on the shared drive of the cluster.
Solution:

Copy the **<physical_drive>:<SOI_HOME>\resources\log4j.xml** to **<shared_drive>:<SOI_HOME>\resources\log4j.xml** on the Microsoft cluster node.

Restart CA SAM Integration Service.

Configure CA SOI with EEM High Availability

This article describes how to configure CA SOI with EEM HA. Configuring CA SOI with EEM HA allows the EEM user to be logged in to the CA SOI during EEM failover.
Perform the following procedures to configure CA SOI with EEM HA:

Failover Configuration of EEM

CA EEM provides a command-line tool for automating the failover configuration process.
As an Administrator, you can plan a failover setup environment by performing the following steps:

1. Identify the servers that must act as a primary server and secondary servers.
2. Configure the primary server.
3. Add secondary servers to the primary server.
4. Synchronize the secondary servers with the primary server.

Follow these steps:

1. Follow the Prerequisites
2. Set Up a Failover Environment

   NOTE

   Ensure that the same EMM users are added in both Primary and Secondary servers.

Configure CA SOI with EEM HA

After you have set up the failover environment for EMM, you must configure EEM in CA SOI.
Follows these steps:

NOTE

This procedure is applicable for existing SOI user who wants to configure SOI with EEM HA.

1. Navigate to CA SOI Administration Dashboard.
2. Expand **CA Service Operations Insight Manager Configuration**, and click **EEM Configuration**.
3. Provide the following details:
   a. **EEM Server Host:** Specify the primary EEM host and the secondary EEM host separated by comma without spaces..
      For Example: <EEM Server1 Fullqualified Hostname>,<EEM Server2 Fullqualified Hostname>
   b. **User Name:** Specifies the EEM user name.

   c. **Password:** Specifies the EEM password.
   d. **Application Name:** Specifies the hostname where CA SOI is installed.

      NOTE

      The EEM username and password must be the same for both Primary and Secondary servers.
4. Click **Test** to verify the connectivity.
5. Click **Save**.
6. Expand **CA Service Operations Insight UI Server Configuration**, and click **EEM Configuration**.
   a. Follow the **steps (a to g)** as mentioned for the SOI Manager configuration.

The EEM HA is configured on CA SOI Manager and CA SOI UI Server.

NOTE

For fresh SOI installation, provide the EEM Virtual Node Host Name and password in the EEM Configuration page.

Verification of EMM users in CA SOI

After you configured CA SOI with EEM HA, verify EEM users in the Operation Console.
Follow these steps:

1. Open the Operation Console.
2. Right-click a group, and click **Add User**.
3. Select **All users** option and click **Ok**.

The EEM users are added to the User list. Log in to the Operation Console with the user that you have added in SOI and perform EEM failover. You see the user will not be logged out of the Operation Console during EEM failover.

Upgrades and Migrations

This section describes how to upgrade CA SOI components from previous supported releases.

Upgrade CA SOI

Contents

This release supports upgrades from the following previous releases:

- CA SOI r3.3 with CU2 applied
- CA SOI 4.0
- CA SOI r4.0 with CU1 and SP2 applied

Perform an Upgrade

You can upgrade CA SOI by running the installer on any system that has CA SOI components installed. The installer automatically detects which components are installed and upgrades those components only. The upgrade retains all data from previous releases, including the following:

- Service models
- CIs
- Alerts
- Escalation policies
- Configuration settings

Typical Multi-Server Upgrade

When upgrading a typical multi-server installation, you should upgrade CA SOI components in the following order:

1. SA Manager
2. UI Server
3. Standalone IFW or Containers
4. Connectors

UI Server or Universal Connector and Integration Services on the Same Server/SA Manager on Different Server

If you installed the UI Server or the Universal Connector Client on a server with Integration Services and without the SA Manager, then upgrade in the following order:

1. Upgrade the CA SOI components with soi-installer.exe.
2. Run IntegrationServices.exe to upgrade the IFW.

UI Server and/or Universal Connector, Integration Services, and SA Manager on Same Server

If you installed the UI Server or the Universal Connector Client on a server with Integration Services and the SA Manager, then run the soi-installer.exe to upgrade all components on the server.

> **NOTE**
> When you perform an upgrade, you cannot change any settings from the previous installation. You must make any necessary changes after the upgrade. For more information about changing settings such as connector properties and ports, see Installation Maintenance.

Follow these steps:

1. Verify that the database is running.
2. Verify the database schema version:
   a. Open the SQL Server Management Studio and connect to your SAMStore database.
   b. Run the following query:
   ```
   select count (*) from productInfo
   ```
   c. The query should return a result of 1. If the query returns any other result, contact CA Support before upgrading.
3. Delete any large log files from your SOI_HOME directory. Large log files can significantly slow down the upgrade process.
4. Back up the configuration files located in the **\SOI\jsw\conf** folder**.**
5. Run soi-installer.exe from the Disk1\SOI folder of the CA SOI installation image on a system with CA SOI components installed.

   > **TIP**
   >
   > The user who runs the installation must be a Windows Administrator. Additionally, your local security policies may require you to right-click the installer and select Run as Administrator.

   The CA SOI installer opens with the Introduction page displayed.
6. Accept the license agreements.
   The CA Service Operations Insight UPDATE dialog opens.
7. Click OK.
   The LGPL Distribution page opens. You must specify the location of required third-party libraries on this page.

8. Choose the folder where the third-party distribution folder resides (the Disk2\lgpl folder of the installation image) and click Next. The installer stops any running CA SOI services.
9. On the Service Startup page, select whether to start the product services after upgrade completes, and click Next. The Pre-Installation Summary page opens.
10. Verify the parameters indicated and click Install.
    The upgrade begins and is tracked by a progress indicator. When the installation finishes, the Install Complete page opens.
11. Click Done.
    All components installed on the system are upgraded. The CA SOI upgrade also upgrades the embedded CA Catalyst infrastructure, including the USM schema.
12. (Optional) Review the SOI_Install_Upgrade_*releasenumber*.log file in the <SOI_HOME>\log directory to check for installation errors.
13. (Optional) Repeat this procedure on other systems that contain non-connector components, if necessary.
14. (Recommended) After the upgrade, clear the Java cache by deleting Temporary Files through the Java Control Panel.

    NOTE
    The upgraded installation retains the previous installation directory, which could contain references to the old product name.

Upgrade CA Catalyst Connectors

CA Catalyst connectors do not require an upgrade to work with the current release of CA SOI. However, upgrade the IFW on the system to ensure that they can interact with the latest version of the product. Upgrading the IFW is not required in the following situations:

- The connectors exist on the SA Manager. Upgrading the SA Manager to the current release of CA SOI already upgraded the IFW, and the connectors come online and work immediately after the upgrade.
- Connectors provided with the CA SOI image may require a full upgrade.

To upgrade CA Catalyst connectors, install the IFW on the connector system.

The IFW upgrades, and the connectors are configured to work with the current release of CA SOI.

Consider the following information:

- References to the old product name and version number are not updated in the connector documentation and Start menu shortcuts.
- If you are using CA Service Desk with the auto clear alerts option enabled, then shut down a connector, CA SOI automatically changes the cleared alerts to the selected status. You cannot undo this operation. For more information, see How to Work with Configured Help Desk Integrations.
- If the CA Catalyst container with the IFW proxy installed is upgraded from CA Catalyst 3.2 to 3.3 or 3.4, then the IFW proxy must also be upgraded to CA SOI 3.2.

Upgrade Connectors Provided on the CA SOI Image

To upgrade connectors provided on the CA SOI image, run the appropriate connector installation program from Disk1\SOI and proceed through the installations using the instructions at the provided links:

- **Universal Connector**
  If the Universal Connector Client is installed at a previous release, the installer prompts if you want to upgrade. During the upgrade, new Universal Connector client XML files are installed. However, any existing Universal connector XML files will still be available.
- **Domain Connector**
  If the installer detects a previous version of the Domain connector, it will prompt for an upgrade. If the local SA Manager has not been upgraded yet, the Domain connector installer will inform you that you must do so before

proceeding. If the Manager and Domain connector are at the current version, the domain connector installer will allow for configuring an additional domain connector instance.

- **Sample Connector**
  You do not have to upgrade the Sample connector.

Migration Requirements

Contents

This section provides the manual migration steps to recreate your previous CA SOI environment.

Help Desk Integrations

Consider the following migration requirements for reestablishing help desk integrations.

Auto Clear Alert Settings

During upgrade, CA SOI removes any auto clear alert settings that you configured in the Help Desk Configuration dialog. You must configure the auto clear settings again.

Migrate Help Desk Integrations to the Latest CA Process Automation Workflow

The following help desk integrations use CA Process Automation workflows:

- BMC Remedy
- HP Service Manager
- ServiceNow

The CA Process Automation workflows for these integrations ship with CA SOI. If the latest release of CA SOI includes an updated workflow for your help desk integration, perform a manual migration to take advantage of the updated workflow.

Follow these steps:

1. Save your current workflow configuration and any customizations you have made.
2. Apply the new workflow according to the instructions for the help desk integration.
3. Re-apply your specific customization and configuration changes to the new workflow.

Reestablish ServiceNow Connection

During the upgrade, the files that you configured to enable CA Process Automation to connect to ServiceNow are overwritten. To reestablish the ServiceNow connection, update these files as described in the *Integration Guide* packaged with the ServiceNow connector.

HP Service Manager and ServiceNow Integration Post-Upgrade Configuration

After upgrading, the Help Desk Configuration page on the Administration tab incorrectly resets to CA Service Desk for HP Service Manager and ServiceNow integrations. You must set these integrations again.

> **NOTE**
> Perform this procedure only if you are using either an HP Service Manager or ServiceNow help desk integration.

Follow these steps:

1. On the Dashboard, click the Administration tab.
2. Expand CA Service Operations Insight Manager Configuration and the server name.
3. Click Help Desk Configuration.

4. Select either HP Service Manager or ServiceNow from the Help Desk Type drop-down list.
5. Click Save.
6. (Optional) Click Test to verify the connection.
7. Restart the CA SAM Application Server service and the CA SAM User Interface Server service.

How to Upgrade from a Previous High Availability Installation

As an administrator, complete the following process to upgrade a high availability deployment:

1. Rename the SSAHA directory on the shared drive to SSAHA_33.
2. Copy the SSAHA directory from CA SOI release media to the shared drive. The directory name on the shared drive must be SSAHA.
3. Ensure that CA SAM cluster resources are offline.
4. Copy the contents of the SOI_HOME\resources directory from the shared drive to the local drive. Replace new files on the local drive only.
5. Upgrade all components on the active cluster node.
6. Run the High Availability Resource Kit on the upgraded node.
   If CA EEM is installed on the cluster nodes, the resource kit indicates that CA EEM is already configured for high availability. The kit continues without re-configuring CA EEM for high availability. Reconfiguration of CA EEM is not required to enable high availability for CA SOI.
7. Repeat Steps 3 through 6 on all cluster nodes.
8. Online the CA SOI cluster resources.
9. Delete the SSAHA_33 directory from the shared drive.
10. Repeat Steps 1 through 9 on all clusters with installed CA SOI components in a multi-cluster deployment.

SSL Implementation

This section describes how to access CA SOI interfaces through SSL and force SSL access for all users.

Access the CA SOI Interfaces through an SSL Connection

Contents

CA SOI is automatically configured to use SSL with the SSL ports that you entered during installation.

Follow these steps:

1. Enter one of the following URLs:
   Dashboard:
   ```
   https://<UIServername>:<SSL port>/sam
   ```
   USM Web View:
   ```
   https://<UIServername>:<SSL port>/ssaweb
   ```
   Mobile Dashboard:
   ```
   https://<UIServername>:<SSL port>/mobile
   ```
   – **UIServername**
     Specifies the server name where you installed the UI Server.
   – **SSL port**
     Defines the UI Server SSL port number that you defined during installation.
     Default: 7403

   A security certificate dialog opens.

Note: For Microsoft Internet Explorer access of CA SOI through the HTTPS protocol, the browser can prompt you with a dialog. The dialog asks, "Do you want to view only the webpage content that was delivered securely?" Click No, which allows access to the configuration pages of the SA Manager when the SA Manager is not configured with SSL.

2. Click Yes to accept the certificate.
3. Enter valid CA SOI user credentials and click OK.

> **NOTE**
> The USM Web View does not accept the Administrator user defined during installation (samuser by default) as valid credentials.

Import SSL Certificate

If you have an SSL certificate, you can also import it.

Follow these steps:

1. Stop the CA SAM User Interface Server service.
2. Browse to the <SOI Install folder>\SamUI\conf location and copy-paste the ssa.jks file to another location.
3. Open a command prompt and navigate to the <SOI Install folder>\SamUI\conf location.
4. Execute the following command: <SOI Install folder>\jre-64\bin\keytool.exe -delete -alias tomcat -keystore ssa.jsk
5. Open a command prompt and navigate to the <SOI Install folder>\SamUI\conf location.
6. Execute the following command: <SOI Install folder>\jre-64\bin\keytool.exe -v -importcert -storepass catalyst -file <Full_Path_Of_Certificate>\<Certificate_Name>.crt -keystore ssa.jks -trustcacerts -noprompt
7. Start the CA SAM User Interface Server service.

Force SSL Connection for All Interface Access

Although CA SOI supports SSL and non-SSL connections, it uses non-SSL by default. You can configure Tomcat on the UI Server and SA Manager to force the interfaces to use an SSL connection.

> **NOTE**
> The Mobile Dashboard already forces an SSL connection.

To enable SSL on the UI Server, f ollow these steps:

1. Stop the CA SAM User Interface Server service on the UI Server.
2. Open **<SOI_HOME>\SamUI\webapps\sam\WEB-INF\web.xml** and change all <security-constraint><user-data-constraint> entries from **NONE** to **CONFIDENTIAL**.
3. Save and close the file.
4. Start the CA SAM User Interface Server service.
5. Run **registrydownloader.bat.**
6. Open **<SOI_Home>\tomcat\registry\topology\physical\node0\sor\solr.properties** and change the URL to **https://localhost:7403/solr**
7. Run **registryloader.bat.**
8. Restart the CA SA Manager Server Service.
9. Enter an HTTP address for the Dashboard from a browser.
   The browser redirects to an HTTPS address and open a security certificate dialog.

If you cannot launch the Operations Console after you enable SSL, uninstall the Internet Explorer Enhanced Security Configuration feature. You can also add 'https://localhost' to the list of Trusted Sites to enable the Operations Console to start locally using SSL.

To enable SSL on the SA Manager Server, f ollow these steps:

1. Stop the CA SAM User Interface Server service and CA SA Manager Server service.

2. Open **<SOI_Home>\SamUI\webapps\sam\server-config.xml** and change the protocol to https and port to **7493**.
3. Save and close the file.
4. Open **<SOI_HOME>\tomcat\webapps\sam\WEB-INF\web.xml** and change all <security-constraint><user-data-constraint> entries from **NONE** to **CONFIDENTIAL**.
5. Save and close the file.
6. Start the CA SA Manager Server service and CA SAM User Interface Server service.
7. Run r**egistrydownloader.bat.**
8. Open **<SOI_Home>\tomcat\registry\topology\physical\node0\sor\wsman.properties** and change the protocol to https and port to **7493**
9. Run **registryloader.bat.**
10. Restart the CA SA Manager Server service.

The SSL is enabled on the SA Manager Server.

Backing Up CA SOI Components

Contents

This section describes how to back up CA SOI components. Consider backing up CA SOI in the following situations:

- If you want to move the installation to another server or reinstall and want to preserve post-installation configuration changes.
- If you want to preserve a working CA SOI installation before making major changes in case errors occur.

The procedures apply to a typical CA SOI deployment. If you implemented a high availability deployment, multi-tiered deployment, or any other specialized deployment, these procedures may not be applicable, and extra steps may be required.

Back Up Post Installation Configuration

If you want to restore your post-installation customization and configuration over an existing or new installation of the same release with all relevant maintenance applied, back up all custom data.

Follow these steps:

1. Back up the <SOI_HOME>\jsw\conf folders on all systems with CA SOI components installed. This folder includes files that define class paths and other properties for the JVM under which the product services run.

   NOTE
   These files include information about system paths. If you are migrating to a new system and the CA SOI installation folder has changed, update the paths to reflect the new installation directory path.
2. Back up the following files and folders on all connector systems and the SA Manager (which always installs the IFW):
   – All connector policy files that you have customized located at <SOI_HOME>\resources\Core\Catalogpolicy
   – The <SOI_HOME>\resources directory, which contains IFW and connector configuration files in the Configurations directory and current and archived event records in the EventStore directory. Also, the log4j.xml file may have been updated to customize the diagnostic logging configuration.

   NOTE
   SSA_IFW_servername.xml and eventManagerServerConfig.xml include hostname information related to the local host, connector hosts, and the ActiveMQ host for connector and SA Manager communications. If you are migrating to a new system, update this information to reflect changes in hostname. You must also update all individual connector configuration files to include the proper hostname.
3. Back up the following data on the SA Manager (or all SA Manager systems in a multi-tiered deployment):

- The SAMStore database using the standard Microsoft SQL Server backup procedure
- All scripts or executables that run as part of escalation or enrichment actions
- The <SOI_HOME>\tomcat\custom directory, which contains files with help desk, CA Process Automation, email, and other connection settings

  NOTE
  If you are moving the installation to another system, set the email host (mailhost) for the email server connection in the Windows\system 32\drivers\etc\hosts file on the new system. If you use the usm_web_view_cfg.xml or mobile_ui_srvr_cfg.xml files when defining help desk properties, you must change the host name if you are moving the UI Server installation.
- The <SOI_HOME>\tomcat\webapps\sam\eem-config.xml file to save the CA EEM configuration settings
- The <SOI_HOME>\jsw\bin\sam_topo_images directory if you have added custom images to the service topology views
- The following CA Catalyst Registry files located at <SOI_HOME>\tomcat\registry\topology\physical\node0\sor if you have customized them in the file system:

  NOTE
  If you have updated Registry files in the Registry Administration UI, they are backed up with the database. Several of the files contain connection settings that you must update if you are moving the installation to a different system.

  a. ssaserver.xml file (default reconciliation formula, contains connection settings)
  b. defaultsheet.xml file (property-specific reconciliation formulas)
  c. singlesourceoftruthmdr.xml file (single source of truth, contains connection settings)
  d. sorapp.xml and plancontrol.xml (synchronization, contains connection settings)
  e. ssaweb.xml file (USM Web View connectivity properties)
  f. ucf-broker.properties.xml (UCF Broker properties)

4. (Optional) Run the following command from <SOI_HOME>\plugin\ServiceDiscovery\bin in a command prompt if you want to back up defined service discovery policies:

   ```
   read_rules_from_db.bat filename.xml
   ```

   All service discovery policies are saved to the specified XML file. The policies reside in the database and should be backed up with the database. However, you can back up the service discovery policies to a file for extra protection or if you want to reload the database.
5. (Optional) Back up the <SOI_HOME>\plugin\UniversalConnector conf and jsw folders if you are using the Universal connector.
6. Back up the following data on all UI Server systems:
   - The <SOI_HOME>\SamUI\conf\soi_conf \mobile.properties file if you have changed any Mobile Dashboard connection or timeout properties
   - The <SOI_HOME>\SamUI\webapps\mobile directory if you have customized the Mobile Dashboard metric icons or performed a custom standalone Mobile Dashboard implementation
   - The <SOI_HOME>\SamUI\webapps\mobile_extra_style.css file if you have customized the Mobile Dashboard display
   - The <SOI_HOME>\SamUI\webapps\sam\eem-config.xml file to save the CA EEM configuration settings
   - The <SOI_HOME>\SamUI\custom\email-config.xml file to save the email server configuration for the Operations Console

NOTE
If you are moving the installation to another system, set the email host (mailhost) in the Windows\system 32\drivers\etc\hosts file on the new system.

- The <SOI_HOME>\SamUI\custom\common\config\custom-jnlp-config.xml file to save changes in the JNLP configuration
- The <SOI_HOME>\SamUI\custom\console\config\custom-menu-config.xml if you customized the Operations Console menu
- The <SOI_HOME>\SamUI\webapps\sam\thinuiconf\custom_metric_definition.xml file if you added custom metrics to the Dashboard
- The <SOI_HOME>\SamUI\webapps\sam\thinuiconf\tables_definition.xml file if you customized the service hierarchy on the Dashboard
- The <SOI_HOME>\SamUI\webapps\sam\ui\refresh.properties file if you customized the dashboard refresh rate
- Any custom images in the <SOI_HOME>\SamUI\webapps\sam\ui\images directory if you customized the dashboard logo
- Any custom PNG file in the SamUI\webapps directory if you configured a custom Mobile Dashboard logo
- The <SOI_HOME>\SamUI\webapps\sam\WEB-INF\console\config\custom-menu-config.xml file if you added custom Dashboard links

7. (Optional) Back up CA Business Intelligence and reporting systems by following the backup procedures documented in theSAPBusinessObjects documentation.
8. Follow the backup procedures documented in the CA EEM *Getting Started Guide*, including how to back up CA EEM data in CA Directory.

Back Up a Full Installation

If you intend to restore a full CA SOI implementation to the original installed configuration, back up the following directories:

- The <SOI_HOME> directory and all subdirectories on all hosts where CA SOI components are installed. This preserves all configuration and customization changes for CA SOI and all maintenance applied post installation.

CA SOI Uninstallation

This section describes how to uninstall CA SOI and related components.

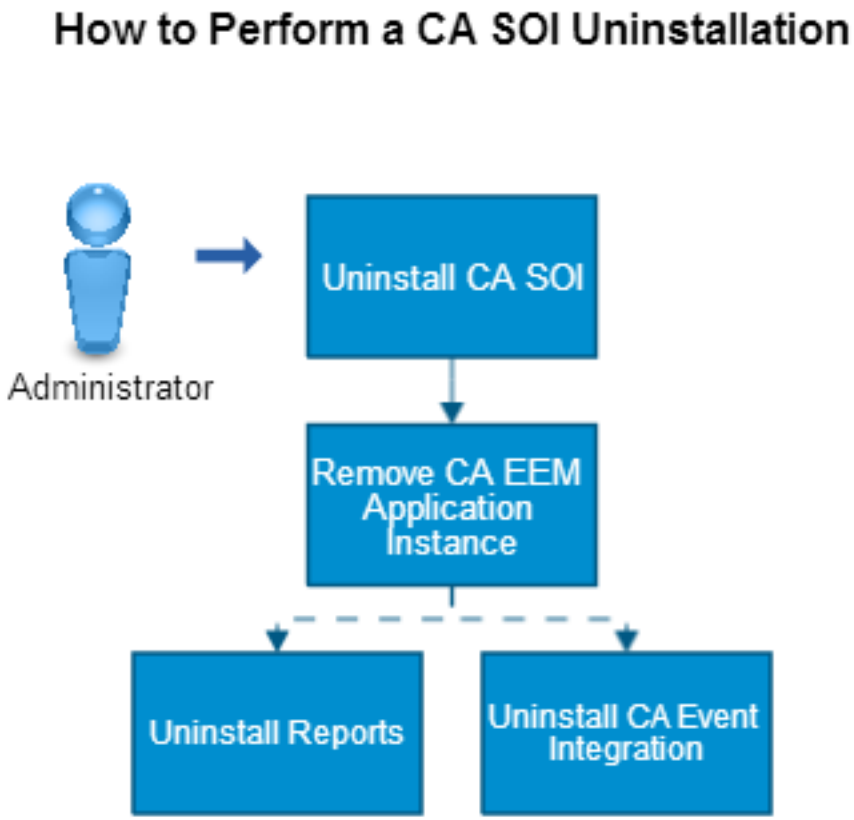
How to Perform a CA SOI Uninstallation

Contents

As an administrator, you can uninstall CA SOI. Completely uninstalling a full CA SOI deployment requires you to remove many components in a specific order.

Use this scenario to guide you through the process:

Figure 25: how to perform ca soi uninstallation



1. Uninstall CA SOI:
   - Connectors
   - UI Server
   - SA Manager
   - MQ Server
2. Remove CA EEM application instance.
3. (Optional) Uninstall reports.
4. (Optional) Uninstall CA Event Integration.

Uninstall CA SOI

Uninstall CA SOI components in the following order:

1. Connectors
2. UI Server
3. SA Manager
4. MQ Server

If the UI Server and SA Manager share a system, they uninstall simultaneously from the same program.

NOTE
User preferences are not removed when you delete the product because the preferences are stored in CA EEM.

Follow these steps:

1. Uninstall connectors, and uninstall the IFW from connectors systems.

   NOTE
   For general information about uninstalling connectors, see Uninstall Connectors. For connector-specific information*, see* the *Connector Guide* provided with the connector.
2. Click Uninstall.
   Uninstallation begins, and a summary displays when the process completes.
3. Repeat Steps 2-3 on all additional systems that contain any of the following components:
   – SA Manager
   – UI Server
   – MQ Server
4. Remove the SA Store database from the system where you installed it, or ask your database administrator to remove it.

   TIP

   If you are having trouble reinstalling CA SOI, check for the existence of the Zero G Registry folder under C:\Program Files. Open the file to see whether applications other than CA SOI are tracked. If CA SOI is the only application in the file, delete the file. If other applications are in the file, delete the line for CA SOI. Zero G Registry is a hidden folder, so ensure you can see hidden folders before looking for it.

Remove CA EEM Application Instance

After you uninstall CA SOI, remove the application instance in CA EEM for importing users into CA SOI.

Follow these steps:

1. Select Start, Programs, CA, Embedded Entitlements Manager, EEM UI.
   The login page of the CA EEM user interface opens.
2. Enter CA EEM administrator credentials.
3. Click Log In.
   The Home tab of the CA EEM user interface opens.
4. Click the Configure tab.
   The Applications pane of the Configure tab opens.
5. Click the application instance that was created for CA SOI. By default, this instance is prefixed with SOI.
   The Application Instance pane opens with details about the instance.
6. Click Unregister.
   A confirmation dialog opens.
7. Click OK.
   The application instance is removed.

Uninstall SOI Reports and CABI JasperReports Server

When you install SOI Reports, if you had installed SOI Reports and CABI JasperReports Server together, you can uninstall them together.

Follow these steps

1. Click Start, All Programs, CA, Uninstall CA SOI-Reports.
2. Select the check box if you want to remove the CABI JasperReports Server database and click Uninstall.

Uninstall CA SOI Reports and Retain CABI JasperReports Server

If you had installed SOI Reports without CABI JasperReports Server, you can uninstall them individually.

Pre-Requisites

- CABI JasperReports Server and Tomcat should be running.
- If CABI JasperReports Server or Tomcat was uninstalled, you cannot uninstall CA SOI Reports.

Follow these steps

- Click Start, All Programs, CA, Uninstall CA SOI-Reports.
- Click Uninstall in the Uninstall CA Service Operations Insight - Reports page.
- Type the required information to confirm the JasperReports Server with SOI Reports, and click Next.
  - **Select Protocol:** Specify HTTP or HTTPS
  - **JasperReport Server Port:** Specify the port number
  - **Jasper Server instance name:** Type the web app name. Example: jasperserver-pro
  - **User Id/Password:** Type the login credentials of the user with administrator privileges who does not belong to an organization in JasperReports Server. **Default:** superuser/superuser
- Click Next, confirm the summary, and click Uninstall.

> **NOTE**
>
> Click here to view the steps to uninstall CABI JasperReports Server.

Uninstall CA Event Integration

If you uninstall CA SOI and you also want to remove the Event connector, you also uninstall CA Event Integration where you installed the Event connector. If you are using CA Event Integration apart from CA SOI, you can retain CA Event Integration.

Follow these steps:

1. Select Start, Programs, CA, Event Integration, Uninstall CA Event Integration on the Event connector system.
2. Click Next.
3. Select the Remove database check box if you want to delete the database and click Uninstall.
   The uninstallation initializes, runs, and completes. A page appears summarizing the uninstallation. This page can list files or directories that were not removed.
4. Click OK.
   When the dialog closes, further cleanup is performed. Verify the directories listed in the summary to ensure that they were deleted.

Clean up Windows User Information

When you uninstall CA Event Integration on Windows, the following directories and registry entry that is related to the operating system user created with the product (ca_eis_user by default) may persist:

- C:\Users*username* (Windows 2008)
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileLists\S-1-5-21-...

> **NOTE**
> The full name of the registry entry varies.

If you uninstall CA Event Integration and you do not plan to reinstall, remove these persistent files.

Follow these steps:

1. Delete the C:\Users*username* (Windows 2008) directory if it still exists.

2. Run the delprof.exe utility that is provided with Windows to delete all other materials, including the registry entry that is associated with the user profile (and other inactive user profiles).
   If you do not have this utility, you can download the utility from the Microsoft website.

Uninstall Connectors

Uninstall connectors to remove the connector software. After uninstallation, you must perform the operations described in Connector Removal to fully remove references to the uninstalled connector in the product.

Follow these steps:

1. Select one of the following from the Start menu on the connector system:
   – Connectors provided with the CA SOI image: Start, Programs, CA, Service Operations Insight, Uninstall *Connector Name*.
   – CA Catalyst connectors: Start, Programs, CA, Service Assurance Manager, Uninstall *Connector Name*.
   The Uninstall *Connector Name* dialog opens.
2. Click Uninstall.
   The connector uninstalls. If the connector is the only one installed on the system (and no other CA SOI components are also installed), the IFW is uninstalled. If other connectors exist on the system, the IFW remains.

Connector Removal

Contents

You can either permanently or temporarily remove a connector from the CA SOI database and all interfaces:

- Permanently remove an uninstalled connector.
- Temporarily disable an installed connector.

Remove an Uninstalled Connector

After you uninstall a connector, remove the database references using the Administration tab. This also removes the connector database entry and the connector name from the list of connectors in the tree view.

If the connector comes back online after the operation is completed, the connector reregisters with CA SOI.

Follow these steps:

1. Select the Administration tab and the Connector Configuration option.
2. Select the name of an offline connector in the Connector column.

   NOTE
   You cannot remove an online connector.
3. Click Remove Connector and confirm the deletion.
   The selected connector registration is permanently removed from the CA SOI database, and the connector name is removed from the tree on the Administration tab.

Disable an Installed Connector

You can disable an installed connector if you want to remove connector records temporarily, but keep the connector installed for potential future use.

WARNING
To ensure data integrity, perform this operation after business hours for connectors with a large amount of CIs and relationships. Removing the database information for these CIs and relationships locks the system for approximately ten minutes for every 100,000 CIs managed by the connector.

NOTE
If you are using CA Service Desk with the auto clear alerts option enabled, then shut down or remove a connector, CA SOI automatically changes the cleared alerts to the selected status. You cannot undo this operation. For more information, see How to Work with Configured Help Desk Integrations.

Follow these steps:

1. Select the Administration tab and the Connector Configuration option.
2. Verify that the Connector Service status is Online for the connector to disable.
   The IFW for the connector must be running for the disable operation to work. If it is not running, start the CA SAM Integration Services service on the connector system.
3. Select the connector to disable.
4. Click Stop if the connector appears as Online.

   NOTE
   You cannot disable an online connector.
5. Click Remove Connector after the connector appears as Offline and confirm the disable.
   The selected connector's registration is removed from the CA SOI database and the connector name is removed from the tree on the Administration tab. However, if the connector is still installed, all of its required files still exist in case you want to re-enable the connector.

Enable a Disabled Connector

You can enable a connector that you previously disabled to bring it back online and collect data from its domain manager again.

Follow these steps:

1. Open the connector configuration file that in the <SOI_HOME>\resources\Configurations folder on the connector system.
   The format of the connector configuration is typically the connector name followed by the connector system name (for example, sampleConnector_server1.xml).
2. Change the State property value from not Enabled to Enabled and save the file.
3. Restart the CA SAM Integration Services service on the connector system.
   The connector entries reappear in the Connector Configuration tree and in all other interfaces, and the connector begins collecting data from its domain manager.

Connector Removal Success and Failure Messaging

CA SOI provides an Administration tab confirmation message that indicates success or failure upon connector removal.

NOTE
This procedure assumes that you have already uninstalled the connector and are trying to remove its data from the CA SOI interfaces.

Follow these steps:

1. On the Dashboard, click the Administration tab.
2. Click the plus sign (+) next to the Connector Configuration.
3. Click the plus sign (+) next to a connector.
4. Click the connector.
5. If the connector is still online, click Stop.
6. Click Remove Connector.
   A confirmation appears at the top of the page that provides the connector removal success or failure.

Uninstall CA SOI HA Manager

Contents

As an administrator, when you uninstall a highly available CA SOI implementation, do not destroy the CA SOI database. Wait until the installation has been removed from the last cluster node.

Follow these steps:

1. Offline the CA SOI cluster resources.
2. Uninstall CA SOI as you would in a non-clustered uninstallation, but do not remove the SA Store database yet.
3. Move the cluster group to the next node.
4. Repeat the uninstallation on all cluster nodes.
5. Do the following on the last cluster node only:
   a. Remove the SA Store database.
   b. Delete CA SOI cluster resources after the uninstallation completes.
   c. Delete all files in the SOI_HOME directory on the shared drive.

Disable CA EEM Data Store Replication

You can disable the CA EEM data store replication as part of a CA SOI HA uninstall without uninstalling CA EEM.

Follow these steps:

1. Run the following command:
   ```
   Dxserver stop all
   ```
   All CA EEM services are stopped.
2. Copy the contents of the <CA_DIRECTORY_HOME>\dxserver\config\knowledge_Habackup directory to <CA_DIRECTORY_HOME>\dxserver\config\knowledge.
   All customized knowledge files are restored.
3. Run the following command:
   ```
   Dxserver start all
   ```
   CA EEM services are restarted.

Uninstall CA SOI HA Connector

As an administrator, you can uninstall the CA SOI high availability connector. Each CA SOI connector provides a Start menu uninstallation link. If an installed connector is no longer required, uninstall the connector on all cluster nodes. After uninstallation, further steps are required on the cluster node to remove the connector completely.

> **WARNING**
> Do this procedure on the last cluster node on which you are uninstalling the connector.

Follow these steps:

1. Delete the connector configuration file from the shared drive.
2. Offline the CA SAM integration Services cluster resource to begin removal of the connector and wait until the offline status is complete.
3. Online the CA SAM Integration Services cluster resource to finalize removal of the connector.
   The CA SOI HA Connector is uninstalled.

Administrating

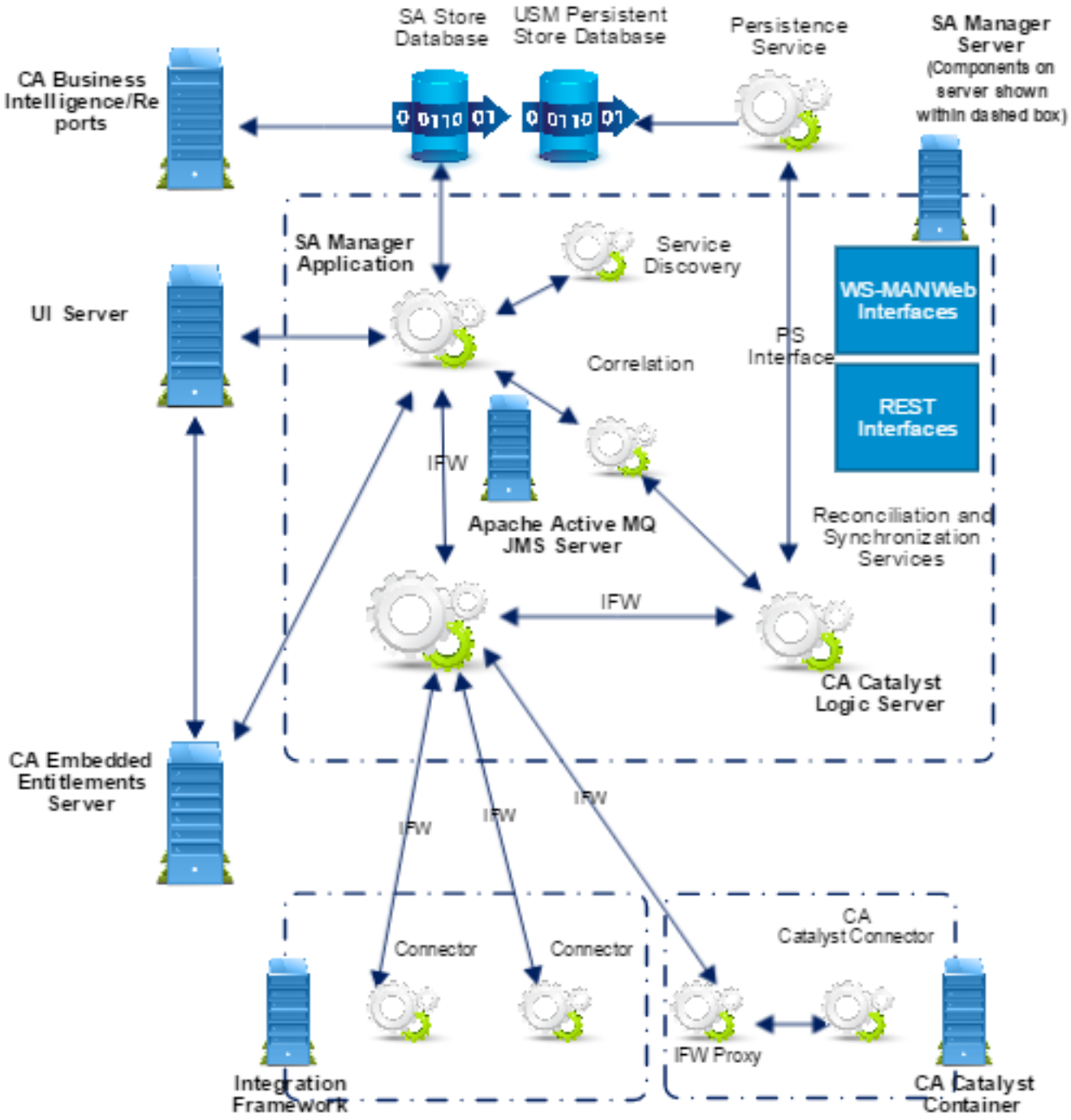
An administrator is responsible for the initial set up and configuration of the application, and for enabling the application to process all tasks as per the needs of the organization. Configuring the application includes tasks such as setting up various records in the application, creating required workflow actions, enabling permission to the users to access the required application components, and ensuring that the software is able to enhance productivity and efficiency as required.

CA SOI Architecture

Contents

The following graphic summarizes the product architecture:

Figure 26: product architecture



This section describes the key components in this diagram and their product function.

Integration Framework

The *integration framework (IFW)* is the mechanism that CA SOI uses to connect to domain managers and gather CI, service, topology, and state information.

It exists on any system with a connector or the SA Manager, and it interfaces with the connector framework to prepare connector data for transmission to the manager components. The IFW contains a transformation engine that uses a connector policy to transform connector data to the USM format. The IFW also includes the infrastructure of the Event Management component. The component provides the mechanism for storing events from connectors for exposure to event policy and eventual display as alerts after event processing completes.

The IFW uses the Apache ActiveMQ message broker, which fully implements the Java Message Service (JMS) as its protocol.

MQ Server

Apache ActiveMQ is an open source message broker that fully implements the Java Message Service 1.1.

The MQ Server controls all messaging and communication from external sources. The server also receives alerts and CI information from connectors through the IFW and sends this information to various components for storage and analysis. This component is always installed with the SA Manager.

Connectors

A connector is software that provides the interface for the data exchange between the CA Catalyst infrastructure and a domain manager. Connectors are the gateway through which data is retrieved from various domain managers for a consolidated management. Each integrated product has its own connector that supports one or both of the following operation types:

- **Outbound from connector**
  Outbound from connector operations obtain data (such as services, CIs, topology, alerts, and status) from the source domain manager. All connectors must implement outbound operations. Outbound data populates the CA Catalyst Persistence Store.
  Outbound data flows to one or more clients. Clients such as CA Catalyst consume the data to implement a unified view of data from multiple domain managers and their connectors.
- **Inbound to connector**
  Inbound to connector operations (also referred to as "southbound") use records in the CA Catalyst Persistent Store and the CA Catalyst Synchronizer to create, update, or delete items in the source domain manager. The inbound operations enable domain manager synchronization with the changes that CI reconciliation, CI creation, and CI updates initiate in other domain managers.
  Many provided connectors support inbound operations. Connectors that implement inbound operations sometimes limit the implementation to a subset of the types and properties their outbound operations support.
  A *bidirectional connector* supports both inbound and outbound operations. Outbound-only connectors contain one connector policy file that transforms the gathered data to the standard USM format. Bidirectional connectors contain two connector policy files that transform outbound data to the USM format and transform inbound data to the source format of the domain manager.

You can configure connectors and start and stop them by accessing the CA SOI Administration UI.

CA SOI also provides the following tools for defining custom integrations:

- The Universal connector that can retrieve services, CIs, and status events from various CA Technologies and third-party products. The Universal connector provides a web services interface that products can use to publish new services, CIs, and events, which are normalized to a common format and made available to the SA Manager.
- A connector SDK for developing custom connectors. The SDK includes a Sample connector, which provides the framework for writing a connector to integrate with important applications in your enterprise.
- An Event connector that collects events from low-level event sources, transforms them into the CA SOI alert format, and displays them as infrastructure alerts in CA SOI associated with existing or created CIs.

NOTE
For more information about the connector architecture and how to build custom connectors, see Connectors Overview. Each connector also ships with a connector-specific *Connector Guide* that contains information about connector installation, configuration, how the connector interprets data from its domain manager, and whether it supports inbound operations.

CA Catalyst Infrastructure

CA SOI fully adopts the CA Catalyst integration platform as its infrastructure. CA Catalyst is the CA Technologies common integration platform that provides the groundwork for unifying data from all CA Technologies products and many third-party products. CA Catalyst is fully embedded in the SA Manager installation. CA Catalyst provides the following functionality:

- A common semantic schema for data from all integrated products
- CI reconciliation to ensure that resources managed in multiple products have a unified set of property values
- CI synchronization that triggers bidirectional connector updates to source domain managers according to CI reconciliation and other operations
- The ability to enact specific use cases (including use cases that were available in previous releases of CA Catalyst), and the ability to manipulate the infrastructure to configure custom use cases, reconciliation formulas, and synchronization rules

Unified Service Model

The *Unified Service Model (USM)* is the semantic schema that is used as the CA Catalyst and CA SOI infrastructure.

Connectors transform all data that is collected from domain managers to the USM format before sending the data through CA Catalyst. The USM schema is stored in the CA Catalyst Registry.

USM is a high-level abstraction and generalization of IT management concepts that facilitate the semantic merging and interoperability of more specific domains. USM is developed to abstract and integrate information across many management products and domains. USM provides a single point for data federation, interoperability, and access to management data across an enterprise.

CA Catalyst provides the mechanisms to make all outbound from connector data adhere to the USM schema. CA SOI provides the interfaces to display the USM-compliant data.

Persistence Service

The Persistence Service enables other components to manipulate the Persistent Store. Operations such as reconciliation and synchronization require CA Catalyst components to modify the USM data. The Persistence Service enables interactions through a flexible interface that supports the following operations:

- Creating and storing USM data in response to incoming CIs from connectors
- Updating and deleting USM data in response to the reconciliation that the Logic Server performs
- Retrieving the USM details for display by the USM Web View
- Creating and updating USM data in response to operations initiated from the USM Web View

The Persistence Service provides an abstraction layer for working with data in the Persistent Store, which is the database record of USM data.

Logic Server

The CA Catalyst Logic Server provides the logic and the modules that carry out the following operations:

Reconciliation

Creates a unified set of properties and values from instances of a single entity that multiple connectors retrieve. The Logic Server reconciles CIs using formulas that define the property values to use. The policy rules that you can define include first non-null wins, majority wins, and data from a specific domain manager wins.

Synchronization

Detects the following CI changes within the Persistent Store:

- reconciliation
- CI creation in the USM Web View
- CI update or deletion in a source domain manager, or through some other method

Synchronization pushes the changes to applicable domain managers integrated through bidirectional connectors. Synchronization policies can create specific synchronization rules or use cases that keep source domain managers synchronized with the USM data.

The Logic Server lets CA Catalyst create and maintain a unified set of reconciled, correlated data in the Persistent Store. From the Operations Console, you can view the reconciled set of USM properties for any CI, named the reconciled sheet. You can also view the USM notebook for any CI. The notebook lists the reconciled sheet and the USM properties for each managed instance of the CI in source domain managers.

Registry

The CA Catalyst Registry is the repository for the USM schema and the policies that control the behavior of the Logic Server. You can access the Registry Administration UI from the CA SOI Administration UI to manipulate the Logic Server policies that control reconciliation and CI synchronization.

UCF Broker

The UCF Broker is a communication layer that controls access to the enabled bidirectional connectors, which can invoke inbound to connector operations on source domain managers. The Logic Server communicates synchronization changes to bidirectional connectors through the UCF Broker.

SA Store

The SA Store is the central repository for all CA SOI configuration and management data. It is a relational database from which the other CA SOI components retrieve their configuration policy and the read-write management data about the state of services and resources. The SA Store includes the following components:

- The CA Catalyst Persistent Store, which maintains a record of reconciled USM data (CIs, alerts, and relationships)
- Tables that contain data specific to CA SOI, such as escalation policies and service models

SA Manager

The SA Manager integrates the data that the connectors send:

- Correlates data so that CIs managed in multiple products are managed as one entity in CA SOI
  - Updates the Persistent Store with USM data
  - Provides correlation information to the Logic Server for reconciliation
- Manages CI and service status as follows:
  - Monitors the health and availability of managed CIs and services
  - Performs service impact and risk analysis
  - Monitors service-level agreements against defined thresholds
  - Updates the SA Store with analysis results and state changes
- Provides event and alert management functionality as follows:

  - Event policies to filter, correlate, and enrich events in the event store
  - Federated query of events across all integrated domain managers
  - Management of service impacting and all non-service impacting alerts
  - Alert queues to manage alerts by common properties
  - Escalation policies that can automate the response to alert occurrence, such as creating a help desk ticket, sending an email, and running a custom script or a CA Process Automation process

UI Server

The User Interface Server (UI Server) is the server that hosts the user interface applications. The UI Server is hosted within a web server, and you can deploy multiple UI Servers in a single CA SOI installation to support load balancing.

CA SOI has the following user interfaces:

- **Operations Console**
  Supports all administrative functions, including service modeling, defining alert queues and defining associated policy, and provides an operational view of the data for analysis purposes. Operators and other technicians use this interface to view and respond to alerts that report fault conditions. Administrators use this interface to define users and user groups, set role-based security, create and maintain service definitions, and more.
- **Dashboard**
  Displays service data that is tailored to the role of the user. Managers and others use this interface to analyze the overall health and availability of monitored services. They can also determine who is resolving problems and when those problems are fixed.
- **Mobile Dashboard**
  Provides content similar to the Dashboard in a format suitable for mobile devices. The Mobile Dashboard also lets you view service, customer, and alert queue details and take actions on alerts.
- **Report Console**
  Displays several types of scheduled and on-demand reports for service data in a portal-style interface. The reports provide service stakeholders with historical information that includes details about the availability and risk of a service.
- **Administration tab**
  Provides the tools to maintain connectors and SA Manager settings. The Administration tab also lets you configure single sign-on using CA EEM, email notifications, and other administrative functions.
- **USM Web View**
  Lets you browse and search all USM data in the Persistent Store. You can use the USM Web View to locate specific information, browse data based on many different criteria, and subscribe to RSS feeds to be notified of updates to specific CIs. This interface also lets you create new CIs and update existing CI information.
- **Debug Pages**
  The CA SOI Debug pages let you test and debug various CA SOI components. For more information, see the Debug Consoles.

CA EEM

CA Embedded Entitlements Manager (CA EEM) provides role-based authentication services for the CA SOI user interfaces and supports single sign-on across most interfaces. Single sign-on (SSO) requires all applications participating in SSO to use the same CA EEM server.

CA Business Intelligence

CA SOI implements BusinessObjects, which is a third-party business intelligence platform that provides interactive reporting. CA Business Intelligence hosts predefined CA SOI reports, which include scheduled and on-demand reports.

General Administration

This section contains general administration tasks that CA SOI administrators can perform after installation.

SA Manager Details

Contents

View SA Manager Details

As an administrator, you can view details about the SA Manager to help troubleshoot messaging and database connection issues. You can also access the Web Server Debug page and server log pages from this page. For more information about troubleshooting and using the CA SOI debug pages, see Debug Consoles.

Follow these steps:

1. Click the Administration tab.
2. Click the plus sign (+) next to the CA Service Operations Insight Manager Configuration option.
3. Click the name of the server you want details about.

Configure SA Manager Correlation Cache File

Whenever CA SOI service is stopped, a cache file is created at <SOI_Install>\tomcat\work\soi.correlation.cache. The cache file contains the key property values of a CI from one domain manager to another domain manager. The cache file data helps in the quick restarting of SA Manager. Once the SA Manager is up and running, the cache file was deleted. So, if the SA Manager was shut down abruptly, the cache file could not be created and SA Manager took considerable time to restart.

Starting CA SOI 4.0 SP2, the SA Manager cache file is not deleted. Also, administrator can define the time interval to update the cache file. The more the cache file is updated at the time of abrupt shutdown, lesser the time to reboot.

To change the default time interval, follow these steps:

1. Locate and open the following file on the SA Manager: <SOI_Install>\jsw\conf\soi-manager.conf.
2. Edit the interval in the following section: wrapper.java.additional.n=-Dcache.write.interval=7200
   Default: 7200; **Units:** seconds

View UI Server Connection Details

Contents

As an administrator, you view details about the CA SOI UI Servers that are connected to the SA Manager server and active user sessions and enable debugging. Typically, one UI Server for each SA Manager though multiple UI Servers are supported.

> **NOTE**
> You can perform this procedure on the SA Manager server only. The UI Server has a similar feature where you can view details about the users that are logged in to the Operations Console.

Follow these steps:

1. Click the Administration tab.
2. Click the plus sign (+) next to CA SOI Insight Manager Configuration.
3. Click the plus sign (+) next to the server you want to configure.
4. Click UI Server Connection Details.
   A table displays that provides details about each active user session.

5. (Optional) Click the link in the Debug column to switch between enable and disable for the corresponding UI server. When the Debug option is set to enable, client debugging is enabled and the ClientPollServlet outputs all requests and other activity for that client.

View and Manage the UI Server Connection Log

The CA SOI installation application installs a client.log file on the SA Manager. By default, it is located at <SOI_HOME>\tomcat\webapps\sam\console\logs.

The client.log file contains an entry for every user who logs in to the Operations Console. The Client Log page allows you to view the contents of client.log file. You can also use this page to clear the log and remove old entries.

NOTE

You can perform this procedure on the SA Manager only. The UI Server has a similar feature where you can view and manage the users that are connected to the Operations Console.

Follow these steps:

1. Click the Administration tab.
2. Click the plus sign (+) next to CA SOI Insight Manager Configuration.
3. Click the plus sign (+) next to the server you want to configure.
4. Click UI Server Connection Log.
5. (Optional) Type a number in the 'Purge entries older than # days' field, and click Go.
   The log entries older than the specified number of days disappear from the log.
6. (Optional) Click Clear Log to clear the log entries.

Configure Single Sign-On

Contents

As an administrator, you can configure single sign-on (SSO) with CA EEM and CA Siteminder.

NOTE

The Single Sign-On integration does not work when LDAP integration is enabled in CABI JasperReports Server. The LDAP integration is not required as the JasperReports Server can be accessed without prompting for user credentials.

Configure CA Embedded Entitlements Manager Single Sign-On

As an administrator, you can enable single sign-on with CA EEM. The SA Manager and UI Server use CA EEM to authenticate user credentials against multiple applications.

Establish or update CA EEM connection settings in the following circumstances:

- You did not define CA EEM connection settings during the CA SOI installation.
- The password for the EiamAdmin user changes.

NOTE
Other changes to the CA EEM server can also require you to update the CA EEM configuration settings.

Follow these steps:

1. Click the Administration tab.
2. Click the plus sign (+) next to the CA Service Operations Insight Manager Configuration or CA Service Operations Insight UI Server Configuration option.

3. Click the plus sign (+) next to the server you want to configure.
4. Click the EEM Configuration option.
5. Specify the following parameters:

   NOTE
   If you do not know these values, refer the Installation Worksheet for the CA EEM values. For more information, see Obtain the Installation Worksheet.

   – **EEM Server Host**
     Defines the name of the server where CA EEM is installed.
   – **User Name**
     Defines the name of the CA EEM administrator user (typically EiamAdmin).
   – **Password**
     Defines the password of the CA EEM administrator user.
   – **Application Name**
     Defines the application name CA EEM uses for CA SOI user management. The entry uses the following format: SOI_*soi_server_name*.
   – **Change Password Allowed**
     Specifies whether users can change their password from the CA SOI interface. Select this box from the UI Server EEM Configuration page to enable password changes from the CA SOI interface.
6. Click Test to verify that the settings are valid on the CA EEM server, and click Save.
   – If you clicked Test, the settings are temporarily applied and your user credentials are validated. If errors are displayed, fix them before saving or retesting.
   – If you clicked Save, the settings are saved and a confirmation message displays.
7. (Optional) Click Launch EEM to open the CA EEM login page.

   NOTE
   The login page opens and the CA SOI instance you specified in the Application Name field is added to the CA EEM login page Application drop-down list.

WARNING
Update the CA EEM configuration settings for the SA Manager, and the UI Server.

NOTE
If the Administration tab changes do not save or the fields clear when you click Save or Test, adjust your browser settings. One of the following actions should solve the problem:

- Add the CA SOI URL to your trusted sites and lower the privacy to accept all cookies. For more information about trusted sites and privacy settings, see your browser documentation.
- Add the domain name of the CA SOI UI Server and SA Manager (for example, company.com) to the privacy managed websites and set to Allow. For more information about managing privacy settings, see your browser documentation.

CA SiteMinder Single Sign-On

You can use CA SiteMinder to enable single sign-on across products that are using the same CA EEM server. For example, if you log in to CA Spectrum and use the same browser to log in to CA SOI, the CA SiteMinder session token is passed to CA SOI through CA EEM. You do not have to log in again.

To use this feature, use the same CA EEM server to manage users for multiple products, then integrate the CA EEM server with CA SiteMinder.

Configure CA SOI Reports Single Sign-On

Single sign-on (SSO) is available for reports, so that you can run reports from the CA SOI Dashboard without providing the login credentials for CABI JasperReports Server. You can still access reports without SSO configured, but you need to enter the login credentials each time.

Perform following steps in the system where CA SOI Reports is installed:

1. Browse to the following location: <SOI_REPORT_HOME>\Reports\deploy\keystore.
2. Open the soi.properties file, type plain password value in place of "<STORE_PASSWORD>" against key "ks.storepass", and save the file.
   Example: ks.storepass=jasper-soi-encrypt
3. Browse to the following location: <SOI_REPORT_HOME>\Reports\deploy
4. Execute the jkscreation.bat file to create the jks file.
5. Create a folder and name it as "config" in the following location: <Jaspersoft Install Folder>\<tomcat folder>\webapps\<Jasper_Instance_Name>\WEBINF\
   Example: D:\Program Files (x86)\CA\SOI\Reports\jr\apache-tomcat-7.0.68\webapps\jasperserver-pro\WEB-INF
6. Copy soi.jks and soi.properties from the following location: <SOI_REPORT_HOME>\Reports\deploy\keystore and paste to the location <Jaspersoft Install Folder>\<tomcat folder>\webapps\<Jasper_Instance_Name>\WEB-INF\config\
7. Restart CABI JasperReports Server.

Perform the following steps in the system where CA SOI UI Server is installed: Do these steps after you configure single sign-on in the CA SOI Reports system.

1. Copy soi.jks and soi.properties file from the CA SOI Report system and paste it at any location.
   Example: C:\Program Files\CA\SOI\SSO
2. Create an environment variable named CA_TRUSTEDAUTH_HOME and set the path where jks and properties files is copied in Step 1.
   Example: CA_TRUSTEDAUTH_HOME=C:\Program Files\CA\SOI\SSO
3. Restart the CA SOI User Interface service.

Manage CA SOI Reports User Group in CABI JasperReports Server

Contents

Roles and Privileges

Some of the commonly used roles in CABI JasperReports Server are as follows:

- ROLE_USER: This is the role assigned to a user by default. This role has Read access to all SOI reports.
- ROLE_SOI_REPORTS: This is the custom role with Read and Write access to a user to access SOI Reports.
- ROLE_REPORT_DESIGNER - Users in this role can create reports, adhoc views, and dashboards.
- ROLE_DOMAIN_DESIGNER - Users with this role can create data sources, domains, adhoc views and adhoc reports.

By default, any valid SOI user will be assigned to ROLE_USER. Users with administrator privileges in CA SOI can provide assign a user to other roles.

To view the other default roles provided by CABI JasperReports Server, click here.

Adding Users

A valid SOI user alone will get access to view or run reports in CABI JasperReports Server. You can add users to the SOI report group in CABI JasperReports Server or can remove users from CABI JasperReports Server. However, if you add or delete users in CABI JasperReports Server, it does not reflect in CA SOI.

Follow these steps:

1. Configure Report Server on CA SOI Administration Dashboard.
   The report server is configured.
2. Open the Operations Console.
3. Click Users and add or remove a user.
4. Click Reports.
   The CABI JasperReports Server login page appears.
5. Type the credentials.
   Organization: Leave it blank
   User ID/Password: Type the login credentials of the user with administrator privileges who do not belong to an organization in JasperReports Server. **Default:** superuser/superuser
6. Click Manage, Users.
   The user added or removed reflects accordingly in CABI JasperReports Server.

   NOTE

   If you have added or removed users without selecting the check boxes in step 5 and if you restart the SOI Manager services, the change reflects in CABI JasperReports Server.

Configure CA SOI Reports for CABI JasperReports Server

Contents

Configure Reports Link on the Dashboard

As an administrator, you enable the report functionality and configure CA SOI to access the CA Business Intelligence JasperReports Server.

You can also configure CA SOI to automatically synchronize CA SOI and CA Business Intelligence JasperReports Server users. User synchronization lets you add or remove a user in the Operations Console and have CA SOI automatically add or remove the user from CA Business Intelligence access.

Follow these steps:

1. Click the Administration tab on the Dashboard.
2. Click the plus sign (+) next to CA Service Operations Insight UI Server Configuration.
3. Click the plus sign (+) next to the server you want to configure.
4. Click Report Configuration.
5. Type the appropriate information in the following fields and click Save:

   NOTE
   Refer to your Installation Worksheet in the CA Business Intelligence section for these values.

   – **Report Server**
     Specifies the server name where CA Business Intelligence JasperReports Server and the CA SOI Reports component are installed.
   – **Report Port**
     Specifies the port number on which the report server is listening. If you used the default port when installing CA SOI, it is set to 8080 for report servers using Tomcat.
   – **Report URL**
     Specifies the interface on the report server.
     Example: jasperserver-pro/login.html
   – **Enable SSL**

Specifies whether to use SSL to communicate with the specified report server. Select this check box if the report server is configured to use SSL.

– **Enable automatic additions to the CA Service Operations Insight report group**
Specifies whether users added to the Operations Console are automatically added to CABI JasperReports Server and the SOI Reports group on the report server.
Default: Selected

NOTE

- If you create users in the Operations Console without selecting this option, you must select this option and restart the CA SAM Application Server service to add the users to the user group in soi organization in JasperReports Server.
- The users that are automatically added to the CABI JasperReports Server SOI Reports group are assigned a temporary password of SAM*userid* (for example, if the user name is jeromeG, the password is also SAMjeromeG).
- Do not use the users in the CA SOI Super Users group (including the "samuser" super user) to run reports.
- Do not use the CABI JasperReports Server administrator user (superuser by default) to run reports.
- If you configure the automatic addition of CA SOI users for reports, do not create a user in CA SOI with the same name as the CABI JasperReports Server administrator user (superuser by default). The same name can cause problems with the CABI JasperReports Server administrator user.

– **Enable automatic removal from the CA Service Operations Insight report group**
Specifies whether members of the SOI Reports group (on the CABI JasperReports Server report server) are automatically deleted from the group and CABI JasperReports Server user list when they are deleted using the Operations Console.

NOTE
Users that are removed using this functionality are deleted from *all* user groups not only the SOI Reports group.

– **User ID/Password**
Specifies a user with administrator privileges who does not belong to an organization in JasperReports Server. This user must have the right to create users on the CA Business Intelligence JasperReports Server.
Default: superuser/superuser

6. Click Save & Test at the top of the page and then click Refresh.
The Reports link becomes active at the top right of the interface. Click the link to access the reporting interface on the CA Business Intelligence JasperReports Server.

NOTE
If the Administration tab changes are not saved or the fields clear when you click Save or Test, adjust your browser settings. One of the following actions should solve the problem:

- Add the CA SOI URL to your trusted sites and lower the privacy to accept all cookies. For more information about trusted sites and privacy settings, see your browser documentation.
- Add the domain name of the CA SOI UI Server and SA Manager (for example, company.com) to the privacy managed websites and set to Allow. For more information about managing privacy settings, see your browser documentation.

When you click the Save & Test button in the Report Configuration page, CA SOI checks the provided report URL, but not the credentials. So, provide the correct credentials when you configure the Reports link on the Dashboard.

Validate CA SOI Reports Configuration

Follow these steps to validate the CABI JasperReports Server and CA SOI Reports installations and their configurations.

1. Open CA SOI UI and click the Reports link.
   CABI JasperReports Server log in page appears.

   NOTE

   If the Reports link is not available or the CABI JasperReports Server pages does not appear, check the configuration procedure.
2. Log in to CABI JasperReports Server as the user with administrator privileges who does not belong to an organization in CABI JasperReports Server. **Default:** superuser/superuser.
3. Click Manage, Organizations.
   soi appears in the Organization list on the right pane.
4. Click Manage, Roles.
   List of existing roles appear and the role corresponding to the soi organization appear.
5. Click View, Repository.
   The folders in the repository appear.
6. Expand the root, public, ca, Service Operations Insight, datasources node in the capability node.
7. Select soi ds in the right pane and click Edit.

   NOTE

   When using the Dynamic option, the installer does not support JDBC URL with the instance name. To view the reports, enter the JDBC URL that points to the default port to a JDBC URL with the instance name.
8. Update the JDBC URL with the following format in the URL text field.

   ```
   jdbc:tibcosoftware:sqlserver://{hostname}\{SQL instance
   name};databaseName=SAMStore
   ```

   For example, if the jdbc url is *jdbc:tibcosoftware:sqlserver://{hostname}:1433;databaseName=SAMStore*, then update the url with *jdbc:tibcosoftware:sqlserver://{hostname}\{SQL instance name};databaseName=SAMStore*
9. Expand root, public, ca, Service Operations Insight, reports.
   List of reports available for the administrator user appears. The drill-down reports node appears only for the administrator.
10. Click a report title node. The detailed list appears on the right pane.
11. Click on the report title, provide Input Controls, and click OK to generate report.

Enable CA SOI Reports for SSO Users

Prerequisites

Before you enable the CA SOI Reports for the SSO Users, perform the following prerequisites:

- Install the latest SOI patch.
- Install CABI and keep it running.
- Install the Report Installer on CABI Server.

Follow these steps to enable CA SOI Reports for the SSO Users.

On SOI UI Server

1. Create a user on EEM Server.
2. Add the user into SOI using Console.

On Jasper Server

1. Log in to the JasperSoft Server as "*superuser*".
2. Verify if the newly created SOI user is added into CABI with default role: **ROLE_USER**.
   Assign the **ROLE_SOI_REPORTS** to the user.
3. Execute the `jkscreation.bat` in the file directory `C:\CA\SOI\reports\deploy` to create the jks file and add the password in the properties file.
   The following files are created in the directory:
   `C:\CA\SOI\reports\deploy\keystore\soi.jks`
   `C:\CA\SOI\reports\deploy\keystore\soi.properties`
4. Copy jks and property files in the file directory.
   `C:\Program Files\CA\SC\CA Business Intelligence\apache-tomcat\webapps\jasperserver-pro\WEB-INF\config\`
5. Uncomment the following line in the file `catalina.properties` in the location: `C:\Program Files\CA\SharedComponents\CA Business Intelligence\apache-tomcat\conf\`.
   `tomcat.util.http.parser.HttpParser.requestTargetAllow=|`
6. Restart CABI Tomcat Service on the services console.

On SOI UI Server

1. Copy the jks and the properties files in the location: `C:\Program Files (x86)\CA\SOI\SSO\`
   `C:\Program Files (x86)\CA\SOI\SSO\soi.jks`
   `C:\Program Files (x86)\CA\SOI\SSO\soi.properties`
2. Create an environment variable with the name `CA_TRUSTEDAUTH_HOME` and set the path of the jks and the properties file location.
3. Restart SOI UI Service

The newly created user can now access the SOI Reports from the account.

Configure Help Desk Integration

Contents

As an administrator, you can configure CA SOI to communicate with CA Service Desk, BMC Remedy IT Service Management Suite, HP Service Manager, ServiceNow, or a custom help desk application.

After you complete this configuration, CA SOI can create help desk tickets that are automatically based on alert escalation policies and their associated actions.

For help desk product integration procedures, see Help Desk Integrations .

Follow these steps:

1. Click the Administration tab.
2. Click the plus sign (+) next to CA SOI Insight Manager Configuration.
3. Click the plus sign (+) next to the server you want to configure.
4. Click Help Desk Configuration.
5. Select or enter the appropriate information in the following fields:
   - **Help Desk Type**
     Specifies the type and version of help desk software you want to integrate with CA SOI.
   - **Server**
     Specifies the name of the help desk host server.
     Note: Install the Service Desk Web Services component on the CA Service Desk server for the integration with CA Service Desk to work.
   - **Server Port**

Specifies the port number that the help desk host server uses.

- **Launch in Context Port**
  (CA Service Desk Only) Specifies the port number that is configured on the help desk host server to launch the ticket detail page.
- **User**
  Specifies the user account with which to access the help desk.
- **Password**
  Specifies the password that corresponds with the help desk user account.
- **SSL**
  Specifies whether to use SSL to communicate with the selected help desk application.

  NOTE
  Import an SSL certificate from the help desk application or (if you selected BMC Remedy or HP Server Manager) into the CA SOI trust store for the SSL connection to work. For more information about importing the SSL certificate, see How to Configure a BMC Remedy Integration.

6. (Optional and available only if CA Service Desk integration with Polling is off and SDM Notifications ON) Click the button that is available:
   - **Resync**
     Synchronizes the CA Service Desk Manager ticket statuses. Use when the Service Desk Manager server is restarted to ensure that the ticket statuses are synchronized.
   - **Refresh Status**
     Refreshes and displays the status of the Resync button. This button is available only while a resync is in progress.
7. Click Test to verify that CA SOI can connect to the selected help desk application.
   A message confirms the connection.
8. Click Save.
   CA SOI is configured to communicate with the specified help desk application.
   The updated configuration file is in the <SOI_HOME>\tomcat\custom\help-desks.xml folder on the SA Manager server.
9. Restart the CA SAM Application Manager service and verify that all changes function correctly.

   NOTE
   If the Administration tab changes do not save or the fields clear when you click Save or Test, adjust your browser settings. One of the following actions should solve the problem:

- Add the CA SOI URL to your trusted sites and lower the privacy to accept all cookies. For more information about trusted sites and privacy settings, see your browser documentation.
- Add the domain name of the CA SOI UI Server and SA Manager (for example, company.com) to the privacy managed websites and set to Allow. For more information about managing privacy settings, see your browser documentation.

Edit Help Desk Configuration

After you configure integration with a help desk product, you can configure global preferences for a help desk configuration. You can also create and maintain custom help desk and announcement properties. For more information about integrating CA SOI with help desk products, see Help Desk Integrations.

Follow these steps:

1. Select Tools, Help Desk Configuration in the Operations Console.
   If the Help Desk Status entry is Not Configured, then the connection to a help desk is not configured correctly in the Administration UI. You can still set the help desk configuration in this dialog, but it does not go into effect until a valid help desk connection is established.
2. Select the type of help desk in the Help Desk Type drop-down list.

   NOTE
   If you select BMC Remedy or HP Service Manager, the General tab becomes unavailable. You configure these options for BMC Remedy or HP Service Manager in the CA Process Automation form used to

configured BMC Remedy or HP Service Manager server settings. For more information, see Help Desk Integrations.

3. Set the following properties in the General tab and click Save:
   - **Auto clear alert**
     Specifies whether to clear the alert automatically in CA SOI when the corresponding help desk ticket changes to the selected status in the help desk.
   - **Auto change trouble ticket status when alert is cleared**
     Specifies whether to change the help desk ticket status automatically to a specified value when the corresponding alert in CA SOI clears.
   - **Enable Polling**
     Specifies whether to poll the help desk application for synchronization. If you select this option, specify a polling interval in the Polling interval for synchronization field. Enable polling if you selected to automatically clear an alert based on ticket status, and the help desk product cannot perform activity notifications triggered by certain user actions and call external binaries as a result of these actions. If this functionality is absent in the help desk, CA SOI must poll the help desk to query for ticket closure.

   The help desk configuration is saved.
4. (Optional) Click the Ticket Properties or Announcement Properties tab and add a custom ticket or announcement properties.
   On this tab, you can also edit or delete existing custom tickets or announcement properties.

Add Custom Ticket or Announcement Properties

You can add custom ticket or announcement properties. You make the properties available for the create ticket or announcement actions.

> **NOTE**
> If you added custom ticket properties using other methods in previous releases (such as the SD-ticketProps.xml file), migrate the custom properties to the Help Desk Configuration dialog.

Video: Add Custom Help Desk Ticket Properties

Follow these steps:

1. Click the Ticket Properties or Announcement Properties tab on the Help Desk Configuration dialog.
   The current list of available ticket or announcement properties appears.
2. Click

   .
3. Complete the following fields and settings:
   - **Property Name**
     Defines the name of the property in the help desk.
   - **Property Label**
     Defines the property name to display on the Create Action dialog.
   - **Value Type**
     Specifies whether the value for the custom property should be selectable from a drop-down list or definable in a text field.
   - **Required Property**
     Specifies whether the property should be required in all ticket or announcement actions.
   - **Show Create Option**
     Specifies whether to display the 'Create Object if not present' check box for the property to provide the option to create the object in the help desk if it does not currently exist.

4. (Drop-down value type only) Define the values to make selectable for property assignment in the 'Drop down values' pane as follows:
   - Select potential values from the 'Available property values' list and move them to the 'Allowable values for this property' list. Any of the expandable runtime tokens take the specified value from the alert to populate the property. Both CA SOI and USM alert properties are available as expandable runtime tokens. The list contains CA SOI properties; click More to view the full list of available USM and CA SOI properties.
   - Click New to add custom potential values to the 'Allowable values for this property' list.
5. Complete the following fields and click OK:
   - **Default Value**
     (Optional) Defines the default value that is displayed when adding the property to an escalation action. This property must exist in the 'Allowable values for this property' list if the value type is a drop-down list.
   - **Data Type Mapping**
     (Optional) Defines the object type to use for ticket creation that is based on the custom property value. Available values are ASSET, CONTACT, GROUP, STRING, and TEMPLATE. For example, if you create an EmergencyContact custom property and map it to the CONTACT data type, the help desk looks up any CONTACT with the matching value for this property and uses that object for ticket creation.
   - **Hint Text**
     (Optional) Defines the text that appears when you click the Hints link next to the property when adding it to an escalation action.

   The property appears in the Current List of Properties pane in the Ticket Properties or Announcement Properties tab.

Value Mapping

The values for priority and severity are different in CA SOI and CA Service Desk. The following tables show how the values equate to each other. The value mapping happens automatically when the expandable runtime tokens are used. You are responsible to implement the mappings when setting custom property values.

Priority

The priority value from CA SOI is the maximum priority of all of the services that the alert impacts. The service priority value is set for each service. The value is located in the Operations Console on the Information tab of the Component Detail pane. CA Service Desk calculates priority based on the values of Urgency and Impact. For more information, see the CA Service Desk documentation.

Impact

The impact value from CA SOI is the maximum impact of all the services that the alert impacts. CA SOI sets the impact value and it is located in the Operations Console on the Alert Details tab of the Component Detail pane.

The CA Service Desk impact is the greatest with lower numbers.

CA SOI value (Low to High)	CA Service Desk value (Low to High)
1 (Slight)	5 (One Person)
2 (Moderate)	3 (Single Group)
3 (Severe)	2 (Multi Group)
4 (Down)	1 (Entire Org)

Severity

The severity value from CA SOI is the severity of the alert. This value is located in the Operations Console on the Alert Details tab of the Component Detail pane.

CA SOI value (Low to High)	CA Service Desk value (Low to High)
1 (Minor)	1 (Escalation)
2 (Major)	3 (Manager Escalation)
3 (Critical)	4 (HD Mgr. Escalation)
4 (Down)	5 (All Hands Escalation)

Configure Email and Failure Notifications

Contents

As an administrator, you configure email for CA SOI notifications, including failure messages.

Configure Email

As an administrator, if a *mailhost* DNS alias for the email server does not exist, you provide a server name so that CA SOI can send email notifications. Mailhost is a common DNS alias for the email server, and is the default setting in CA SOI. These settings are also used to update settings in Registry used by CA Catalyst components. CA Catalyst sends a notification email if there is an error during CI reconciliation or synchronization.

CA SOI uses the following process to determine the FROM address:

1. ENOTIFY_EMAIL_FROM_ADDR value from the server's JSW configuration file
2. The Error Notification FROM Email field on the Error Notification Configuration page
3. Administrator user email address from CA EEM; however, samuser does not provide an email
4. The Administrator Email field on the Error Notification Configuration page
5. SYSTEM@server_name, which is the hardcoded default

When you define outbound email, use the FROM address that helps identify the source or originating component:

- SOI_MGR_NoReply: SA Manager originated emails
- SOI_UI_NoReply: UI Server originated emails
- SOI_Error_NoReply: SA Manager Error Notification
- SOI_Action_NoReply: SA Manager Notification Action
- SOI_JSW_NoReply: SA Manager JSW originated internal errors
- SOI_BOXI_NoReply: Reporting

Follow these steps:

1. Click the Administration tab.
2. Click the plus sign (+) next to the CA Service Operations Insight Manager Configuration or CA Service Operations Insight UI Server Configuration option.
3. Click the plus sign (+) next to the server you want to configure.
4. Click the Email Configuration option.
5. Complete the fields and click Save.
6. Depending on the server you configured, restart the CA SAM Application Server service or the CA SAM User Interface Server service.

WARNING
Set or update the email configuration settings for both the SA Manager and UI Server.

NOTE

If the Administration tab changes do not save or the fields clear when you click Save or Test, adjust your browser settings. One of the following actions should solve the problem:

- Add the CA SOI URL to your trusted sites and lower the privacy to accept all cookies. For more information about trusted sites and privacy settings, see your browser documentation.
- Add the domain name of the CA SOI UI Server and SA Manager (for example, company.com) to the privacy managed websites and set to Allow. For more information about managing privacy settings, see your browser documentation.

Configure an SMTP Server

Delivery of a message is initiated by transferring the message to a designated Simple Mail Transfer Protocol (SMTP) server. The SMTP service is used in delivering e-mail messages. If the mail server uses HTTPS or the SMTP server is password protected, configure the SMTP server.

Follow these steps:

1. Copy the certificate file from SMTP mail Server.
2. Backup the <SOI_HOME>\tomcat\conf\ssa.jks file.
3. Execute the following command from a command prompt on the SA Manager system:

```
<JAVA_HOME>\bin\keytool.exe" -importcert -file <DIR>\<Certificate> -keystore
 "<SOI_HOME>\tomcat\conf\ssa.jks" -alias "<ALIAS_NAME>
```

   where, <*DIR*> : Defines the path to the directory where you copied the certificate file; use storepass as catalyst
4. Restart the CA SAM Application Server service on the SA Manager.

Configure Failure Email Notifications

As an administrator, you can configure email notifications to a specified administrator or administrators when certain failures occur. CA SOI also provides logging for the failures:

- The action mechanism fails due to a third-party server connection error. The action mechanism relies on a connection to external servers for the help desk (CA Service Desk), workflows (CA Process Automation), and so on. CA SOI now notifies a specified administrator when the connectivity is lost for more than a specified number of minutes.
- The SA Manager fails due to looping errors. For more information, see Loop Detection during Impact Analysis.
- A connection to the SA Store database fails.
- A connector status changes or fails. For more information, see Connector Shutdown Notifications.
- After a configured retry duration period, an escalation policy action fails.

The email notifications provide a description of the problem and troubleshooting tips to resolve the problem.

For any failure, CA SOI updates the soimgr.log file with failure information.

Follow these steps:

1. Click the Administration tab.
2. Click the plus sign (+) next to CA SOI Insight Manager Configuration.
3. Click the plus sign (+) next to the server you want to configure.
4. Click Error Notification Configuration.
5. Enter the full email address for the administrator.

NOTE
Enter multiple email addresses separated with commas (,).

6. For each notification section, perform the following actions:
   a. Select Yes from the drop-down to turn on email notifications.
   b. Enter the number of minutes until CA SOI sends the email notification.

NOTE
The minimum value is 1 minute.

Loop Detection During Impact Analysis

CA SOI detects loops during an impact analysis. An administrator can configure CA SOI to send an email notification when the SA Manager fails due to looping.

When CA SOI detects a loop in a service or service hierarchy, the SA Manager marks that service as being in TEST mode. This blocks the service state propagation and, in effect, takes the service offline. If the email notification is enabled for the SA Manager events, then CA SOI generates an email. The email contains the name of the service that CA SOI detected the loop in and the SA Manager that CA SOI detected the loop on. To return the service to a production state, edit and then save the service.

When CA SOI detects a loop during an impact analysis, CA SOI appends the following message with the name and internal model handle to the soimgr.log file. CA SOI appends a message each time that CA SOI detects looping.

```
*******************************************************************************
*******************************************************************************
*******************************************************************************
ATTENTION! Loop detected in service 'XYZ'?Setting all service properties for MH(0x10000000003) to TEST
*******************************************************************************
*******************************************************************************
*******************************************************************************
```

Connector Shutdown Notifications

CA SOI provides notifications and detailed logging when a connector shuts down. CA SOI logs and sends an immediate notification to the administrator in the event any connector goes offline for any reason. You can set an interval where CA SOI consolidates all failed connector information into one notification. The improved and faster notification mechanism provides comprehensive information about the connector shutdown behavior, including any appropriate reason for the shutdown. The information in the log messages helps you troubleshoot connectors, audit the connector status in your infrastructure, and take any prompt actions. You can review the related log files to find more information about any anomaly that you encounter in the connector behavior. The easy identification of the problem that is associated with the connector also lets you manage your domain managers more efficiently.

The enhanced connector notification and logging mechanism, therefore, helps you as follows:

- Logs the appropriate reason (for example,connector failure or an IFW shutdown) for the connector shutdown in the log file, SAM-IntegrationServices_wrapper.log. You can review the log file to locate and analyze the message information.
- Includes the shutdown message in the connector status notification as part of the *heartbeat* message. The heartbeat message is logged into the <*ConnectorName*>_HEARTBEAT_PUB.txt. You can review this file to see the message included in the statusDesc property.
- Displays the reason for the connector shutdown in the Administration UI and Console.
- Sends an email notification to the CA SOI administrator about the connector shutdown.

Configure Global Settings

As an administrator, you configure global settings that determine various default behaviors:

- maintenance mode impact and domain manager maintenance settings propagation
- alert clearing and unknown alerts
- root cause analysis source
- service model granularity and automatic policy maintenance
- escalation policy/actions retry behavior
- projection sheets maximum number

Follow these steps:

1. Click the Administration tab.
2. Click the plus sign (+) next to CA SOI Insight Manager Configuration.
3. Click the plus sign (+) next to the server you want to configure.
4. Click Global Settings.
5. Make the appropriate selections from the drop-down lists and fields.
   The Global Settings page provides detailed information about each field; however, the following fields require additional information:
   - **Maintenance Mode Settings**
     - **Propagate Maintenance Impact**
       Note: Changing a Global Settings flag while the SA Manager is running only impacts future alertsit does not affect existing alerts and associated services.
     - **Propagate Domain Manager Maintenance Settings**
       The maintenance setting from the SA Manager can also propagate to other domain managers, depending on your configured reconciliation and synchronization formulas.
   - **Root Cause Analysis Setting**
     Controls the source for root cause analysis. Select CA SOI, Combination, or Domain Manager.
     Consider the following:
     - If you select Combination, CA SOI uses both itself and the domain manager to determine the root cause.
     - The Domain Manager and Combination modes require that the domain manager and its connector both support sending the root cause analysis information to CA SOI. For current root cause analysis support see the product-specific *Connector Guide* that is provided with each connector.

       NOTE
       For more information about working with the root cause analysis mode, see Set Root Cause Analysis Mode.
   - **Self Monitoring Setting**
     Enables self monitoring capabilities for connector status. When a connector is no longer available, CA SOI generates an alert associated with the CA Service Operations Insight Application CI. Like any service, the CI status propagates to the service level, and the impact of the connector failure on the CA Service Operations Insight service changes accordingly. You can perform any actions on these alerts, including acknowledge, assign, and so on. Use the self monitoring capability to take proactive measures in response to service degradation, as you would for any CA SOI service. In this case, you can create escalation policies to take specific actions when a connector goes down, such as creating a help desk ticket or using a script to query the source domain manager to detect potential issues with the underlying application.

     Select Yes to enable self monitoring.

     Turning off the Self Monitoring feature after you have previously enabled it stops further alerts on the Application CI. However, it does not remove the CA Service Operations Insight service model from the topology or clear existing alerts on the Application CI. You have to manually remove the service model and clear existing alerts.

NOTE

Removing the Universal connector from the Administration UI removes the Application CI in the CA Service Operations Insight service model. A running Universal Connector is required for self monitoring. Re-install the Universal connector and restart the CA SOI Application Server service to re-create the CI.

– **Modeler Settings**
  - **Default Service Model Granularity**
    Select Normal to use a normal granularity service model, which employs explicit modeling and does not aggregate alerts from child CIs not included in the service model.
    Select Low to use low granularity so the service model employs implicit modeling and aggregates alerts from child CIs not included in the model.
  - **Low Granularity Impact**
    The Low Granularity Impact parameter is applicable only if you select Low in Default Service Model Granularity.
    Select Normal to calculate the impact by multiplying relationship significance of the modeled CIs or sub CIs and significance of the parent CI for unmodeled sub CIs.
    Select Advanced to calculate the impact by multiplying the relationship significance of the modeled CIs or sub CIs and default ‘class significance’ for the unmodeled sub CIs.

– **Maximum Number of Projection Sheets per Notebook for a Connector Setting**
  When a connector imports CIs into CA SOI, CA SOI can correlate the CIs to existing CIs if their correlation keys match. Typically, you expect CA SOI to correlate a CI from one connector to a CI from another connector. However, there may also be duplicate CIs from the same connector matching and being correlated to the same CI on CA SOI.
  When CA SOI imports duplicate CIs from the same connector, CA SOI can create duplicate projection sheets for each CI in the notebook. You can limit the number of projection sheets CA SOI creates by adjusting this setting. For more information about projection sheets and notebooks, seeReconciliation Concepts. The suggested value of 1 indicates that CA SOI creates one projection sheet for each unique CI (from the same connector) in the notebook.
  CA SOI updates soimgr-debug.log with a warning about duplicated CIs from the same connector.
  CA SOI updates the SOI_HOME\tomcat\logs\ci-invalid.log file with the USM attributes for the duplicate CIs. Even if you set the maximum number of projection sheets higher than one, CA SOI updates the log file any time the number of sheets is greater than one.

– **Action Progress Bar**
  Use this setting to define the visibility of the progress bar when you take a policy action on multiple alerts. If you select True, the progress bar appears to indicate the status of the action taken on alerts and you cannot perform further action till a ticket ID is assigned to the selected alerts. If you select False, the progress bar does not appear and the action on alerts is taken in the background.

6. Click Save.
7. Restart the CA SOI Application Server service for the change to take effect.

NOTE
If the Administration tab changes do not save or the fields clear when you click Save or Test, adjust your browser settings. One of the following actions should solve the problem:

- Add the CA SOI URL to your trusted sites and lower the privacy to accept all cookies. For more information about trusted sites and privacy settings, see your browser documentation.
- Add the domain name of the CA SOI UI Server and SA Manager (for example, company.com) to the privacy managed websites and set to Allow. For more information about managing privacy settings, see your browser documentation.

Configure Auditing Levels

As an administrator, you change or disable the auditing level for various CA SOI components such as CIs, policies, and events. The audit records are stored in the CA SOI database.

Follow these steps:

1. Click the Administration tab.
2. Click the plus sign (+) next to CA SOI Insight Manager Configuration.
3. Click the plus sign (+) next to the server you want to configure.
4. Click Audit Configuration.
5. You can perform the following actions on this page:
   – Select a default audit level setting for all components or disable auditing entirely.
   – Override the default settings by selecting the auditing options in any section.

   NOTE
   The Generic Type audit is for customized implementations. For more information, contact CA Support.
6. Click Save.

   NOTE

   If the Administration tab changes do not save or the fields clear when you click Save or Test, adjust your browser settings. One of the following actions should solve the problem:

   - Add the CA SOI URL to your trusted sites and lower the privacy to accept all cookies. For more information about trusted sites and privacy settings, see your browser documentation.
   - Add the domain name of the CA SOI UI Server and SA Manager (for example, company.com) to the privacy managed websites and set to Allow. For more information about managing privacy settings, see your browser documentation.

CA Process Automation Integration

Contents

You can integrate with CA Process Automation to include CA SOI as a part of automated CA Process Automation workflows. Alert escalation policy can trigger CA Process Automation processes to automate complex responses to alert conditions. For example, a high CPU utilization alert on a virtual machine could trigger a process in CA Process Automation. The trigger provisions a new virtual machine in CA Server Automation to offset the virtual machine processing load.

Configure CA Process Automation Integration

As an administrator, you configure a connection with a CA Process Automation installation to invoke CA Process Automation Forms and Processes in an alert escalation action. For example, when an alert occurs specifying that a system is down, you can configure escalation policy to invoke a CA Process Automation Form or Process that restarts the system.

Follow these steps:

1. Click the Administration tab.
2. Click the plus sign (+) next to CA SOI Insight Manager Configuration.
3. Click the plus sign (+) next to the server you want to configure.
4. Click Process Automation Server Configuration.
5. Complete the fields and click Save.

   NOTE
   for the SSL connection to work. For more information about importing the CA Process Automation certificate, see How to Configure a BMC Remedy Integration.
6. (Optional) Click Launch Process Automation Web Admin.

7. Log in and create forms for any processes that you want to run as an alert escalation action. Processes must have an associated form to include in escalation actions.

   NOTE
   For more information about defining CA Process Automation Forms for Processes, see the CA Process Automation documentation.

   NOTE

   If the Administration tab changes do not save or the fields clear when you click Save or Test, adjust your browser settings. One of the following actions should solve the problem:

   - Add the CA SOI URL to your trusted sites and lower the privacy to accept all cookies. For more information about trusted sites and privacy settings, see your browser documentation.
   - Add the domain name of the CA SOI UI Server and SA Manager (for example, company.com) to the privacy managed websites and set to Allow. For more information about managing privacy settings, see your browser documentation.

Configure SSL Connection with CA Process Automation

For CA SOI to communicate with a CA Process Automation server that has been configured to use SSL, you must import a certificate into the CA SOI trust store.

Follow these steps:

1. Copy the itpamcertificate.cer file from the following location on the CA Process Automation server to a directory on your SA Manager server:
   PA_HOME\ITPAM\server\c20\.c20repository
2. Make a backup copy of the SOI_HOME\tomcat\conf\ssa.jks file.
3. Run the following command from a command prompt on the SA Manager system to import the certificate into CA SOI:

   ```
   "JAVA_HOME\bin\keytool.exe" -v -importcert -storepass password -file DIR\itpamcertificate.cer -keystore
   "SOI_HOME\tomcat\conf\ssa.jks" -trustcacerts -noprompt
   ```

   – **password**
     Defines the password of the CA Process Automation server to login as administrator.
     .
   – **DIR**
     Defines the path to the directory to which you copied the itpamcertificate.cer file.
4. Restart the CA SAM Application Server service.
5. Configure CA Process Automation integration in the Administration tab on the Dashboard. Select the SSL check box and use the SSL port number.
6. Click Test.

Configure Google Maps Integration

As an administrator, you can configure CA SOI with Google Maps. You can view the services by clicking the Maps View on the Dashboard. You ca view normal and degraded services that are based on their defined location. To configure the Google Maps integration, perform the following action:

- Set the Location property for each service in the Operations Console on the service Information tab. The Location must be a valid address or location. Entries that are too generic or specific may not appear in the appropriate location in Google Maps.

For more information about viewing service information, see View Services in Google Maps.

Configure JNLP

As an administrator, you can change the JRE requirements for starting the Operations Console. You can edit any of the default Java Network Launching Protocol (JNLP) settings.

The default memory amount that CA SOI allocates to the Operations Console deployment is 512 MB. This setting may not be sufficient in larger environments where CA SOI manages many services and CIs are staged.

CA SOI notifies you with a message when the Operations Console is using 80 percent (410 MB). CA SOI displays subsequent messages on the status bar.

CA SOI does not force the Operations Console to have the same JRE version as the UI Server. You can upgrade the UI Server to a newer JRE version and upgrade the user desktops later. For more information about JRE version for UI Server and Operation Console, see Software Requirements.

If a user does not meet the minimum JRE requirement, the Console link is not available on the Dashboard and a message appears that explains the problem.

Follow these steps:

1. Click the Administration tab.
2. Click the plus sign (+) next to the CA Service Operations Insight UI Server Configuration option.
3. Click the plus sign (+) next to the server you want to configure.
4. Click JNLP Configuration.
5. Edit the settings in one or more of the following fields and click Save:
   - **Required Minimum JRE Version**
     Displays the oldest version of the Java Runtime Environment (JRE) that can be used to start CA SOI.
     Default: AdoptOpenJDK JRE 8.0.212
   - **Minimum client memory usage (megabytes)**
     Displays the minimum amount of memory allocated for starting CA SOI.
     Default: 64
   - **Maximum client memory usage (megabytes)**
     Displays the maximum amount of memory allocated for starting CA SOI.
     Default: 512
     Note: The suggested value for large implementations is 1024 MB.

   The JNLP configuration settings update in the custom-jnlp-config.xml file.

   NOTE

   If the Administration tab changes do not save or the fields clear when you click Save or Test, adjust your browser settings. One of the following actions should solve the problem:

   - Add the CA SOI URL to your trusted sites and lower the privacy to accept all cookies. For more information about trusted sites and privacy settings, see your browser documentation.
   - Add the domain name of the CA SOI UI Server and SA Manager (for example, company.com) to the privacy managed websites and set to Allow. For more information about managing privacy settings, see your browser documentation.
   - When the Operation Console connects to the server for the first time, an authentication window appears. To disable the authentication window in JNLP file, set the value to **true** as follows:
     property name="javaws.cfg.jauthenticator" value="true".

Configure Metric Definition

As an administrator, you configure the service health definition to determine the health level that determines when a service is down.

Follow these steps:

1. Click the Administration tab.
2. Click the plus sign (+) next to the CA Service Operations Insight UI Server Configuration option.
3. Click Metric Definition Configuration.
4. Select a health level that determines when a service is down from the drop-down list and click Save.

NOTE

If the Administration tab changes do not save or the fields clear when you click Save or Test, adjust your browser settings. One of the following actions should solve the problem:

- Add the CA SOI URL to your trusted sites and lower the privacy to accept all cookies. For more information about trusted sites and privacy settings, see your browser documentation.
- Add the domain name of the CA SOI UI Server and SA Manager (for example, company.com) to the privacy managed websites and set to Allow. For more information about managing privacy settings, see your browser documentation.

Configure Mobile Dashboard Integration

As an administrator, if you want to launch the Mobile Dashboard from the Administration tab and to enable use of the $[Mobile UI URL] runtime token, you configure the Mobile Dashboard connection information.

Follow these steps:

1. Click the Administration tab.
2. Click the plus sign (+) next to CA SOI Insight Manager Configuration.
3. Click the plus sign (+) next to the server you want to configure.
4. Click Mobile Dashboard Server Configuration.
5. Complete the server and port fields.

NOTE
Unless you performed a custom installation, the Mobile Dashboard resides on the system where you installed the UI Server.

6. (Optional) Click Launch Mobile Dashboard to test that the URL launches successfully.
7. Click Save.
8. Restart the CA SAM Application Server service for the change to take effect.

NOTE

If the Administration tab changes do not save or the fields clear when you click Save or Test, adjust your browser settings. One of the following actions should solve the problem:

- Add the CA SOI URL to your trusted sites and lower the privacy to accept all cookies. For more information about trusted sites and privacy settings, see your browser documentation.
- Add the domain name of the CA SOI UI Server and SA Manager (for example, company.com) to the privacy managed websites and set to Allow. For more information about managing privacy settings, see your browser documentation.

Configure Synchronization

As an administrator, you use the Synchronization Configuration page to enable or disable the following synchronization use cases:

- Synchronize the Cleared and Acknowledged alert properties in CA SOI - CA Spectrum and CA SOI - Microsoft SCOM
- Synchronize services, CI types, and relationships in CA SOI and BMC Atrium or CA CMDB
- Synchronize maintenance mode status with connectors that support inbound to connector operations on the IsInMaintenance USM property, and set the reconciliation formulas for the IsInMaintenance property

Follow these steps:

1. Click the Administration tab.
2. Click the plus sign (+) next to CA SOI Insight Manager Configuration.
3. Click the plus sign (+) next to the server you want to configure.
4. Click Synchronization Configuration.
5. Enable or disable the component synchronization for alerts, services, or maintenance mode.
6. Click Save.

NOTE

If the Administration tab changes do not save or the fields clear when you click Save or Test, adjust your browser settings. One of the following actions should solve the problem:

- Add the CA SOI URL to your trusted sites and lower the privacy to accept all cookies. For more information about trusted sites and privacy settings, see your browser documentation.
- Add the domain name of the CA SOI UI Server and SA Manager (for example, company.com) to the privacy managed websites and set to Allow. For more information about managing privacy settings, see your browser documentation.

Configure USM Web View Integration

As an administrator you configure the USM Web View URL setting, which enables the USM Web View link on the Dashboard. The URL also enables use of the $[USM Web View URL] runtime token to invoke USM Web View as part of an escalation action. For information about using USM Web View, see USM Web View for PC and USM Web View for Mobile Devices.

Follow these steps:

1. Click the Administration tab.
2. Click the plus sign (+) next to CA SOI UI Server Configuration.
3. Click the plus sign (+) next to the server you want to configure.
4. Click USM Web View Configuration.
5. Complete the server and port fields. Unless you performed a custom installation, USM Web View resides on the system where you installed the UI Server.
6. Click Save.
7. Restart the CA SAM Application Server service for the change to take effect.

View Client Details

Contents

As an administrator, you can view the Client Details page that lists the users that are currently logged in to the Operations Console. From this page you can also log off users from the Operations Console and send a message to a user.

Follow these steps:

1. Click the Administration tab.
2. Click the plus sign (+) next to the CA Service Operations Insight UI Server Configuration option.
3. Click the plus sign (+) next to the UI Server whose client details you want to view.

4. Click Client Details.
   A table of all clients currently logged in to the Operations Console displays.

   NOTE
   You can only perform this procedure on the UI Server. The SA Manager has a similar feature where you can view details about the UI Servers that are connected to it.

Send a Message to a User

You can send a message similar to an email message to any user who is logged in to the Operations Console.

Follow these steps:

1. Click the Administration tab.
2. Click the plus sign (+) next to the CA Service Operations Insight Manager Configuration or CA Service Operations Insight UI Server Configuration option.
3. Click the plus sign (+) next to the server you want to configure.
4. Click Client Details.
5. Click the check box next to one or more users to whom you want to send a message, and then click Send Message.
6. Enter the message and click Send.

Set Client Debug Status

You can enable or disable tracing of HTTP activity for each client session for troubleshooting purposes.

Follow these steps:

1. Click the Administration tab.
2. Click the plus sign (+) next to the CA Service Operations Insight UI Server Configuration option.
3. Click Client Details.
   The Debug column shows whether debugging is enabled or disabled for each client session.
4. Click the link in the Debug column to toggle enable and disable for any user.

Log Off Clients

As an administrator, you can log off other clients when you perform maintenance on the SA Manager or UI Server, or to upgrade the software.

Follow these steps:

1. Click the Administration tab.
2. Click the plus sign (+) next to the CA Service Operations Insight UI Server Configuration option.
3. Click the plus sign (+) next to the server you want to configure.
4. Click Client Details.
5. Click the check box next to one or more user names and click Log Off Clients, then confirm the log off.

Open the Client Debug Console

WARNING
The Debugging Console is designed to be used only with help from CA Technical Support.

For more information about using the debug pages, see Debug Consoles.

Follow these steps:

1. Click the Administration tab.
2. Click the plus sign (+) next to the CA Service Operations Insight UI Server option.

3. Click the name of the server you want to debug.
4. Click Debugging And Logs.

Access the SOI Console

You can save the JNLP file at an easily accessible location to access the SOI Console.

Follow these steps:

1. Open SOI UI on an internet browser.
2. Click the Console link.
3. Perform the following steps on the Microsoft Internet Explorer.
   a. Select **Open**, when prompted to Open or Save the JNLP file.
   b. Select javasw file from `<JAVA_HOME>\jdk-8.0.222.10-hotspot\bin` directory when prompted. The SOI console opens when you click Console in the SOI UI.
4. Perform the following steps on Google Chrome.
   a. Select Keep, when prompted to Keep or Discard the JNLP file.
   b. Select javaws file from `<JAVA_HOME>\jdk-8.0.222.10-hotspot\bin` directory when prompted. The SOI console opens when you click Console in the SOI UI.

Connector Administration

This section describes how administrators manage installed connectors.

Connector Configuration Tasks

Contents

As an administrator, you can view a connector status, edit a connector configuration including the CMDB connector, configure the IFW configuration file, and change the connector properties.

For more information about working with connectors, see Managing CA Catalyst r3.2 Connectors and Connectors.

View Connector Status

As an administrator, you can view the Connector Status table, which lists all the connectors available to CA SOI. From this table, you can monitor the IFW and connector status.

Follow these steps:

1. Click the Administration tab, and select Connector Configuration.
   The Connector Status table displays with the following columns:
   – **Connector Service**
     Displays the name of the server where the IFW is installed. The IFW exists on every system that has at least one connector installed, and it connects to the domain manager that CA SOI is monitoring.
   – **Status**
     Displays one of the following states for the IFW on the connector server:
     - **Online/Offline**
       Indicates that the IFW is running or not running.
     - **Initializing**
       Indicates that the IFW is starting up.
     - **Handshake**

Indicates that the IFW was started and is waiting for CA SOI to contact it.
  - **Closing**
    Indicates that the IFW is shutting down.
- **Connector**
  Displays the connector identifier using the following format: *connector_mdrproductvalue@product_host*. The connector runs on its installed system, communicates with the individual domain manager, collects data from it, and lets the IFW process this data. The connector identifier typically includes a unique ID number for each connector and the system on which the integrated domain manager is installed.
- **Status**
  Displays one of the following states:
  - **Online\Offline**
    Indicates that the connector is running or is not running.
  - **Initializing**
    Indicates that the connector is starting. This state is short-lived during which connectors initialize their connections to domain managers and build their caches before switching to the Online status.
  - **Handshake**
    Indicates that the connector was started and is waiting for CA SOI to contact it.
  - **Closing**
    Indicates that the connector is shutting down.
- **Source Description**
  Contains information about the source domain manager with which the connector integrates, specifically on which system the domain manager is running.
- **Connector Description**
  Contains the connector identifier and version number.

2. Click one of the server names in the Connector Service column.
   The read-only details page opens and displays the following details about the IFW on each connector server:

   NOTE
   The tree view in the Administration Pages pane is expanded, and the selected server is highlighted under the Connector Configuration node. You can also navigate to this details page by clicking the tree nodes.

   - **CA Service Operations Insight Integration Framework Status**
     Indicates the status of the IFW on the system.
   - **Messaging Service Properties**
     Displays information regarding the MQ Server messaging service that controls the exchange of information between the connector, the IFW, and manager components.
   - **Connector Status**
     Contains status information about the connectors that are installed on this system.
   - **UCF Broker 1/UCF Binding 1**
     Contains information about UCF, or the common CA Catalyst connector framework.
3. Select Connector Configuration in the tree to return to the Connector Configuration page.
4. Select one of the connector identifiers in the Connector column.

   NOTE
   The tree view in the Administration Pages pane is expanded and the selected server is highlighted under the Connector Service node. Also, you can navigate to this details page by clicking the tree nodes.For more information about editing the connector, see Edit Connector Configuration.

Edit Connector Configuration

As an administrator, you can change the configuration settings on the Connector Configuration page.Connectors are initially configured when you install them.

Follow these steps:

1. Click the Administration tab and the Connector Configuration option.
   The Connector Configuration page opens and displays the Connector Status table.
2. Click one of the connector names in the Connector column.
   The connector details page opens and displays a number of tables with additional details about the connector and the server where it is installed. Many connectors also have a read-only Connector Type Data table that displays information such as the supported CI types.

   NOTE
   If your connector supports Scheduler Configuration and has the scheduler property enabled, then the button is available. For more information about Scheduler Configuration availability, see the connector documentation provided with your connector.

   WARNING
   The CA:00056_service-discovery@*servername* entry on the SA Manager server represents the Service Discovery component. This entry always has blank Connection Details. Do not edit any of the connector controls for this entry.
3. (Optional) Edit the following settings in the Connector Controls column as necessary, and click Save:
   - **dns_resolution**
     Specifies whether to use DNS resolution to resolve device names. If a reliable DNS mechanism is not in place (for example, no DNS server on the network, or configuration items (CIs) not defined to the DNS), disable DNS lookups to prevent CI resolution and normalization failure.
     Default: on
   - **useAlertFilter**
     Specifies whether to filter alerts based on their existence in a managed service in CA SOI. If the control is turned on, the connector only sends domain manager alerts that are associated with a CI that is part of an existing managed service in CA SOI. If the control is turne*d* off, the connector forwards all alerts from the domain manager, regardless of whether they relate to a service.
     Default: on

     NOTE
     For CA SOI r3.2 and later, the IFW no longer requires or uses EVENT_FILTER requests. The IFW now sends the alerts (managed or unmanaged) that the connectors report to the SA Manager. CA SOI ignores the global setting sendAllAlerts (in the IFW configuration file) and the connector-specific setting useAlertFilter (on the Administration tab Connector Configuration page). CA SOI now behaves as if sendAllAlerts=1 and useAlertFilter=0, regardless of the actual settings.
   - **getCIsAtStartUp**
     Specifies whether to rediscover CIs every time the connector starts. This control is enabled by default so that connectors always provide a current record of all CIs from their domain managers. You can turn this control off if the connector does not support collecting CIs at startup, such as the Universal or SNMP connector.
     Default: on
   - **isRemotable**
     Specifies whether to allow the connector framework to access the connector remotely for create, update, and delete operations on the source domain manager.
     Default: on
   - **useServiceFilter**
     Specifies whether to send all relationships to CA SOI or only the ones associated with modeled services. Set the control to true to run relationships through a service filter and receive only the relationships associated with modeled services.
     Default: on
   - **getRelationshipsAtStartup**
     Specifies whether to rediscover relationships every time the connector starts. This control is turned off by default so that relationships are only obtained and imported as a part of service model imports. You should only enable this

control if you require relationship CIs outside of imported service models, for example, if you define an Unmanaged Relationship Service Discovery rule.
Default: off

– **performDeltaProcessing**
Specifies whether to process and publish deltas on CIs between the time the connector or SA Manager was last stopped or restarted. When enabled, this setting also performs delta processing on relationships if the getRelationshipsAtStartup property is enabled.
Default: on

4. Select any field in the Connection Details, Connector Instance Data, and Launch in Context Details tables, edit the configuration setting, and click Save in the same table.
The changes are saved. Verify that any changes you make the connection settings do not break the connection. For example, if you modify a Host field in the Connection Details, first confirm that the domain manager is installed and configured on the new host.

> **NOTE**
> You cannot save the changes done in the Launch in Context Details pane. Do the changes in the configuration file. For information about the connector parameters or configuration file of a connector, see the CA Service Operations Insight Connectors wiki.

5. Click Stop and wait until the connector status changes to Offline.
6. Click Start and wait until the connector status changes to Online.
The connector is restarted. Depending on the type of connector, it can take a few minutes before the connector Status displays Online.

> **WARNING**
> Do not perform rapid start and stop operations on the connector. Each stop and start sends the corresponding command to the connector. Rapid start and stop operations from the interface can cause these commands to queue on the connector and cause the connector to start and stop repeatedly until all commands in the queue are processed.

Configure the CA CMDB Connector

To ensure that CA CMDB View works properly with the CA Service Desk connector version 3.2, it is necessary to add LicURL information to the CA CMDB connector configuration file.

Follow these steps:

1. Locate the ServiceDeskManagerConnector.conf connector configuration file.
The configuration file is located in the Catalyst registry in: topology\physical\<ContainerName>\modules\configuration.
If the Catalyst container was configured during the installation to use a file-based registry (which is the most common way for a CA SOI solution), then the location is: <CatalystContainerHome>\registry\topology\physical\<ContainerName>\modules\configuration.
If a Catalyst server is being used for the registry, then it is a similar directory accessible through the Catalyst Admin UI (or ws02 registry UI).
2. Add the following element to the configuration file:

```
<LaunchInContextUrls>
    <LicUrlTarget>
     <Label>Service Desk Manager</Label>
     <Type>CI</Type>
     <Url>http://CMDB-HOST:8080/CAisd/pdmweb.exe?OP=SHOW_DETAIL
+FACTORY={UrlParams}+PERSID={MdrElementID}</Url>
    </LicUrlTarget>
</LaunchInContextUrls>
```

This will allow you to launch the CA Service Desk Manager Web Client in the context of a CI in the CA SOI console.

NOTE
If you save the configuration from the Catalyst Admin UI, CA Catalyst alters the configuration slightly, adding the 'ns2' namespace to the elements. It also adds an empty section for LaunchInContextUrls:

```
<ns2:LaunchInContextUrls\>
```

Make sure you delete this section.

Configure the IFW Configuration File

As an administrator, you can change the connector-related settings that are saved in the IFW configuration file (<SOI_HOME>\resources\Configurations\SSA_IFW_HostName.xml). The settings are applicable to all connectors running under that IFW. You can configure the IFW configuration file to change these settings globally for all connectors on the system. For example, changing the alert filter setting at the IFW level automatically overrides the individual alert filter settings for all the connectors that are installed on that IFW system.

Follow these steps:

1. Open the <SOI_HOME>\resources\Configurations\SSA_IFW_HostName.xml file on a system with the IFW installed (any system with connectors or the SA Manager).
2. Change the value for the appropriate parameters in the ConnectorConfig section as follows, and save and close the file when finished:

   WARNING
   Use caution when changing these settings, and change documented settings *only*.

   – **retryCount**
   Defines how many times to retry connecting to the MQ Server component on the SA Manager when the connection fails. Before changing the default retry settings, consider that the amount of time a connector waits for the MQ Server connection reflects the amount of queued data that will consume memory in the IFW until the connection is reestablished.
   Default: 5
   – **retryInterval**
   Defines the number of seconds between retrying to connect to the ActiveMQ Server.
   Default: 30
   – **dns_resolution**
   Defines whether to use the DNS lookups for the CI name resolution. The default value of 1 turns on DNS lookups for all connectors on the system. When set to 1, you can also manage DNS lookup settings by connector if you want to use different settings for each connector. Change this value to 0 to disable the DNS lookups for all connectors. When you set this value to 0, you cannot manage DNS lookup settings by connector; DNS is always disabled for all connectors.
   Default: 1
   – **sendAllAlerts**
   Defines whether to filter or send alerts that are not associated with a managed service. The default value of 1 sends all alerts from all connectors. Change the value to 0 to let you specify this behavior by connector.
   Default: 1

   NOTE
   For CA SOI r3.2 and later, the IFW no longer requires or uses EVENT_FILTER requests. The IFW now sends the alerts (managed or unmanaged) that the connectors report to the SA Manager. CA SOI ignores the global setting sendAllAlerts (in the IFW configuration file) and the connector-specific setting useAlertFilter (on the Administration tab Connector Configuration page). CA SOI now behaves as if sendAllAlerts=1 and useAlertFilter=0, regardless of the actual settings.

   – **throttleConnectorStartup**
   Defines whether to initialize connectors on the system simultaneously or one after another. The default setting of 0 initializes all connectors simultaneously. You can change this setting to 1 for connectors to initialize sequentially,

which can improve the performance. The setting can also prevent memory overuse if there are multiple connectors on the system that manage large amounts of CIs.
Default: 0

3. Restart the CA SAM Integration Services service.

Change Connector Properties Using Connector Configuration File

As an administrator, you can change the basic connector properties. The following examples show common changes:

- The domain manager migrated to a different server.
- The domain manager user password changes every three months.
- In-context launch details may change such as the launch URL.

You can change these settings in the connector whenever they change in the domain manager so that communication is not interrupted.

Follow these steps:

1. Open the *connectorname_servername*.xml file located at <SOI_HOME>\resources\Configurations for the connector to edit.
2. Edit the necessary properties and save and close the file.
3. Perform one of the following actions:
   – Restart the connector from the Administration UI using Steps 4 and 5 from the previous procedure if you changed the properties of one connector.
   – Restart the CA SAM Integration Services service if you changed the properties of multiple connectors.
   – The change is applied, and the connectors begin monitoring the domain managers using the modified properties.

Configure Self Monitoring

CA SOI provides a global setting that enables basic self monitoring capabilities for connector status.

If you enable self monitoring, CA SOI creates a service named CA Service Operations Insight with a single Application CI of the same name. When a connector is no longer available, CA SOI generates an alert associated with the CA Service Operations Insight Application CI. Like any service, the CI status propagates to the service level, and the impact of the connector failure on the CA Service Operations Insight service changes accordingly.

Use the self monitoring capability to take proactive measures in response to service degradation, as you would for any CA SOI service. In this case, you can create escalation policies to take specific actions when a connector goes down, such as creating a help desk ticket or using a script to query the source domain manager to detect potential issues with the underlying application.

Follow these steps:

1. Access the CA SOI Dashboard.
2. Click the Administration tab, expand CA Service Operations Insight Manager Configuration, and click Global Settings.
3. Set Enable Self Monitoring to Yes.
4. Restart the CA SAM Application Server service, and, for distributed installations, the CA SAM User Interface Server service.
   The change takes effect, and CA SOI creates the CA Service Operations Insight service.
   Note: If you do not also restart the CA SAM User Interface Server service in distributed installations, the CA Service Operations Insight service model is not visible from the Dashboard, Mobile Dashboard and REST queries.

Turning off the Self Monitoring feature after you have previously enabled it stops further alerts on the Application CI. However, it does not remove the CA Service Operations Insight service model from the topology or clear existing alerts on the Application CI. You have to manually remove the service model and clear existing alerts.

NOTE

Removing the Universal connector from the Administration UI removes the Application CI in the CA Service Operations Insight service model. A running Universal Connector is required for self monitoring. Re-install the Universal connector and restart the CA SAM Application Server service to re-create the CI.

Managing CA Catalyst r3.x Connectors

Contents

As an administrator, you can change the connector administration settings, configure the encrypted connector property values, change the CA SOI password on a CA Catalyst connector, and delete the connector.

Configure Connector Administration Settings

By default, you can start or stop a CA Catalyst r3.2 connector from the CA SOI Administration UI using the Start and Stop buttons. You can configure whether these buttons actually start or stop the connector or simply stop the flow of connector data into CA SOI.

Follow these steps:

1. Access the ifw.properties file located at CATALYST_HOME*containerName*\registry\topology\physical *connectorserver*\ifw on the CA Catalyst connector system.
2. Set the connectorAdmin value to one of the following, and save and close the file:
   - **true (default value)**
     Lets you actually start and stop the connectors in the CA SOI Administration UI. The Start and Stop buttons in the Administration UI start or stop the connector in CA Catalyst.
   - **false**
     Lets you start or stop the data flow from connectors in the CA SOI Administration UI. The Start and Stop buttons do not actually control the connector status in CA Catalyst, only whether connector data makes it into CA SOI. If you set this value, you can start and stop connectors only from the CA Catalyst Registry files.
3. Restart the CA Catalyst Container *ContainerName* service on the connector system if you made a change to the file.

Change Encrypted Connector Property Values from Dashboard Administration Tab

The CA SOI Administration tab supports modifying CA Catalyst connector properties. However, if you change the value of an encrypted property, use CA Catalyst encryption so that CA Catalyst can process the new encrypted value.

Follow these steps:

1. Open a command prompt on a system with CA Catalyst r3.2 components installed.
2. Navigate to CATALYST_HOME*containerName*\tools\encrypt and run the following command:

   ```
   encrypter password
   ```

   - **password**
     Defines the new value of the encrypted property.

   An encrypted string appears.
3. Copy the encrypted string, and paste it into the property field on the CA SOI Administration UI.
4. Save the changes, and restart the connector.
   The encrypted value change takes effect.

Change the CA SOI Password on a CA Catalyst Connector System

If you change the CA SOI administrator password, you also change the password in the IFW Proxy on the CA Catalyst Container. Keep this password synchronized in CA SOI and the IFW Proxy to avoid connection issues.

Follow these steps:

1. Open a command prompt on the connector system, navigate to CATALYST_HOME*containerName*\ifw, and run the following command:
   ```
   EncryptSAMCreds newpassword
   ```
   – **newpassword**
     Defines the new password for the administrator user.

   The command generates an encrypted password.
2. Copy the password.
3. Open the CATALYST_HOME*containerName*\ifw\resources\configurations\SSA_IFW_*servername*.xml file, paste the new encrypted password into the password property, and save and close the file.
4. Open the CATALYST_HOME*containerName*\registry\topology\physical*connectorserver*\ifw\eventManagementServer.properties file, paste the new encrypted password into the password property, and save and close the file.

   NOTE
   If your CA Catalyst deployment includes a CA Catalyst Server, log in to the Registry Administration UI to access and modify this file.
5. Restart the CA Catalyst Container *ContainerName* service.
   The password change takes effect.

Remove the Connector and Connector Data from CA SOI

Removing CA Catalyst r3.2 connectors and their data from CA SOI does not require uninstalling the connector. Uninstalling the IFW Proxy deletes the connection between CA SOI and CA Catalyst, and you can remove existing connector data from CA SOI.

Follow these steps:

1. Uninstall the IFW Proxy on the CA Catalyst connector system by selecting Uninstall CA Service Operations Insight IFW Proxy from the Start menu.
   Uninstalling the IFW Proxy removes the connection between CA SOI and the CA Catalyst connector.
2. Find the connector entry in the CA SOI Administration UI and click Remove Connector.
   The connector data is deleted from the SA Store database, and the entry disappears from the tree when the operation is complete.
3. (Optional) Uninstall the connector if it is not required for other CA Catalyst solutions.

If you need to maintain a connection between CA SOI and other connectors on the same system, retain the IFW Proxy. Clicking Remove Connector on the CA SOI Administration UI still removes the CA Catalyst r3.2 connector data and updates the connector configuration to prevent subsequent use of the connector in CA SOI.

To enable a disabled connector, set the SOIState property to Enabled or remove the property altogether in the *connectorname*.conf file located at CATALYST_HOME*containerName*\registry\topology\physical*nodename*\modules\configuration.

NOTE

If your CA Catalyst deployment includes a CA Catalyst Server, log in to the Registry Administration UI to access and modify this file.

Security Administration

This section describes how administrators configure role-based security to control access to product functions.

Security Policy Statement

The CA SOI Security Policy Statement applies to the CA SOI product and is applicable as long as the product is used within the documented procedures defined in the product documentation.
The CA SOI Security Policy Statement details the encryption and hashing that is used by specific CA SOI components.
The CA SOI Security Policy Statement communicates the FIPS 140-2 statement for the CA SOI product. Specifically, it does the following:

- Clearly states what CA SOI modules are FIPS-compliant and FIPS-compatible
- Identifies FIPS certificate numbers for the encryption modules or hash algorithms used
- Communicates additional items that require extra physical security or protection
- Identifies the application boundaries surrounding the different application modules using encryption and or hashing
- Identifies what data is protected
- Communicates how keys are protected
- Explains how to enable FIPS mode on the software component

Definitions

The following terms are used in the CA SOI Security Policy Statement:
FIPS-compliant means that the component is capable of running FIPS-compliant encryption and hashing modules and offers the ability to run in FIPS mode.
FIPS-compatible means that the component uses FIPS-certified algorithms for encryption and hashing, but does not offer the ability to run in FIPS mode.

FIPS 140-2 Compatibility Matrix

The following table shows the extent to which CA SOI uses FIPS-compliant algorithms:

CA SOI Component	Module	Version	Certificate	Algorithms	Algorithm Certification No.	Mode
SOI Manager Password Hashing	BSAFE Crypto-J	5.1.1	1502	SHA-256	1549	Compatible
SOI UI Server Password Hashing	BSAFE Crypto-J		1502	SHA-256	1549	Compatible
Catalyst Password Hashing	BSAFE Crypto-J		1502	SHA-256	1549	Compatible
SOI UI Server Password Storage	BSAFE Crypto-J		714	AES-256	1766	Compatible
SOI OneClick Password Storage	BSAFE Crypto-J		714	AES-256	1766	Compatible
SOI Manager Password Storage	BSAFE Crypto-J			AES-256	1766	Compatible

EEM Password (Manager)	BSAFE Crypto-J			AES-256	1766	Compatible
EEM Proxy Password (Manager)	BSAFE Crypto-J			AES-256	1766	Compatible
EEM Password (Container)	BSAFE Crypto-J			AES-256	1766	Compatible
SQL Server Password	BSAFE Crypto-J			AES-256	1766	Compatible
Catalyst/IFW Password Storage	BSAFE Crypto-J			AES-256	1766	Compatible
Servicedesk Password Storage (NIM)	Bouncy Castle	3.2	N/A	AES-256	1766	Compatible
NIM Password Hashing	Bouncy Castle	3.2		SHA-256	1549	Compatible
NIM Password Storage	BSAFE Crypto-J	5.1.1		AES-256	1766	Compatible
CMDB Password Storage (Connectors)	BSAFE Crypto-J			AES-256	1766	Compatible
SNMPv3 Privacy (SNMP Connector)	SNMP4j	2.2.2		DES, 3DES, AES-128/192/256		Compliant
SNMPv3 Authentication (SNMP Connector)	SNMP4j	2.2.2		SHA, MD5		Compliant
Web Proxy Password (ServiceNow Connector)	BSAFE Crypto-J			AES-256	1766	Compatible
HTTPS TLS (ServiceNow Connector)	SunJSSE/NSS					Optional

Notes:

- HTPS TLS (ServiceNow Connector) can be used in any mode.
- These are the only algorithms that the software supports.
- The module column explains the types of modules the CA SOI components uses.
- To enable the FIPS mode on the software component, see Enable FIPS Mode in CA EEM.
- N/A means that the software does not offer the ability to operate in FIPS mode. Compatible means that the software is capable of operating in FIPS mode according to the definitions of those terms.

How to Configure Role-Based Security

As an administrator, you define user groups to manage users and user access privileges to services, alert queues, customers, and CA SOI features.

CA SOI integrates with CA EEM to manage the user authentication and configure the resource access. CA SOI adds users that are defined in CA EEM to product-specific user groups with configurable privileges and access levels.

The access privileges determine the features user group can access: CA SOI features, services, customers, and alert queues. For example, a person responsible for monitoring the Payroll Service may need to view or access only HR-related services. Likewise, if CA SOI is monitoring services for several internal or external customers, each customer should have access to their own information only.

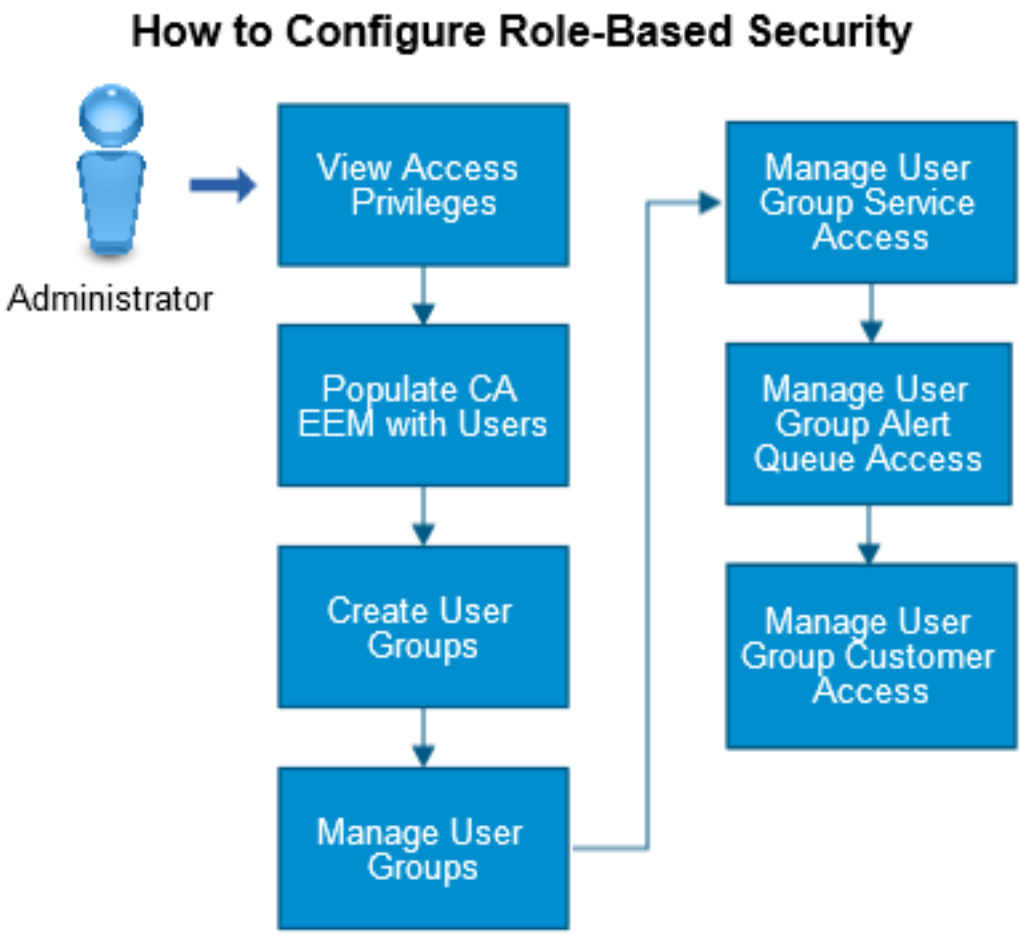
> **NOTE**
> For more information about installing CA EEM to integrate with CA SOI, see Install Java Runtime Environment and CA EEM. For more information about the CA EEM functionality, see the CA EEM documentation.

You create and use different administrative users to manage the product. The only user that is defined in CA SOI before you configure the role-based security is the user that was created during the installation ("samuser" by default).

For non-administrator user groups, you manage the user group access to services, alert queues, and customers.

Use this scenario to guide you through the process:

Figure 27: role-based security



1. Understand the Super User (samuser).
2. View the access privileges for the current user groups.

3. Populate the users in CA EEM and optionally in BusinessObjects.
4. Create the user groups, assign privileges, add users to the user group, and define users in BusinessObjects for report access.
5. Manage the user groups in CA SOI to define user group privileges to give the appropriate level of access to each group role.
6. Manage the user group access to services.
7. Manage the user group access to alert queues.
8. Manage the user group access to customers.
9. (Optional) Enable a guest user account.

Super User (samuser)

The only user that is defined in CA SOI before you configure the role-based security is the user that was created during the installation ("samuser" by default). The samuser is a super user with all privileges. Use samuser for the initial login and to configure user access. You can also use samuser if CA EEM connectivity is lost. Otherwise, you create and use a different administrative user for managing the product. The samuser has limitations that prevent it from being a long-term management solution, including the following limitations:

- It cannot set persistent Operations Console preferences.
- It cannot run reports.
- It cannot log in to USM Web View.

Predefined User Groups and Access Privileges

Contents

Predefined User Groups

CA SOI provides several default user groups with access to different CA SOI components and functionality. You can view the default access privileges in the Operations Console User tab.

- **Administrators**
  Users who have access to all CA SOI functionality.
- **Operators (read-only)**
  Users who view most CA SOI information such as alerts, customers, and services, but cannot modify information. These users cannot view the Dashboard Administration tab.
- **Operators (read-write)**
  Users who can view and modify most CA SOI information such as alerts, customers, and services, but cannot modify information. These users also have access to the Dashboard Administration tab.
- **Super Users**
  The default "samuser" user who has access to all CA SOI functionality. Use the "samuser" super user until you create other administrator users or if CA EEM problems prevent logging in as another administrator user. You cannot modify the privileges of the Super Users group.

View Access Privileges

CA SOI lets you define which user groups can view services and associated data and can manage alerts, CIs, UI access, and users. The Privileges tab in the Operations Console Contents pane lists the privileges and describes each one.

Follow these steps:

1. Start the Operations Console.
2. Click the Users tab in the Navigation pane.

3. Click a user group.
4. Click the Privileges, Service Access, Alert Queues Access, or Customer Access tab in the Contents pane.
   The access privileges for the selected user group appear.

Populate CA EEM with Users

Contents

As an administrator, you populate CA EEM with users. You can also import users into CA EEM from an external directory such as Active Directory or manually create users and store them to an internal database.

Import Users

You can import users from an external directory into CA EEM for use in CA SOI.

Follow these steps:

1. Open CA EEM as follows:
   – Enter the following URL in your web browser: http://<eem_server_name>:<port_number>/spin/eiam/eiam.csp. The default port number for CA EEM is 5250. Refer to the CA EEM section on the Installation Worksheet that you filled out during installation for these values. For more information, see Obtain the Installation Worksheet.
   – Select Start, Programs, CA, Embedded Entitlements Manager, EEM UI on the CA EEM system.

   The CA EEM login page opens.
2. Select <Global> from the Application drop-down list.
3. Enter the CA EEM Administrator user name (EiamAdmin) and password and click Log In.
4. Click the Configure tab.
5. Select the EEM Server (r8.x) or User Store (r12.x) from the submenu.
6. Click Global Users/Global Groups (r8.x) or Group Configuration (r12.x) in the left pane.
7. Select *one* of the following options in the right pane:
   – Store in internal datastore
   – Reference from an LDAP Directory (r8.x) or Reference from an external LDAP Directory (r12.x)
   – Reference from CA SiteMinder

   Note: For information about the page fields, see the CA EEM online help.
8. Click Save.
   One of the following icons appears:

   – **Success icon (green circle icon with a white check mark)**
     Indicates that both the External directory bind and the External directory data were loaded successfully.

   – **Warning icon (yellow triangle with red exclamation point)**
     Indicates that the External directory data is still loading. Allow additional time for the process to complete.

   – **Error icon (red circle with white x)**
     Indicates that the External directory bind failed. Check the inputs for each of the parameters and try again.

If you selected to have CA EEM reference an external directory and the operation was successful, the CA EEM integration is complete. You can start adding users to CA SOI.

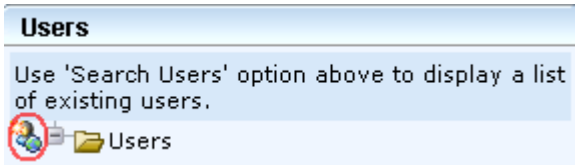
If you selected to store users in an internal datastore, you now create your users within CA EEM.

Create Users

You can create users directly in CA EEM for use in CA SOI. Later, you add these users to user groups in CA SOI.

Follow these steps:

1. Select the Manage Identities tab.
2. Select Users from the submenu.
3. Click the New User icon in the Users pane (lower left):



> **NOTE**
> When this icon is not visible, it means that users are being imported from external sources, such as Active Directory.

4. Complete the user details in the New User pane and click Save in the New User pane on the right side. At a minimum, enter a Name (alphanumeric characters only) and a Password for the user.
   A message at the top of the User panel confirms that the user was created.
5. Repeat Steps 3 - 4 for each new user.

Create User Groups

Contents

As an administrator, you create user groups to create user groups and add users to the group.

Define a User Group and Group Privileges

You assign feature privileges, service access, and alert queue access at the group level. A user group defines the access level for users in the group.

You can use the predefined groups that are provided with CA SOI, customize the predefined groups, or you can create your own user groups to use more specialized roles. Adding privileges during group creation defines what features, services, alert queues, and so on are available to users in the group.

> **NOTE**
> By default, each user group has access to *all* services and alert queues, but each user group has different feature privilege sets.

Follow these steps:

1. Log in to the Operations Console:
   - If you have not defined any administrator users, log in as the default user defined during installation ("samuser" user by default). Refer to the Installation Worksheet. For more information, see Obtain the Installation Worksheet.
   - If you have defined an administrator user, log in as that administrator.
2. Click the Users tab in the left pane.
3. Click the New User Group icon

   .
4. Enter a name for the group, and select the existing group to use as a template for assigning privileges in the Privilege Set drop-down list.
   You must select either Operator or Administrator as a starting point for assigning group privileges.

A list of privileges appears that are available for the privilege set you selected. Privileges are divided into folders corresponding to the functional areas of the product. Each privilege contains a detailed description, and all privileges are disabled by default.
Select the minimum privileges necessary for the group to manage their services, alert queues, and customers that their job function requires.

5. Expand all privileges, click the check boxes next to the privileges to enable for the group, and click OK.

   NOTE
   Select the Enabled check box in a parent folder to enable its children automatically.

   The group is defined. The privileges appear on the Privileges tab in the right pane, where you can click Add/Remove to modify the privileges.

Add Users to a Group

You add a user created in CA EEM to a group that has the privileges appropriate for their organizational role. A user can belong to one group only.

Note: Users that are created or imported into CA EEM are not added to a default user group in CA SOI, so you must add each user manually.

Follow these steps:

1. Open the Operations Console and click the Users tab in the left pane.
2. Expand the group to which to add users, and click the New User icon

   .
3. Click 'All users' or 'Users by filter'.

   NOTE
   When a large number of users exists, the list shows a truncated amount. Use filters in this case to find the users that you need.

   If you clicked 'Users by filter', complete the filter criteria.
4. Click OK.
   The Select Users dialog opens with users defined in CA EEM or available through CA EEM if integrating with an external directory.
5. Select one or more users and click OK.
   The user name appears beneath its group in the left pane. Users immediately inherit all privileges, access rights, and preferences that are assigned to the group.

   NOTE
   When you select a group in the left pane and click the Users List tab in the right pane, users in the group appear in a table in the right pane. From there you can delete, copy, and paste users and export the list in spreadsheet format.

Manage User Groups

Contents

As an administrator, you can change user group access privileges, update preferences, and export or delete user groups.

Modify Group Privileges

You can change the privileges that are assigned to user groups when group roles change.

NOTE
The Administrators group must have all privileges, so do not modify privileges for that group. You cannot modify the Super Users group privileges.

Follow these steps:

1. Open the Operations Console and click the Users tab in the left pane.
2. Select a group.
3. Click the Privileges tab.
4. Click Add/Remove.
5. Enable or disable privileges using the check boxes and click OK.
   The changes take effect and are reflected in the Privileges tab.

Edit User or Group Preferences

You can customize the product look and feel for an individual user or an entire user group. You can control the default settings for features such as field fonts, default alert filter, and Modeler settings. You can also set whether user groups can modify the values.

Follow these steps:

1. Right-click the user or group in the Users tab and select Set Preferences.
2. Set preferences for the user or group.
3. Select the check box in the Locked column to prevent users from modifying specific preferences.

   NOTE
   Privileges that are locked at the group level are not available when setting preferences for a user in that group.
4. Click OK.
   The preferences are saved.

Export the Users in a Group

You can export user definitions in a group to a CSV file and import that file to spreadsheet software.

Follow these steps:

1. Open the Operations Console and click the Users tab in the left pane.
2. Select a user group.
3. Click Export

   in the right pane.
4. Enter a file name, select a path, and click Save.

Delete a User or User Group

You can remove a user or user group that is no longer involved with CA SOI. CA EEM controls user authentication only. If you delete a user in CA EEM, you cannot log in to CA SOI with those user credentials. However, the user does not disappear from the Operations Console until you also delete the user from CA SOI.

NOTE
You cannot delete the predefined user ("samuser" user by default) or any of the predefined groups.

Follow these steps:

1. Open the Operations Console and click the Users tab in the left pane.

A list of user groups appears.

2. Expand the tree if necessary, select a user or group, and click Delete

The user or group is removed.

Manage User Group Access to Services

Contents

As an administrator, you can grant or revoke user groups access to all service models or specific service models. However, before you grant user group access, you should close the corresponding console of the Operator user. Each user group can have access to different services based on their role. For example, consider the following users and service requirements:

- Service owner users see only the services for which they are responsible.
- Customer users see only the services they consume.
- IT manager users see all services because they are responsible for maintaining the health and availability of all services in the data center.

Before you can grant user group access to the service models, you model services. For more information, see Service Modeling.

> **NOTE**
> The Administrators group must have access to all privileges, so do not modify privileges for that group.

Example User Group Access to Services

In this example, we have the following information in CA SOI:

Services: Sales, Finance, Operations

User Groups: Group1, Group2, Group3, Admin

The following table shows sample User Groups and their access to available services that are set by the system administrator:

User Group	Service Access
Group1	Sales, Operations
Group2	Finance, Operations
Group3	Operations
Admin	All Services

The following table shows the User Groups and what each user group sees on the Services tab based on their service access:

User Group	Sees on the Services Tab
Group1	Sales, Operations
Group2	Finance, Operations
Group3	Operations
Admin	All Services

Service and Sub-Service Access Situations

The following table shows various non-administrator user group permission access to services and its subservices and which services and subservices the user group sees.

The Example column uses the following service names: A, B, and C with each service having the subservice D.

User Group Service Permission	User Group Subservice Permission	Example	User Group Sees*	Notes
Not set	Not set	Access permissions are not set for services A, B, C, or subservice D	No services or subservices	
Allowed	Not set	Access permissions are set to Allowed for services A, B, C but not set for subservice D	Services A, B, C and subservice D	If service access is set to Allowed, then all subservice access is automatically set to Allowed
Allowed	Not Allowed	Access permissions are set to Allowed for services A, B, C and set to Not Allowed for subservice D	Services A, B, C	CA SOI does not support this situation. The user group sees services A, B, C and subservice D.
Not Allowed	Allowed	Access permissions are set to Allowed for services A and B and subservice D, but set to Not Allowed for service C.	Services A, B and subservice D under services A and B only	
Not Allowed for any parent	Allowed	Access permissions are set to Not Allowed for services A, B, C and set to Allowed for subservice D.	Subservice D	CA SOI does not support this situation. The user group does not see any service or subservice. However, you can create a placeholder parent service with subservice D and you can set both access permissions to Allowed.

* This column indicates what the user group expects to see. Unsupported situations are noted in the comments column and what the user group actually sees in CA SOI.

Grant/Remove User Group Access to Services

You can grant a user group access to any or all services defined in CA SOI.

1. Click the Users Tab and select a user group in the Navigation Pane.
2. Click Remove All Access to disallow users in the user group to access any services and click Yes when the confirmation dialog opens.
3. Click the Service tab in the Navigation pane.
4. Select a Service, right click, and choose Add User/Group Access.

   NOTE

   As an administrator, if you grant access to a user/group, close the user console and proceed further.
5. Choose the Groups you want to have access to the service and click OK.

Manage User Group Access to Alert Queues

Contents

As an administrator, you can permit or revoke user group privileges to all alert queues or specific alert queues.

You can think of alert access in several layers: user groups, alert queue access, and customers. In CA SOI, service access takes precedence over alert queue access. Therefore, if an alert is on a service that a user does not have access to, the alert does not appear in any alert queue that the user can see. Five users can view the same alert queue and see five different sets of alerts, depending on their user group service access privileges.

For procedures about working with alert queues, see How to Create and Manage Alert Queues.

> **NOTE**
> The Administrators group must have all privileges, so do not modify privileges for that group.

Example User Group Access to Alert Queues

In this example, we have the following data in CA SOI:

Services: Sales, Finance, Operations

> **NOTE**
> Because service privileges also dictate the services that appear in a particular alert queue, this example also includes the service access settings.

Alert Queues: Database Alerts, Critical Alerts

> **NOTE**
> For this example, the alert queue names indicate the type of alerts that each alert queue is configured to show. For example, if the Sales service has a critical alert, the Sales service appears in the Critical Alert queue (assuming the User Group has access privileges for the Sales service.)

User Groups: Group1, Group2, Group3, Admin

The following table shows the User Groups and their access to available services and alert queues:

User Group	Service Access	Alert Queue Access
Group1	Sales, Operations	Database Alerts
Group2	Finance, Operations	Critical Alerts
Group3	Operations	Database Alerts, Critical Alerts
Admin	All Services	All Alert Queues

The following table shows the User Groups and what they see in CA SOI based on their service and alert queue access:

User Group	Sees on the Services Tab	Sees on the Alert Queues Tab
Group1	Sales, Operations	Database Alerts queue with database alerts impacting to Sales and Operations services and all unmanaged database alerts.
Group2	Finance, Operations	Critical Alerts queue with critical alerts related to the Finance and Operations services and all unmanaged critical alerts only.

Group3	Operations	Database Alerts and Critical Alerts queues with critical alerts related to the Operations service and unmanaged critical database alerts only.
Admin	All Services	All alert queues with all managed and unmanaged alerts.

Grant User Group Access to Alert Queues

You can grant a user group access to any or all alert queues defined in CA SOI.

Follow these steps:

1. Click the Users tab then select a user group in the Navigation Pane.
2. Click the Alert Queues Access tab in the Contents pane.
3. Perform one of the following tasks:
   – Click Allow All Access to allow users in the user group access to all alert queues and click Yes when the confirmation dialog opens.
   – Click Add/Remove and select the alert queues available from the Available Alert Queues pane and move them to the Alert Queues Assign to this User Group Pane, then click OK.

   The user group alert queue access is updated and appears in the Contents pane.

Remove User Group Access to Alert Queues

You can revoke user group access to any or all alert queues defined in CA SOI.

Follow these steps:

1. Click the Users Tab then select a user group in the Navigation Pane.
2. Click the Alert Queues Access tab in the Contents pane.
3. Perform one of the following tasks:
   – Click Remove All Access to disallow users in the user group to access any alert queues and click Yes when the confirmation dialog opens.
   – Click Add/Remove and select the alert queues that are allowed in the Alert Queues Assigned to this User Group panel and move them to the Available Alert Queues panel, then click OK.

   The user group service access is updated and appears in the Contents pane.

Manage User Group Access to Customers

Contents

As an administrator, you can permit or revoke user group privileges to all customers or specific customers.

For more information about customers, see How to Create and Manage Customers.

Example User Group Access to Customers

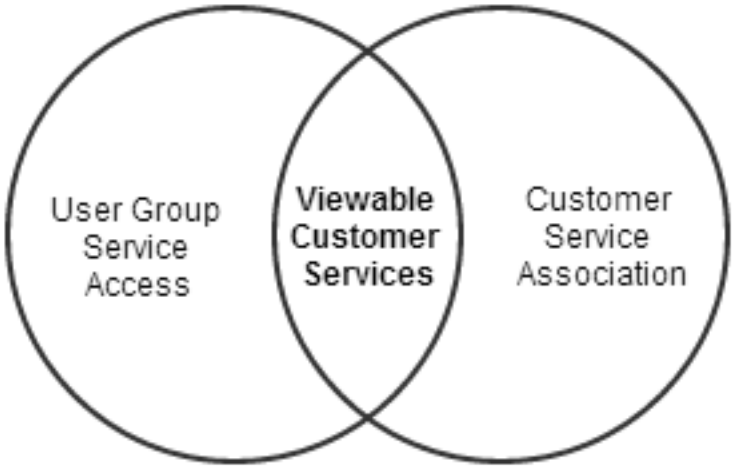
The following example shows how both service access and customer access limit the availability of services to user groups.

For a user group to see customer services, the user group must meet the following requirements:

- The user group must have access privileges to view the services.
- The services must be associated with a customer.
- The user group must have access privileges to the customer.

The following graphic shows that the viewable customer services is the intersection between the user group service access and the customer service association. User Group Service Access includes all services that the user group has access privileges to view. Customer Service Association includes all services that are assigned to a customer.

Figure 28: user group access to customers



The user group sees the Viewable Customer Services on the Operation Console Customer tab. The user group sees the User Group Service Access (all services to which they have access) on the Services tab.

For example, an administrator defined the following items in CA SOI:

User Groups: Operator1, Operator2, Operator3, Admin

Services Available: Finance, Operations, Sales

Customers: RegionUS, RegionEU

Customer Priorities: RegionUS=8, RegionEU=10

The administrator assigns the RegionEU customer a higher priority than RegionUS.

Customer Service Access: The administrator assigns the following services to the customers:

Region1: Sales, Finance, Operations

Region2: Finance, Sales

The following table shows the user groups, service access, and customer assignments that are set by the CA SOI administrator:

User Group	Service Access	Customer Assignment
Operator1	Operations, Sales	Region1
Operator2	Finance, Operations	Region2
Operator3	Operations	None
Admin	All customers	All customers

The following table shows the user groups and what each user group sees on several Operations Console tabs, based on the user group service and customer access:

User Group	In Services Tab	Services Tab: Impacted Customers Tab	In Customers Tab	Customers Tab: Services Tab
Operator1	Operations, Sales	Region1	Region1 only	Operations, Sales
Operator2	Finance, Operations	Region2	Region2 only	Finance
Operator3	Operations	None	None	None
Admin	All services	All customers	All customers	All service associations

The Operator1 user group sees only Operations and Sales in the Customers Tab. Operator1 is associated with the Region1 customer, which has service access to Finance, Operations, and Sales. However, the Operator1 user group is limited to only Operations and Sales service access; therefore, Operations and Sales are the only services Operator1 can see.

The Operator2 user group sees only Finance services in the Customers tab. The Operator2 user group has access to only Finance and Operations services and its customer association (Region2) has access to only Finance and Sales. Therefore, Operator2 is limited both by its user group access and by its customer access to view Finance only.

The Operator3 user group does not see any services. Although Operator3 has service access to Operations, the Operator3 user group is not associated with a customer (Region1 or Region2). Therefore, Operator3 cannot view any services associated with either customer.

The Admin user group can view all services and customers; therefore, the Admin user group sees all services.

Alert Queues: QueueUS, QueueEU, QueueSA

Customer and Subcustomer Access Situations

The following table shows various non-administrator user group permission access to customers and its subcustomers and what the user group sees.

The Example column uses the following customers and subcustomers:

Customers: Region1 and Region 2.

Subcustomers:

Region1: HR1

Region2: HR2

User Group Customer Permission	User Group Subcustomer Permission	Example	User Group Sees*	Notes
Not set	Not set	Access permissions are not set for customers Region1, Region2 and are not set for subcustomers HR1, HR2	No customers or subcustomers	
Allowed	Not set	Access permissions are set to Allowed for customer Region1 but not set for subcustomer HR1	Customer Region1 and subcustomer HR1	If customer access is set to Allowed, then all subcustomer access is automatically set to Allowed.

Allowed	Not Allowed	Access permissions are set to Allowed for customer Region1 but set to Not Allowed for HR1	Customer Region1	CA SOI does not support this situation. The user group sees customer Region1 and HR1.
Not Allowed	Allowed	Access permissions are set to Not Allowed for customer Region1 but set to Allowed for HR1	Subcustomer HR1 as top-level customer	

* This column indicates what the user group expects to see. Unsupported situations are noted in the comments column and what the user group actually sees in CA SOI.

Grant User Group Access to Customers

You can grant a user group access to any or all customers defined in CA SOI.

Follow these steps:

1. Click the Users Tab then select a user group in the Navigation Pane.
2. Click the Customer Access tab in the Contents pane.
3. Perform one of the following tasks:
   – Click Allow All Access to allow users in the user group access to all customers and click Yes when the confirmation dialog opens.
   – Click Add/Remove and select the customers available from the Available Customers pane and move them to the Customers Assigned to this User Group Pane, then click OK.

   The user group customer access is updated and appears in the Contents pane.

Remove User Group Access to Customers

You can remove a user group access to any or all customers (and sub-customers) defined in CA SOI.

Follow these steps:

1. Click the Users Tab then select a user group in the Navigation Pane.
2. Click the Customer Access tab in the Contents pane.
3. Perform one of the following tasks:
   – Click Remove All Access to disallow users in the user group to access any customers and click Yes when the confirmation dialog opens.
   – Click Add/Remove and select the customers that are allowed in the Customers Assigned to this User Group panel and move them to the Available Customers panel, then click OK.

   The user group service access is updated and appears in the Contents pane.

Enable Guest User Account

As an administrator, you can enable a guest account that lets users access the CA SOI Dashboard without login credentials. The feature is disabled by default.

Follow these steps:

1. Locate and open the following file on the UI Server:
   <SOI_HOME>\SamUI\webapps\sam\server-config.xml
2. Add the following entry as the second to last line (below the </manager> line):
   ```
   <guest-username>guest</guest-username>
   ```
3. Restart the UI Server service.

The patch adds user "guest" to the "Operators (read-only)" user group. You can adjust the access privileges as necessary.
Your users use the following URL to access the CA SOI Dashboard as the guest user:

```
http://<UI_server>:7070/sam/guest.jsp
```

Support for Common Access Card and Smartcard Authentication Using Client Certificates

As an administrator, you can enable Common Access Card (CAC) and Smartcard authentication using client certificates. You can use your client certificates to authenticate users in CA SOI. Only authorized users can access the environment (for example, access to the web services).

Points to Remember:

- **DoD Certificate:** Use this certificate for CAC Card configuration.
- **casoiroot Certificate:** Use this certificate for configuring SSL for CA SOI.
- If you want to generate a server certificate and keystore, ensure that you follow the procedure that is mentioned in Generate Root Certificate, Server Certificate, and Keystore.

The following table summarizes the certificates that are required to enable CAC authentication.

CA SOI Servers:

Keystore	File System Location	Required Certificates
UI Server	<SOI_Home>\SamUI\conf\ssa.jks	• DoD root and intermediate • casoiroot • server (alias tomcatssl)
SA Manager	<SOI_Home>\tomcat\conf\ssa.jks	• casoiroot • server (alias tomcatssl)
Java JRE	<SOI_Home>\jre-64\lib\security\cacerts <SOI_Home>\jre-32\lib\security\cacerts <SOI_Home>\jre\lib\security\cacerts	• DoD root and intermediate • casoiroot

Windows Clients

Keystore	Application	Required Certificates
Windows	certmgr.msc	DoD root and intermediate casoiroot
Mozilla Firefox		casoiroot

To enable the CAC and Smartcard authentication, follow these steps:

1. Enable FIPS Mode in CA EEM
2. Configure Active Directory in CA EEM
3. Generate Certificates and Keystore
4. Configure UI Server to Enable CAC Authentication
5. Configure CA SOI for New Smart Card
6. Enable SSL for SA Manager
7. Enable SSL for Enterprise Domain Connector
8. Configure Windows Client with CA SOI Server Certificates

Prerequisites

Before you enable the client certificates, perform the following prerequisites:

Install Java Patch for CAC

Download the bundled java patch from the following location:

http://www.java.com/en/javaforbusiness/download-8u102b34.jsp

Install ActivIdentity ActiveClient

The ActivIdentity ActiveClient is a card reader software that is used for desktops, network security, and productivity applications. CA SOI supports ActivIdentity ActivClient version 7.0.2.

NOTE

Install this software only on your Client Windows System.

Configure Mozilla Firefox Browser for Card Reader

The Firefox browser requires extra configuration settings to work with Smartcard reader.

Follow these steps:

1. Click **Open menu** on the top right corner of the Mozilla Firefox browser.
2. Click **Options**, **Advanced**, and **Certificates** tab.
3. Click **Security Devices.**
   The **Device Manager** window appears.
4. Click **Load** option, and enter the following information in the **Load PKCS#11 Device** window.
   a. **Module Name:** My CAC Reader
   b. **Module Filename :** C:\Program Files (x86)\ActivIdentity\ActivClient\acpkcs211.dll
5. Click **OK**.

The Firefox browser is configured to work with Smartcard reader.

NOTE

The Internet Explorer and Chrome does not require any extra configuration settings.

Enable FIPS Mode in CA EEM

CA EEM has a Federal Information Processing Standard (FIPS) mode that works with CA SOI. To enable FIPS mode, follow these steps:

1. Stop the CA SOI Services.
2. Open the **eiam.config** file that is available in the following locations:
   – <SOI_Home>\SamUI\webapps\sam\
   – <SOI_Home>\tomcat\webapps\sam\
3. Do the following changes:
   a. Change **<FIPSMode>** from **Off** to **On.**
   b. Provide the **(FQDN) name of your EEM server** in **<Backend>**
      See the bold text in the following example:

```
<EiamConfiguration>

    <!-- Socket timeout in milli seconds. Default value is 2 mins -->
```

```
    <Network sockettimeout="120000"/>

    <SDK type="Java">

        <iTechSDK>

            <FIPSMode>On</FIPSMode>

            <!-- <JCEProvider>JsafeJCE</JCEProvider> -->

            <Security>

                <digestAlgorithm>SHA1</digestAlgorithm>

            </Security>

        </iTechSDK>

    </SDK>

    <!-- configuration to create SafeContext instance from SafeContextFactory -->

    <SafeContext refid="SOI" version="1.0">

        <!-- EEM server hostname -->

        <Backend>EEM-FIPS-SERVER.example.com</Backend>

    </SafeContext>

</EiamConfiguration>
```

NOTE

Ensure that you modify the eiam.config file in both the locations.

4. Restart the CA SOI Services.

NOTE

Configure the EEM Server for FIPS mode and restart the server

Generate Certificates and Keystore

NOTE

These procedures are applicable if your organization has not provided any certificates.

A new CA SOI system requires a server certificate and keystore for UI Server and SA Manager. The server certificate is based on a root certificate. The root certificate can be loaded into the various keystores and marked as trusted.

NOTE

The CA SOI server systems within an organization must share the same root certificate.

To generate certificates and keystore, follow these steps:

Download OpenSSL for Windows.

Before you generate a certificate and keystore, download the OpenSSL for Windows.

Follow these steps:

1. Download Openssl from https://sourceforge.net/projects/openssl/files/latest/download location. The openssl folder is downloaded to your system.
2. Unzip the Openssl folder.
3. Create **CERTS** folder in C drive.
   For example, C:\CERTS
4. Navigate to **<opensslfolder>\bin** and copy the **openssl.exe** and **openssl.cnf** to **CERTS** folder.

Generate Root Certificate, Server Certificate, and Keystore

After you copy the openssl.exe and openssl.cnf file to your local system, you must generate root certificate, server certificate, and Keystore.

NOTE

Generate a root certificate only once. The openssl utility creates a **demoCA** folder (for example, C:\CERTS \demoCA). Ensure that you retain this folder because when you add CA SOI system to an enterprise domain connector you require all the server certificates singed by the same root certificates.

Follow these steps:

1. From the command prompt, navigate to the **CERTS** folder by using the following command:

   ```
   cd C:\CERTS
   ```

2. **Generate root certificate:**
   a. Set the following options:

      ```
      set OPENSSL_FIPS=1

      set OPENSSL_CONF=C:\CERTS\openssl.cnf
      ```

   b. Create a private key and a certificate request, and place it as CASOIRoot.pem by using the following command. The following example creates a certificate that is valid for ten years, that is, (days = 365x10).

      ```
      openssl.exe req -x509 -new -nodes -key CASOIRoot.key -days 3650 -out CASOIRoot.pem
      ```

      You are prompted to enter the following information. The Common Name must be **CA SOI Root.** This information is added to your certificate. For example:

      ```
      Country Name (2 letter code) [AU]:US
      State or Province Name (full name) [Some-State]:MA
      Locality Name (eg, city) []:Boston
      ```

```
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CA Inc
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:CA SOI Root
Email Address []:
```

The root certificate is generated.

3. **Generate Server Certificate and Keystore:**

NOTE

- Generating server certificate and keystore must be done for each CA SOI server. Each CA SOI system must have its own server certificate.
- We assume that your CA SOI Server name is soi1.ca.com.

1. a. Use the openssl**genrsa** command to create the private key file of the server.

```
openssl.exe genrsa -rand openssl.exe -out soi1.key 2048
```

b. To generate a signing request for the server certificate, use the following opensll **req** command.

```
openssl.exe req -new -key soi1.key -out soi1.csr
```

You are prompted to enter the following information:

```
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:MA
Locality Name (eg, city) []:Boston
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CA Inc
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:soi1.ca.com
Email Address []:
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

NOTE

- The Common Name must be the FQDN of the CA SOI server.
- Leave the challenge password blank.

c. To sign the server certificate with the root key, use the following openssl**ca** command. The following example creates a certificate that is valid for ten years (days = 365x10).

```
openssl.exe ca -keyfile CASOIRoot.key -cert CASOIRoot.pem -notext -in soi1.csr -
out soi1.crt -days 3650
```

d. To generate a temporary keystore **KEYSTORE.p12** that holds the CA SOI server certificate, use the following openssl**pkcs12** command.

```
openssl.exe pkcs12 -export -in soi1.crt -inkey soi1.key -out KEYSTORE.p12 -name
 tomcatssl -CAfile CASOIRoot.pem -caname CASOIRoot
```

When prompted for the password, enter **catalyst.**

e. To delete the existing ssa.jks file, use the following command.

```
del ssa.jks
```

f. To generate a keystore**ssa.jks** that holds the CA SOI server certificate, use the following **keytool -importkeystore** command.

```
keytool.exe -importkeystore -srckeystore KEYSTORE.p12 -srcstoretype PKCS12 -
srcstorepass catalyst -destkeystore ssa.jks -deststorepass catalyst
```

g. To import the CA SOI root certificate into the keystore **ssa.jks,** use the following **keytool -importcert** command.

```
keytool.exe -importcert -alias casoiroot -file CASOIRoot.pem -trustcacerts -
keystore ssa.jks -storepass catalyst
```

The keystore (**ssa.jks**) is created.

Use the **ssa.jks** keystore in the next step: Configure UI Server to Enable CAC Authentication.

Configure UI Server to Enable CAC Authentication

After you have generated the server certificate and keystore, configure the UI server to enable CAC authentication.

Follow these steps:

1. Stop the CA SOI services.
2. Modify the following xml files:
   a. **ssa.jks**

      NOTE

      For more information about keystore, see Generate Certificates and Keystore.

      a. Navigate to the following locations:
         a. <SOI_Home>\SamUI\conf
         b. <SOI_Home>\tomcat\conf
      b. Back up the **ssa.jks** file from UI Server and tomcat folder
      c. Copy the new **ssa.jks** file in UI Server and tomcat folder.
   b. **web.xml**
      a. Navigate to **<SOI_Home>\SamUI\webapps\sam\WEB-INF\web.xml** location.
      b. Change all instances of **<transport-guarantee>** from **NONE** to **CONFIDENTIAL**For example,

```
<user-data-constraint>
<description>
</description>
```

```
<transport-guarantee>CONFIDENTIAL</transport-guarantee>
</user-data-constraint>
```

c. Save and close the file.

c. **server.xml**

a. Navigate to **<SOI_Home>\SamUI\conf\server.xml** location.

b. Edit the xml as follows:

a. Add "**tomcatssl**" to keyAlias

NOTE

keyAlias is the alias used in the keystore for CA SOI UI Server certificate.

b. Change the clientAuth to "**true**"

c. Add "**TLS**" to sslPortocol

For example,

```
<Connector address="${tomcat.inaddr.bind}" port="${tomcat.port.ssl}"
 protocol="HTTP/1.1" SSLEnabled="true"
  maxHttpHeaderSize="8192" maxThreads="150" minSpareThreads="25"
 maxSpareThreads="75"
  enableLookups="false" disableUploadTimeout="true"
  acceptCount="100" scheme="https" secure="true"
  keystoreFile="${tomcat.keystore.file}"
  keystorePass="${tomcat.keystore.pswd}"
  keyAlias="tomcatssl"

  clientAuth="true" sslProtocol="TLS" />

  .

  .

  ./>
```

c. Save and close the file.

d. **cac-system-config.xml**

Encrypt the value that used for the password field <trust-keystore-pass> by using the following command. The argument to the command is the password to encrypt.

```
"<SOI_Home>\tomcat\bin\WSSamEncryptCmd.bat" catalyst
```

NOTE

The password for <trust-keystore-pass> must be same as CAC keystore password, which is "catalyst".

a. Navigate to **<SOI_Home>\SamUI\webapps\sam\WEB-INF\cac\config\cac-system-config.xml** location.

b. Edit the xml file as follows:

a. Change all the instances of <crl-enabled> or <ocsp-enabled> to **true**.

NOTE

If both <crl-enabled> and <ocsp-enabled> are set to true, then <crl-enabled> takes precedence.

b. Set <cac-debug-level> value to **0**.

NOTE

Set **0** for production and **3** for debugging.

c. Set <trust-keystore-pass> to the value that you have generated while encrypting the password.
d. Set <ocsp-url> value to **http://ocsp.nsn0.rcvs.nit.disa.mil/ http://ocsp.nsn0.rcvs.nit.disa.mil/** .
e. Set <ext-rule> value to **CN=(.*)\.\d+** For example,

```
<root>
  <cac-enabled>true</cac-enabled>
  <cac-debug-level>0</cac-debug-level>
  <trust-keystore-pass>EBrNaLYMXwvJS5e5C6JkPHfq2nz5QTgnrJ0o7MgedgjZ</trust-
keystore-pass>
  <cert-edipi>subject</cert-edipi>
  <ext-rule>CN=(.*)\.\d+</ext-rule>
  <cac-crl>
    <crl-enabled>true</crl-enabled>
    <crl-dp>false</crl-dp>
    <crl-url></crl-url>
    <crl-dir>C:/Program Files (x86)/CA/SOI/SamUI/CRLS</crl-
dir>
    <crl-poll-int>10000</crl-poll-int>
  </cac-crl>
  <cac-ocsp>
  <ocsp-enabled>true</ocsp-enabled>
  <ocsp-cert-alias></ocsp-cert-alias>
  <ocsp-aia>true</ocsp-aia>
  <ocsp-url>
http://
ocsp.nsn0.rcvs.nit.disa.mil
/</ocsp-url>
  </cac-ocsp>
</root>
```

c. Save and close the file.

3. Start the CA SOI services.

Configure CA SOI for New Smart Card

To configure the CA SOI environment with a Smart Card (CAC) and when a new CAC user is added to an already configured CA SOI system, perform the following steps:

NOTE

Some cards use the same root or intermediate certificates. If you already have a CA SOI system or Windows client that is configured for CAC, then the following procedures might not be required.

For example, if the new CAC has the same root or intermediate certificates as one for which the system has already been configured.

Prerequisites

Ensure that the required certificates are installed on the Windows client and CA SOI Server.

Windows Client

From the command prompt, run **certmgr.msc** command and verify if the certificates exist in the following folders:

- "Trusted Root Certification Authorities" for root certificates, or
- "Intermediate Certification Authorities" for intermediate certificates

CA SOI Server

Use the following command to list the contents of the keystore:

```
keytool -list -keystore <keystore file> -storepass <keystore password>
```

For CA SOI, the following two sets of keystores (truststores) are available:

1. **Tomcat for UI Server and SA Manager:**
   a. <SOI_Home>\SamUI\conf\ssa.jks
   b. <SOI_Home>\tomcat\conf\ssa.jks

   NOTE

   Use catalyst as the password.
2. **Java:**
   a. <SOI_Home>\jre-64\lib\security\cacerts
   b. <SOI_Home>\jre-32\lib\security\cacerts
   c. <SOI_Home>\jre\lib\security\cacerts

   NOTE

   Use changeit as the password.

Example to list the contents of the keystores:

```
 "C:\Program Files (x86)\CA\SOI\jre-64\bin\keytool.exe" -list -keystore "<SOI_Home>
\SamUI\conf\ssa.jks" -storepass catalyst
"C:\Program Files (x86)\CA\SOI\jre-64\bin\keytool.exe" -list -keystore "<SOI_Home>
\jre-64\lib\security\cacerts" -storepass changeit
```

Add CAC User to CA EEM Server

Add each CAC user to the CA EEM server which is used by CA SOI servers. For more information, see User Access Permissions. The username in CA EEM must match the name on the CAC Certificate. You can find the name by performing the following steps:

NOTE

This procedure is applicable on a CAC installed system and Internet Explorer.

1. From the Internet Explorer, navigate to **Settings**, **Internet Options**.
2. Click the **Content** tab, **Certificates**.
3. Select a certificate from the **Issued To** field and note the certificate name (EMM username). The username is everything before the last period and 9 or 10 digit number, including any embedded spaces and periods.
   For example, If the certificate name is FOREIGN.NON-CITIZEN.1403001852, then the EEM username must be FOREIGN.NON-CITIZEN. Use this name to add a user in CA EEM. For more information about how to add users, see Add Users.

Obtain CRLs for DoD Intermediate and Root Certificates

If you have configured the CA SOI CAC feature to enable certificate verification through CRLs, you must download the CRL for each intermediate and root certificates for each CAC configured.

NOTE

The CRL files expire so ensure that you update it on an ongoing basis.

1. Create a folder <SOI_Home>\SamUI\CRLs.
2. Copy each CRL file to the CA SOI folder.

Obtain DoD Intermediate and Root Certificates for CAC

Download the intermediate and root certificates used by each CAC. You can install these certificates into the various keystores (in the next step).

Install DoD Intermediate and Root Certificates on Client Windows Keystore

To install the DoD intermediate and root certificates for CAC on client Windows Keystore, follow these steps:

NOTE

The following steps must be performed on each Windows client that connects to the CA SOI server.

1. In Windows Explorer, right-click on certificate, and click **Install Certificate.**
2. Ignore security warnings, and click **Open**.
   The Certificate Import Wizard appears.
3. On the Welcome page, select **Local Machine**, and click **Next**.
4. On the Certificate Store page, select **Place all certificates in the following store,** and click **Browse** .
5. In the **Select Certificate Store** dialog, select any *one* certificate:
   a. "Trusted Root Certification Authorities" for root certificates
   b. "Intermediate Certification Authorities" for intermediate certificates
6. Click **Next**.
7. On the summary page, review the details, and click **Finish.**
   The certificate is imported.

Import DoD Intermediate and Root Certificates to CA SOI UI Server Keystore and Java Keystore

To import DoD intermediate and root certificates for CAC on CA SOI UI Server Keystore and Java Keystore, follow these steps:

1. Add a certificate to a keystore by using the following command.
```
keytool -importcert -alias <certificate alias> -file <certificate file> -
trustcacerts -keystore <keystore file> -storepass <keystore password>
```

For a root certificate, enter **Yes** to trust the certificate.

2. View the location of keytool.exe and the two keystores used by CA SOI (Tomcat and Java).
3. For the <certificate alias>, use the name of the certificate file in all lowercase, without the extension, and without any non alphanumeric characters (like '-' and '_', and so on.)
   For example, The alias for DODOMCA_30.cer would be "dodomca30", and the alias for DODJITCROOTCA2.cer would be "dodjitcrootca2"

Examples:

1. The following example imports a root certificate that exists in the CA SOI folder into the CA SOI UI server keystore:
```
"C:\Program Files (x86)\CA\SOI\jre-64\bin\keytool.exe" -importcert -alias
dodjitcrootca2 -file DODJITCROOTCA2.cer -trustcacerts -keystore
"C:\Program Files (x86)\CA\SOI\SamUI\conf\ssa.jks" -storepass catalyst
Trust this certificate? [no]: yes
```

2. The following example imports a root certificate that exists in the CA SOI folder into the CA SOI Java keystore:
```
"C:\Program Files (x86)\CA\SOI\jre-64\bin\keytool.exe" -importcert -alias
dodjitcrootca2 -file DODJITCROOTCA2.cer -trustcacerts -keystore
"C:\Program Files (x86)\CA\SOI\jre-64\lib\security\cacerts" -storepass changeit
Trust this certificate? [no]: yes
```

3. After you have added all of the required DoD certificates to the Java keystore, you can copy the certificates to the following two other Java keystores:
   "C:\Program Files (x86)\CA\SOI\jre-32\lib\security\cacerts"
   "C:\Program Files (x86)\CA\SOI\jre\lib\security\cacerts"

   NOTE

   After you ave performed all the above procedures, restart the CA SOI servers.

Enable SSL for SA Manager

For encrypted communications, enable SSL for each Apache Tomcat server in the CA SOI configuration. To enable SSL for the SA Manager, follow these steps:

1. Stop the CA SOI services.
2. Modify the following files:
   a. Open **server-config.xml** and edit the file as follows. This file is located where the UI server is installed.
      a. Navigate to <SOI_Home>\SamUI\webapps\sam\server-config.xml folder, and edit the xml file as follows:
         a. <port> to 7493
         b. <protocol> to https
         For example,

```
<root>
 <admin-username>samuser</admin-username>
 <admin-password plain="false">EE6wpG1M+BrHrhjcgRe3c8VQk75epwOjDeH+te+MXQKB</
admin-password>
 <manager>
  <host>[FQDN server name]</host>
  <port>7493</port>
```

```
  <protocol>https</protocol>
  <trusted>true</trusted>
 </manager>
</root>
```

b. Save and close the file.

b. **web.xml**
   a. Navigate to **<SOI_Home>\tomcat\webapps\sam\WEB-INF\web.xml** folder.
   b. Change all instances of **<transport-guarantee>** from **NONE** to **CONFIDENTIAL**.
   For example,

```
<user-data-constraint>

    <description>

    </description>

      <transport-guarantee>CONFIDENTIAL</transport-guarantee>

</user-data-constraint>
```

   c. Save and close the file.

c. **server.xml**
   a. Navigate to <SOI_Home>\tomcat\conf\server.xml location and edit the the xml file as follows:
      a. Add **keyAlias** as "**tomcatssl**"
      b. Verify if **clientAuth** is **"false"**
      c. Add **sslProtocol** as **"TLS"**
      For exmaple,

```
<Connector address="${tomcat.inaddr.bind}" port="${tomcat.port.ssl}"
 protocol="HTTP/1.1" SSLEnabled="true"

      maxHttpHeaderSize="8192" maxThreads="150" minSpareThreads="25"
 maxSpareThreads="75"

      enableLookups="false" disableUploadTimeout="true"

      acceptCount="100" scheme="https" secure="true"

      keystoreFile="${tomcat.keystore.file}"

      keystorePass="${tomcat.keystore.pswd}"

      keyAlias="tomcatssl"


      clientAuth="false" sslProtocol="TLS"
```

   b. Save and close the file.
3. Restart the CA SOI services.

Enable SSL for Enterprise Domain Connector

On the CA SOI Enterprise Domain system, configure the Domain Connector for SSL communication with the source CA SOI system.
The following steps assume that you have already configured the Domain Connector without SSL.

To enable SSL for enterprise domain connector, follow these steps:

1. Modify the Connector Configuration file at the <SOI_Home>\resources\Configurations path.
2. Each domain connector for every source CA SOI system has a configuration file. For example, the configuration file name is ssaDomain_<backEnd-hostname>.xml
3. Open each domain connector file in an editor, and change the **ConnectionInfo** section.
   a. Modify the port attribute from "**7090**" to "**7493": port="7493"**
   b. Add an attribute: isSSL="true"
   c. Verify that the host attribute uses the full qualified domain name (FQDN) of the source SOI system.
4. Save and close the file.

Configure Windows Client with CA SOI Server Certificates

The CAC requires mutual authentication, which means that both ends of the communication (the Windows client or browser and the CA SOI server) must send a certificate to the other end. The certificate must have a trusted root certificate for the opposite end to accept that particular certificate.

For example:
Windows client: <SOI server certificate> to <SOI trusted root certificate>
CA SOI server: <CAC certificate> to <DoD intermediate certificate> to <DoD trusted root certificate>
To configure the Windows client system to accept the CA SOI server certificates, follows these steps:

1. Configure the CA SOI system with the CAC certificates.

   NOTE

   We recommend that the same root certificate must sign every CA SOI server certificate. If the Windows client is already configured for one CA SOI system, then it must work with other CA SOI systems without any extra configuration.
2. Verify whether the **CA SOI ROOT** certificate is installed in your system.
   a. From the command prompt, run the following command:

      ```
      certmgr.msc
      ```

      The **certmgr - [Certificates - Current User]** window appears.
   b. Open **Trusted Root Certification Authorities** folder, and verify for the CA SOI Root certificate.

      NOTE

      If the certificates are not installed, then go to step 3.
3. Install the root certificate:
   a. In Windows Explorer, right-click on certificate and click **Install Certificate**.
   b. Ignore security warnings, and click **Open**.
   c. On the Welcome page, select **Local Machine** and click **Next.**
   d. On the Certificate Store page, select **Place all certificates in the following store,** and click **Browse.**

e. In the **Select Certificate Store** dialog, select **Trusted Root Certification Authorities** for the root certificate.
f. Click Next
g. On the summary page, review the details, and click **Finish.**
The certificate is imported.

NOTE

For more information about CA SOI server root certificate, see Generate Certificates and Keystore.

Limitations

The following are the limitations when you use CAC with CA SOI.

- **Jaspersoft**
  Jaspersoft reporting supports only basic authentication when you enable CAC for CA SOI. Jaspersoft reporting does not support CAC authentication.
- **CA SOI RESTful API**
  The Rest API functionality does not work when you enable CAC. You can log in to the REST API from a browser using your CAC, but application cannot to access it.
  The following CA SOI functionalities are not available when CAC is enabled:
  – Mobile View
  – USM Web View
- **SA Manager Debug Configuration Page**
  The CA SOI debug configuration page for SA Manager is not protected by CAC when you enable CAC. The configuration page supports basic authentication. However, only users in the admin group can log in with basic authentication.

  NOTE

  The UI Server debug configuration page is protected by CAC.
- **CA EEM UI**
  CA EEM UI does not support CAC.
- **Universal Connector**
  The Universal Connector does not support SSL. When you enable SSL for SA Manager, the Universal connector fails to start.
- **Launch In Context**
  The Launch in Context feature of the CA SOI Dashboard does not support launching of remote applications. You can only launch local applications.
  You can remove the Launch In Context menu items from the Dashboard by:
  – Editing the appropriate configuration file
  – Removing or commenting the sections **<LICURLS>**
  – Restarting the service
  The configuration files are at **<SOI_HOME>\resources\Configurations\<connector configuration file>** on the connector system.

CAC Troubleshooting

Authorization Failed Error After Entering PIN and Selecting Certificate

Symptom:

After I enter the CAC pin and select a CAC certificate in UI server, the following error appears:

CA Service Operation Insight - CAC Authorization Failure

Authorization failed. Please contact your system administrator.

Solution:

Follow these steps:

1. Ensure that the CAC user is correctly added to the system (especially CA EEM).
   The following error might appears:

   ```
   2016-10-24 16:27:06,592 INFO  [http-bio-0.0.0.0-7403-exec-4]
    authenticator.ExternalSSOAuth.authenticate(265)  - Your certificate cannot be
    verified. Please contact your system administrator.

   java.security.cert.CertPathValidatorException: Path does not chain with any of the
    trust anchors at
    sun.security.provider.certpath.PKIXCertPathValidator.validate(PKIXCertPathValidator.java:153
   ```

   You can view the error in the log file <SOI_Home>\SamUI\logs\soiuis.log
2. Restart the browser.
   For example, If you have any browser sessions open while enabling CAC, the CA EEM sessions might still be valid which interferes with the CAC authentication.

   NOTE

   For more information, see Configure CA SOI for new Smart Card.

Error from Smart Card Reader When Trying to Launch CA SOI OneClick Console

Symptom:

When I try to launch the CA SOI OneClickConsole (JNLP App), the following error appears:

The smart card cannot perform the requested operation or the operation requires a different smart card.

Solution:

The error appears because of a bug in Java, you must install a Java patch. For more information about how to install Java, see Install Java Patch for CAC.

Secure Connection Error when Trying to Connect to UI Server From Browser

Symptom:

When I am trying to connect to the UI Server URL from a browser, it fails with a secure connection error.

Example:

- On Firefox, the following error appears.
  "Secure Connection Failed
  An error occurred during a connection to ricda13-w12vm2.ca.com:7403. SSL peer cannot verify your certificate. Error code: SSL_ERROR_BAD_CERT_ALERT
  The page you are trying to view cannot be shown because the authenticity of the received data could not be verified.

Please contact the website owners to inform them of this problem."

- On Chrome, the following error appears:
  This site can't provide a secure connection ricda13-w12vm2.ca.com didn't accept your login certificate, or your login certificate may have expired.
  Try contacting the system admin.
  ERR_BAD_SSL_CLIENT_AUTH_CERT

Solution:

Ensure that the CAC root and intermediate certificates are added to the Java keystore or trustStore.

> **NOTE**
>
> For more information, see Configure CA SOI for new Smart Card.

Service Modeling

This sectioncontains information about planning, building, and managing service models in CA SOI. Itintroduces the key service concepts and provides detailed procedures, processes, examples, and scenarios about service modeling.

This section also includes details about creating policies that automatically create and maintain services and relationships between source and target CIs according to specified criteria. Additionally, you can find information about how to create, monitor, and manage service-level agreements.

See the following topics for details:

Intended Audience

This section is intended for product administrators who are responsible for modeling and managing services in CA SOI.

Service Modeling Introduction

This section introduces the concepts, best practices, and planning recommendations for service modeling. For more information about basic CA SOI concepts that relate to services, see CA SOI Terminology and Concepts.

Service Models

Service Models are the basis of all administrative and management functions in CA SOI. Service models represent high-level abstract entities; for example, a web-based retail transaction service, a printing service, and a routing service. Service models help you manage your enterprise from the perspective of business services, providing consolidated, holistic, and realistic views of all IT resources.

Each service model consists of the following entities:

- A CI that represents the service itself.
- Child CIs that represent IT elements that support the service or measure and manage some aspect of its behavior.
- Relationships that determine how CIs interact and depend on one another.
- Propagation policies that determine how impact from a CI fault condition propagates to related CIs.

Service Model Concepts

Contents

The concepts in this section help you gain an understanding of how CA SOI monitors services using relationships and propagation (aggregate, bound, custom, and operative) to calculate service impact. The following entities play a role in impact calculation:

- Alert and CI conditions from domain managers display as severity in the appropriate CI and alert.
- CI severity affects related CIs in service models according to relationships and the propagation type and policy settings.
- If propagation settings cause severities to propagate, the multiplied value of severity and significance determines the related CI impact and, ultimately, service impact.

See also:

- Health, Quality, and Risk
- Granularity

Relationships

Relationships in a service model show how CIs are linked to form the service topology.

A relationship between linked objects has a semantic (name or type, for example 'HasAccessTo') and a propagation type (for example, 'Custom'). CA SOI relationships correspond to instances of USM BinaryRelationship type. If you import a service from a connector, its relationships in the domain manager are mapped to USM relationships.

You can assign relationships to every link between objects in a service model. To assign relationships, you select the appropriate propagation type and then select a relationship from the list of relationships that map to that propagation type.

The available USM BinaryRelationship semantics are as follows:

> **NOTE**
> For details about each type or information about relationship updates, see the USM schema documentation.

- **Has Access To**
  Specifies that a CI accesses another CI's functionality. Use this relationship to indicate that a resource can access another resource, or that software communicates with or accesses a specific entity, such as a database. This relationship differs from Has Requirement For, which indicates a mandatory presence of the target CI for the source CI to function.
- **Has Contact**
  Specifies that a CI plays a specific contact role for another CI, such as an owner or assignee.
- **Has Detail**
  Specifies that a CI provides additional information for another CI. For example, an Asset CI can relate to another CI that provides more detail about the asset.
- **Has Member**
  Specifies that a child CI is a member of another CI. For example, several User CIs can be members of a user group. This relationship is the default assignment.
- **Has Requirement For**
  Specifies that a CI requires the existence of another CI, its operation, or both. For example, an Application CI can require the operation of an Application Server CI.
- **Is Affected By**
  Specifies that a CI impacts another CI and that custom policy defines the impact. Whereas the Has Requirement For relationship causes impact when the target CI is in a critical or down state, Is Affected By lets you configure the scenarios that impact the source CI. For example, a web server farm group is impacted if 40 percent of its ComputerSystem CIs are down.

NOTE
CIs in maintenance mode are excluded from custom policy calculations that are related to average or percentage.

- **Is Bound To**
  Specifies a symmetric relationship where two CIs are intrinsically linked, so that one cannot function without the other.
- **Is Cause Of**
  Specifies that a ChangeOrder is the cause of a Request, Incident, or Problem being opened.
- **Is Clone Of**
  Indicates that the source CI is a clone of the Target CI and that the two elements are synchronized.
- **Is Composed Of**
  Specifies a compositional relationship where a source CI is the aggregate of several target CIs.
- **Is Connected To**
  Indicates a network connection carrying data between the source and target CIs, such as between physical ports or application components.
- **Is Discovered By**
  Indicates that the specified ManagementAgent target CI discovers and manages the source CI.
- **Is Evolution Of**
  Indicates that the source CI is evolved from the target. Examples include next generations of hardware or software.
- **Is Hosted By**
  Indicates that the target CI hosts the source CI. This relationship is the inverse of Is Host For.
- **Is Host For**
  Indicates that a source CI is hosting a target CI. For example, a ComputerSystem CI can host a VirtualSystem or RunningSoftware CI.
- **Is Impacted By**
  Indicates that another CI impacts or affects the source CI. For example, a Memory child CI or Port child CIs affect the parent ComputerSystem CI.
- **Is Instance Of**
  Specifies that a source CI is an occurrence of a target CI. For example, a ProvisionedSoftware CI can run an instance of a RunningSoftware CI.
- **Is Location For**
  Specifies that a source CI defines the target CI location.
- **Is Manager For**
  Specifies that a source CI controls or manages a target CI. For example, a DatabaseInstance CI can manage a Database CI.
- **Is Request For**
  Specifies that a source CI has been created to create, provision, or otherwise handle the target CI.
- **Is Resolved By**
  Specifies that the specified ChangeOrder target CI resolves or corrects a Request, Incident, or Problem source CI.
- **Is Result Of**
  Specifies that a source CI is created as a result of an automated workflow or manual processing of a target CI.

Relationships display as color-coded arrows between two objects in the Topology view. The Topology view is available on the Operations Console and the Service Modeler. Propagation types of relationships define CA SOI derives CI and service impact.

Propagation Types

Propagation type defines how CA SOI derives the impact when conditions change in related objects.

Every relationship in a service model has a propagation type, and each USM relationship type maps to a specific propagation type in CA SOI. The propagation types are as follows:

- Aggregate
- Bound
- Custom
- Operative

Relationships are depicted as arrows between two CIs in the Topology view. The color of the arrow reflects the propagation type of the underlying relationship. The Topology view is available on the Operations Console and the Service Modeler. The first letter of the propagation type is a label on the arrows (A, B, C, or O for Aggregate, Bound, Custom, and Operative, respectively).

> **NOTE**
> Sometimes the arrows are animated red dashes to indicate the root cause of a situation.

Aggregate Propagation

Aggregate propagation indicates a general-purpose relationship that propagates impact from one CI to another.

Typically, an association between CIs that are part of a higher-level CI uses this propagation type. CA SOI uses the highest impact condition of all aggregate child CIs when calculating the impact on a parent CI.

Usage

Use aggregate propagation in the following situations:

- When child CIs individually affect a parent CI
- When a parent CI contains multiple child CIs that could potentially impact its condition

Examples

Examples of appropriate situations to use aggregate propagation include the following:

- A CPU or Memory CI may have aggregate propagation to the Operating System on their server.
- Services or components that make up an application may have aggregate propagation to their parent Application CI.
- The components of a network (Router, Bridge, Network Interface Card, and so on) may have aggregate propagation to a high-level Network service, CI, or group.

USM Relationships

The following USM relationships map to aggregate propagation by default in CA SOI:

- Is Impacted By
- Is Composed Of
- Is Host For
- Is Location For
- Is Manager For
- Has Member
- Has Access To

You must select one of these relationships to effect the aggregate propagation behavior. For example, you could assign Is Host For to a relationship between a ComputerSystem CI and its hosted RunningSoftware and VirtualSystem CIs. Or you could assign Is Composed Of to a ComputerSystem CI and its compositional parts. When you add services and resources to a service model, the default propagation type is aggregate with a relationship of Has Member until you change it to something else.

Bound Propagation

Bound propagation indicates a bidirectional relationship between two CIs. If one CI is bound to another CI and either CI has a severity change, the change results in the same impact on both CIs.

Usage

Use bound propagation in the following situations:

- When a fault condition that affects one CI equally affects the other CI
- When two CIs are on the same hierarchical level and cannot function without each other

Examples

Examples of appropriate situations to use bound propagation include the following:

- Mirrored disks may require bound propagation with one another.
- A Database CI connected to an Application CI may have bound propagation with a replication database CI that belongs to a related replication server.

USM Relationships

The following USM relationships map to bound propagation by default in CA SOI:

- Is Bound To
- Is Clone Of
- Is Connected To
- Is Result Of

You must select one of these relationships when you assign bound propagation. For example, you could assign Is Bound To to a set of mirrored disks. When bound propagation is automatically assigned to a relationship, the default relationship assignment is Is Bound To.

Custom Propagation

Custom propagation indicates that one CI may depend on several other CIs for some behavior or function.

Custom propagation lets you specify policy that defines when and how to change the severity value of a parent item in a dependent relationship. If the policy is not met, there is no impact to the CI or service.

Usage

Use Custom propagation in the following situations:

- When a group of CIs work together to deliver an aspect of the service
- When a parent CI is not directly affected by a fault condition in an individual CI and is instead affected by the combined performance of all child CIs

Examples

Examples of appropriate situations to use custom propagation include the following:

- A Group CI for a web server farm may require custom propagation with its servers to indicate that the farm can lose 30% of its servers and still function.
- A clustered server configuration with failover capacity may require custom propagation to indicate that a certain number of failures are permitted as long as the failover servers are functioning.

USM Relationships

The following USM relationships map to custom propagation by default in CA SOI:

- Is Affected By
- Is Evolution Of

You must select one of these relationships to effect the custom propagation behavior. The relationships define that the impact from one CI to another is derived through configurable policy.

Custom propagation requires you to create a propagation policy that defines how to calculate the impact from a group of related CIs to a parent CI. You can configure CA SOI to automatically assign custom propagation and the associated propagation policy when a CI is added to a group that already uses custom propagation instead of the default aggregate propagation.

When you assign custom propagation policy to a CI and the threshold is reached, the target CI attains the specified severity value and an infrastructure alert with the appropriate severity. However, because the alert applies to the CI as it relates to the parent service, the CI icon does not change color as expected, and the severity propagates to a separate impact calculation on the service, which is reflected in the service icon color.

Operative Propagation

Operative propagation indicates that the related item is affected only if the impact value of a CI exceeds a defined threshold.

Operative propagation lets you specify policy that defines the impact value that the CI must exceed for impact to propagate to the parent CI. If the policy is not met, there is no impact to the CI or service.

Usage

Use operative propagation in the following situations:

- When a CI is only needed to be available, even if it is not performing well
- When the service is not affected unless the state of the CI is serious
- When you are not concerned with minor state changes as long as the CI is still running

Examples

Examples of appropriate situations to use operative propagation include the following:

- An application server may have operative propagation with a database server; the application server requires the database server to provide data storage and retrieval functions. As long as the database server is up, it is assumed that the application server is being served adequately to perform its function.
- An operating system may have operative propagation with the CI for its memory. In this case, operative propagation can prevent impact from the memory CI from propagating to the operating system unless the memory usage exceeds a defined threshold.

USM Relationships

The following USM relationships map to operative propagation by default in CA SOI:

- Has Requirement For
- Has Contact
- Has Detail
- Is Instance Of
- Is Request For
- Is Cause Of
- Is Discovered By
- Is Hosted By
- Is Resolved By

You must select one of these relationships to effect the operative propagation behavior. For example, you could assign Has Requirement For to an Application CI that requires an ApplicationServer CI to be running.

By default, operative propagation requires an impact value of 20 to propagate impact, and the default relationship is Has Requirement For. You can create propagation policy to define a different impact threshold.

Configure Default Propagation Policies

Each relationship type has a predetermined propagation type. You can view the relationship mappings and change the parameters of default propagation policies for each relationship that maps to custom or operative propagation.

> **WARNING**
> When you change the default propagation policy parameters for a relationship type, any instance of that relationship in a service model that has not been manually overridden acquires that policys parameters by default.
>
> The changes will affect all new relationships/propagations from that moment on, but will not cause the reevaluation of the policies currently in SA Managers memory. To effect that, you have to restart the CA SOI Application Server.

You must close the Service Modeler to change default propagation policy.

> **NOTE**
> A propagation policy determines how impact is derived and propagated. For more information about each policy type, see Define Operative Propagation Policy and Define Custom Propagation Policy.

Follow these steps:

1. Select Tools, Propagation Policies from the Operations Console.

   > **NOTE**
   > If the Service Modeler is currently active (in any user session), a warning dialog opens stating that the Default Propagation Policy dialog is read-only. Close all Service Modeler sessions and reopen the dialog to make changes.
2. Click the set link next to the value in the Propagation Rule column for the Is Affected By or Is Evolution Of relationship row, which have custom propagation defined..
3. (Optional) Select the Automatically Maintained check box to automatically change the relationship to Is Affected By or Is Evolution Of and apply the custom propagation policy when new CIs are added to a group that uses custom propagation. Auto-maintenance applies to relationships added by both manually editing a service and through automation by service discovery or service import.

   > **NOTE**
   > You can set Automatic Policy Maintenance for the Modeler operation to on by default in the Set Preferences dialog.
4. Select one of the following options from the Policy Type drop-down list:
   - **Average**
     Sets the impact of the parent item based on the average impact values of CIs associated with the policy.
   - **Percentage**
     Sets the impact of the parent item based on a percentage of CIs that have the impact specified in the rule.
   - **Any**
     Sets the impact of the parent item when any CIs associated with the policy have the impact specified in the rule.
   - **All**
     Sets the impact of the parent item when all CIs have the impact specified in the rule.

   Default: Any
5. Complete the following fields and settings for each rule that you want to create:
   - **% of items**
     (Percentage type only) Defines the percentage of items that must exceed the impact threshold to meet the rule criteria.
   - **Threshold**

Defines the impact threshold. The appropriate amount of CIs (as defined by the policy type) must meet or exceed the impact threshold to meet the rule criteria. Define a number between 0 and 40. The impact numbers translate to the following impacts:
- 0: Operational
- 1-10: Slightly Degraded
- 11-20: Moderately Degraded
- 21-30: Severely Degraded
- 31-40: Down

– **Set Severity**
Defines the severity to assign to the parent item if the rule criteria are met.

The text for each rule changes to reflect the new settings. Define as many of the four available rules as the policy requires.

6. Click OK.
The rule changes in the Propagation Rule column for the Is Affected By or Is Evolution Of relationship. You can assign different default policy parameters for each relationship type. The new default custom policy is assigned to all new custom propagation assignments and existing custom propagations that use the associated relationship, including those that previously used a default custom policy. Customized default propagations are not recalculated; they take effect from the point they are customized.

NOTE
CIs in maintenance mode are excluded from custom policy calculations related to average or percentage.

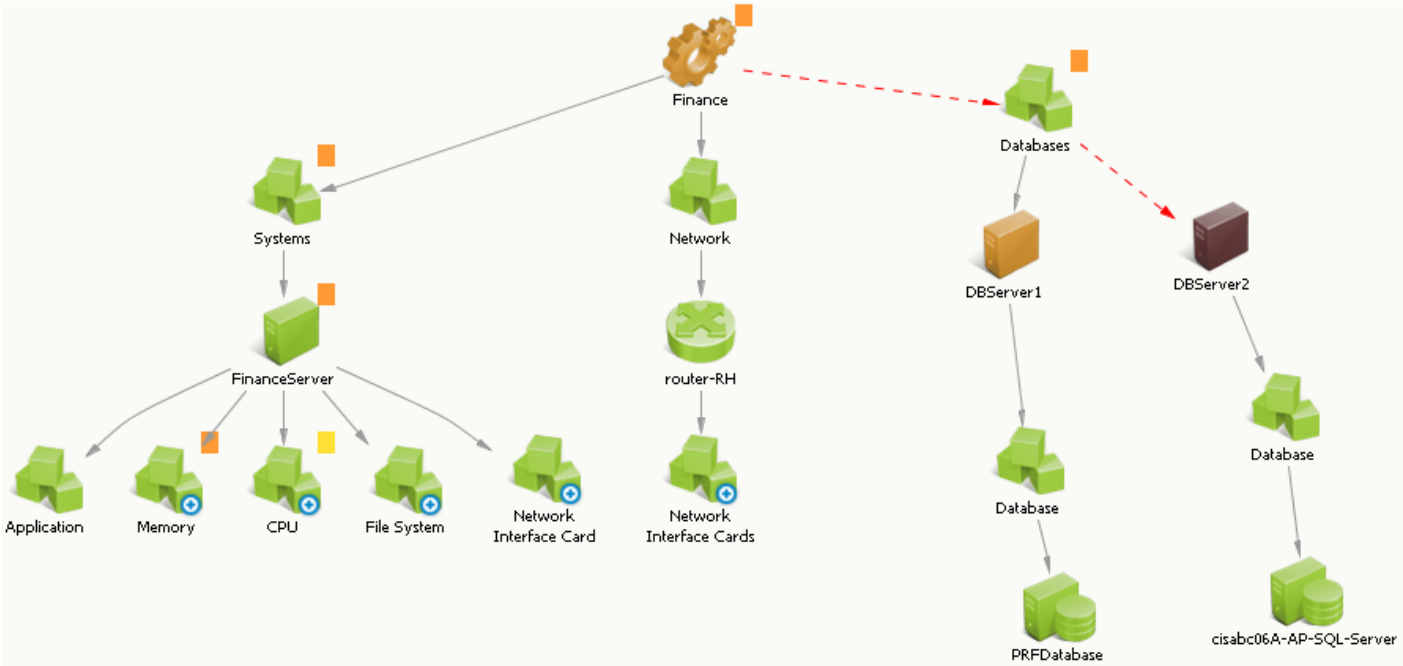
7. Click set next to any value in the Propagation Rule column for relationship rows that map to Operative propagation.
8. Set the default threshold value at which impact starts to propagate to parent CIs, and click OK.
9. Click Save.

Changes take effect for ensuing alerts by default. To recalculate the impact of existing alerts, restart the CA SOI Manager and CA SOI User Interface.

Service Model Types

The most common types of service models are as follows:

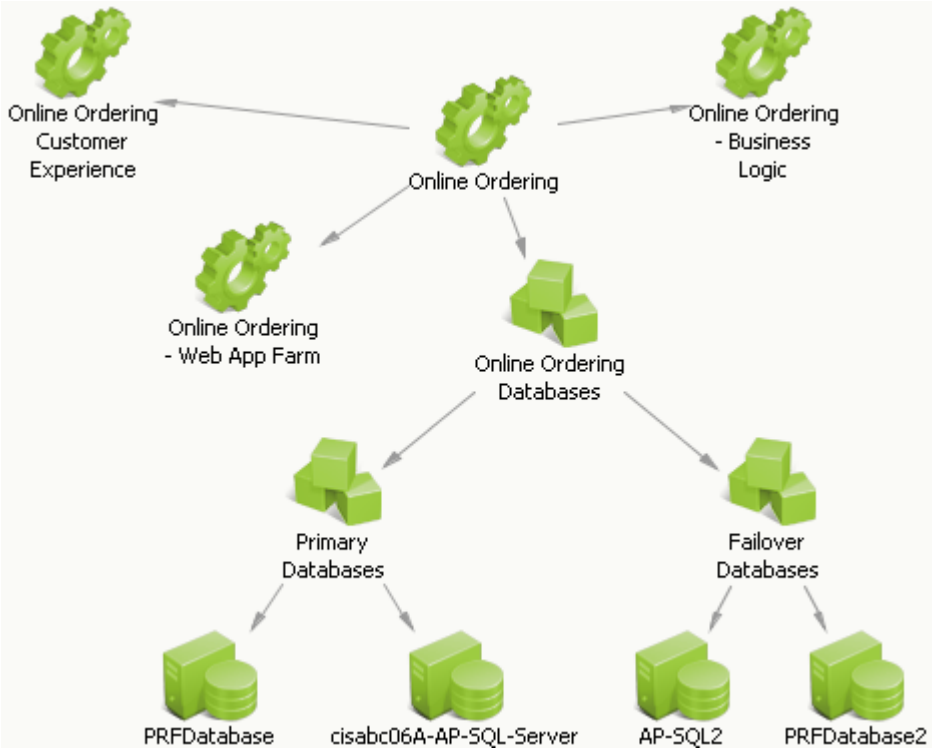
- **Bottom-Up**
Bottom-up service models are constructed from a domain-oriented perspective (network, systems, applications, and so on). These models take less time to define and are easier to understand. They focus on impact analysis, not root cause, and are the most common starting point. Most service models that are imported from domain managers are bottom-up, because they define a service from the perspective of their managed domain.
The following graphic shows an example of a bottom-up service model:



Notice that this bottom-up example is modeled based on groups that define distinct domains: Databases, Network, and Systems. Separate products can manage these domains, and the model combines the intelligence of these domain managers to create a service representation for a Finance department. The model uses groups to aggregate domains and specific objects within those domains (such as the network cards in a router). Note how the model is able to pinpoint the root cause of Finance service degradation as the database server DBServer2.

NOTE
For a detailed scenario that builds on this example, see Service Modeling Examples and Scenarios.

- **Top-Down**
  Top-down service models are organized based on a logical business service topology, not by domain. These models are more advanced. They require a better understanding of the logical structure of your enterprise and of the capabilities of service modeling in CA SOI. They ultimately provide better root cause information than bottom-up models.



This example uses subservices to provide a solid representation of the business-oriented topology of the service. The items in the specific subservices may or may not be specific to domains. Impact still propagates from subservices to the top-level service.

Federated Modeling

A f*ederated modeling* approach indicates that intelligence is distributed across integrated products.

The network management tool has intelligence around the network model, the application management tool has intelligence around applications and transaction models, and so on. CA SOI combines the collective intelligence of all integrated products to derive the overall impact of any CI condition on modeled services.

The benefits of federated modeling are as follows:

- Distributed models are scalable.
- Models in each environment are optimized to suit their role within that environment.
- Leveraging existing models lowers the cost and effort of implementation.
- Models are flexible and extensible to meet the specific detail level that each service requires.

Consider the following example of federated modeling in practice:

- A service in CA SOI models an application and its dependency to a host.
- CA Spectrum models the same host and its dependency to the network and systems infrastructure.
- When a change to the operational state of the infrastructure impacts the host, CA SOI is notified through its CA Spectrum connector. CA SOI can identify the root cause of the problem using the intelligence that CA Spectrum provides without having to model the underlying infrastructure and topology in the service.

Planning Service Models

Contents

Service models are the basis of all administrative and management functions in CA SOI. Managing your enterprise from the perspective of business services moves management away from domain-specific management to consolidated, holistic, and practical views of all IT resources. To accomplish this comprehensive business service management, first gain an accurate picture of your enterprise and what resources make up discrete business services.

This section describes the information you must collect to model accurate, end-to-end business services using the Service Modeler.

Service Identification

Any set of resources that depend on each other to provide a useful business function can compose a service. Identifying services requires an understanding of the resources that make up your enterprise and the typical contents of a service. A service can take many forms, including the following:

- A low-level IT service such as DNS or DHCP
- A set of resources that you want to manage collectively, such as the servers or printers contained in a specific location
- A network service such as VPN
- A systems service such as Active Directory
- A database service such as a Microsoft SQL Server cluster
- A high-level business service such as Payroll, Email, and so on
- External services such as an Internet Service Provider or Cloud

Most services are a combination of all of the above, with high-level business services containing various low-level services. For example, a Blackberry business service may contain subservices that represent key IT services that support Blackberry communication, such as Active Directory, Exchange, DHCP, and so on.

Most modern, complex service models do not, by definition, follow a hierarchical tree-like structure. They follow a more dynamic, multi-layered structure that requires you to collect comprehensive information about what makes up your services. As you begin identifying the services in your enterprise that require modeling, you can consult the following sources for the required information:

- Department leads and line of business owners
- Your IT department
- Existing management products
- Old design documents, statements of work, and so on
- Service Catalog, defined workflows
- Application configuration management tools
- Existing help desk or CMDB products

Other sources may also be useful. In a large enterprise, you rely on all available sources of information to gain an accurate picture of your services.

Existing Services

You may have other management products that manage services or a concept similar to services. You can import services from many integrated products into CA SOI as full service models. Examine your existing management products for services so you can leverage your investment in these products and avoid having to model a similar service from scratch.

Imported services are synchronized with the source definition when the source changes, and are reconciled into a single service where they have been modeled within multiple domains.

For more information about the connectors that support importing services from their domain manager, see the product-specific *Connector Guide* included with each connector package.

Resource Collection

As you define the services in your enterprise, you also define the specific resources that compose those services. Services can contain everything from applications and databases to servers and network devices. If some resources are omitted from a service model, you cannot ensure that the model is an accurate representation of the service health and availability.

Consider the following as you collect the resources that comprise your services:

- Find out where all required resources are currently managed. Verify that all resources in domain managers for which CA SOI provides connectors are actively managed, so that you can include the resources in service models. Consider a scenario where an application for which CA SOI does not provide a connector manages the resources. In this case, you can build a custom integration with the application using various tools provided.

  NOTE
  For more information about implementing provided connectors and building custom integrations, see the *CA Catalyst Implementation Guide* or *Connector Guide* for each specific connector.
- Organize the resources that make up a service into logical subcomponents. Most complex services are sets of subservices or groups. If you can define these sets before modeling the service, you can improve the presentation and performance of your service. You can also reuse these modular components across multiple services.
- Consider the relationships among resources within subcomponents and the service at large. Try to separate subcomponents based on the following high-level relationship types:
  - Resources where the only concern is if a component is providing functionality
  - Resources that are critical to the functioning of a service
  - Resources that have some association or form a group, where the resources in the group can be operated on collectively
  - Resources that are intrinsically dependent on each other

  NOTE
  For more information about relationships and how impact is propagated for each relationship, see Service Model Concepts.
- For basic collections of resources or for areas of your enterprise with a high level of volatility, organize resources and note the criteria by which you can group them. Resources that fall into these categories can be good candidates for service discovery policies that automatically create services and relationships based on policy criteria.

Resource Importance

You must determine the importance of services and the resources in a service. Rely on sources of information such as business continuity plans, disaster recovery plans, and service-level agreements to obtain the services that are most important to your enterprise.

Typically, more information is available about business-critical services, making them the ideal place to start service modeling. Not all services require the same level of complexity. A critical business service may be documented in enough detail for you to create a comprehensive service model, while for a less critical service, a simpler model will suffice. CA SOI provides the flexibility for models of different detail levels, and you can make models more comprehensive as your investment level in the product grows.

Also determine the importance of the resources that make up a service, so that this importance is accurately reflected when issues occur.

Best Practices

As you begin modeling services in CA SOI, consider the following best practices:

- Determine and start with the services that matter the most to your enterprise. The most important services typically contain the most supporting documentation.
- Organize resources into groups and subservices wherever possible to increase the modularity of your service. You can reuse groups and subservices across multiple services. Logical groupings make it easier for you to create a service model whose layout, root cause, and service impact is easy to comprehend.
- Start with bottom-up service models, either imported from domain managers or modeled organically, and move to top-down in a phased approach.
- Take advantage of federated modeling concepts, which let you start with relatively simple service models and still obtain root cause information using the distributed intelligence of integrated products.
- Take advantage of low granularity modeling, so that less detailed models can automatically aggregate alerts from related unmodeled CIs and immediately return results similar to those of more comprehensive models.
- Take advantage of service discovery to automatically add services or create relationships based on criteria that define logical groups of resources and object relationships in your enterprise. Services created by service discovery update dynamically according to changes in your enterprise.
- Gradually move to more comprehensive models as your usage of the product grows.
- Even as your service models become more detailed, keep them as lightweight as possible so that they include only the essential information to support reconciliation, service impact, and root cause.
- As you build the service model, save or validate often, so that potential errors are easier to resolve.

Navigate the Topology View

Contents

As an administrator or an operator, you can view, collapse, or expand the Topology.

The Topology view is a graphical representation of the relationship among services and the devices that support them. Icons represent the object type, and arrows and the position of icons represent the relationships. CA SOI highlights the selected Object on the Services tab with small boxes.

> **NOTE**
> The Topology view is available in the main Contents pane and in the Service Modeler window. Some toolbar buttons described in this section do not appear on both windows.

Follow these steps:

1. Open the Operations Console, and perform one of the following actions:
   - Select a service from the Services tab in the Navigation pane and click the Topology tab in the Contents pane.
   - Select Tools, Create New Service.
   - Right-click a service from the Services tab in the Navigation pane and select Edit Service.

   One or more icons on the Topology pane represent the service.
2. Use the toolbar buttons as necessary to complete the following actions:
   -

     (Pan Tool)

     Moves the topology up, down, right, and left when you click the tool and drag on the screen.

     > **NOTE**
     > You can use the mouse wheel to zoom in and out.
   -

     (Select Tool)

Displays details about a service or resources when you click the tool then click a service or resource in the right pane. The details appear in the Component Detail area under the Topology.

–
(Interactive Zoom Tool)

Enlarges or reduces the topology when you drag the tool on the screen. Other zoom buttons include the Marquee Zoom Tool and Zoom Level Control.

NOTE
You can also zoom by using the mouse wheel.

–
(Link Navigation Tool)

Displays relationships to and from objects when you click the tool and mouseover the object. The tooltip describes the relationship. For large topologies, click the relationship link to pan to the linked object at the other end.

–
(Marquee Zoom Tool)

Increases the magnification in a specific region when you click the tool and select a region in the right pane. Other zoom buttons include the Interactive Zoom Tool and Zoom Level Control.

–
(Relationship Tool)
(Service Modeler only) Specifies the type of relationship to create between objects when you click the tool and select one of the available relationships. Once you select a new relationship, all new objects obtain that relationship type.

–
(Adjust and View buttons)
(Contents pane only) Rearranges the topology when you click the Adjust button and drag items. Click Save when you are finished to save the changes. Select the default option button View to disable further changes.

NOTE
You can also adjust the layout while in View mode when you click Apply Automatic Layout and select a layout from the drop-down list.

–
(Save Topology Layout)
(Contents pane only) Saves topology changes.

–
(Perform Service Validation)
(Service Modeler only) Verifies that a service is complete and correct.

NOTE
Automatic validation occurs after every change to the service.

–
(Apply Automatic Layout)

Changes the type of chart when you click the button and select one of the following options:

- **Circular**
  Emphasizes clusters that are present in a network topology.
- **Grid**
  Arranges objects in horizontal rows and vertical columns.
- **Hierarchical**
  Emphasizes relationships among objects by placing them at different levels. The layout is like an organizational chart at a company, which is the default when you build services from scratch.
- **Orthogonal**
  Minimizes bend points by arranging objects horizontally and vertically, at 90 degree angles.
- **Symmetric**
  Emphasizes symmetries that are present in a network topology, which is the default for newly discovered services. Symmetric is also the fastest and yields the smallest topologies.

–

(Refresh Layout Contents)
(Contents pane only) Updates the service topology according to recent changes. The topology may require a refresh to reflect the current service topology if the SA Manager has a high processing load, a shutdown of the SA Manager interrupted processing, or a heavy volume of import events are being processed. When you click the button, you get a confirmation message that states one of the following:

- No topology updates were necessary
- All necessary topology updates are complete

–

(Delete Selected Topology Objects)
(Service Modeler only) Removes objects from the service when you select them and click the button.

–

(Straighten Selected Edges)
Removes bends in the links between items when you select a wavy line and click the button. This button is available in the Service Modeler, and in the Contents pane when you click the Adjust option button.

–

(Undo Last Action)

Discards topology changes.

–

(Redo Last Action)

Repeats your last action.

–

(Chart Complexity Level)
Displays the type of relationship among objects when you click the button and select Advanced. Simple is the default in the Contents pane, and Advanced is the default in the Service Modeler. When you select Advanced, the arrows between objects are color-coded and contain the first letter of the relationship type. When you select

Advanced with Names, the arrows between objects are color-coded and contain the full name of the relationship type.

–
(Filter Configuration Item Condition Visibility)
(Contents pane only) Emphasizes severities when you click the button and select a severity. Items with severities lower than the one selected are dimmed. For example, if you select Major, items with Normal and Minor severities are dimmed.
For more information, see Severity.

–
(Change Relationship Visibility)

Hides the links between objects when you click the button and select Hide ALL from the drop-down list. Show ALL is the default.

You can select specific relationships (aggregates, bound, custom, and Operative) in the main Contents pane and in the Service Modeler window.

–
(Toggle Item Highlighting)
(Contents pane only) Enables or disables synchronization with the Component Detail pane. By default, details for the selected item in the topology are displayed, but you may want to turn it off to increase performance when you adjust or navigate the service topology.

–
(Show/Hide Item List Pane)

Displays or removes a table of details beneath the Topology pane.

–
(Zoom Level Control)

Specifies the amount of magnification. Other zoom buttons include Interactive Zoom Tool and Marquee Zoom Tool.

–
(Toggle Overview Window)

Displays a small view of the topology. This window is useful when you want to change the region or zoom level of the topology view.

You can use the following shortcut keys to quickly switch between tools when the Topology pane is active and one of the tools is already selected:

- P: Pan Tool
- S: Select Tool
- I: Interactive Zoom Tool
- L: Link Navigation Tool
- Z: Marquee Zoom Tool

Collapse or Expand the Topology View

If a service is complex, you can show fewer items by collapsing child objects in the Topology view of the Contents pane or the Service Modeler.

Follow these steps:

1. Open the Operations Console, and complete one of the following actions:
   – Select a service from the Services tab in the Navigation pane, and click the Topology tab in the right pane.
   – Select Tools, New Service.
   – Right-click a service from the Services tab in the Navigation pane and select Edit Service.

   NOTE
   The Topology view can take several seconds to load.

   One or more icons represent the service on the Topology pane.

   - Right-click an item with child objects, and select Collapse/Expand, Collapse.

     NOTE
     An alternative way to collapse is to press Shift+click over the parent item.

     The child items are no longer visible. The parent item has a small plus icon (+) in the lower right.

   To expand the Topology view

   Right-click a parent item that is displayed with the + icon, select Collapse/Expand, and select one of the following options:

   - **Expand**
     Opens all child items. (Expand performs the same function as Shift+click.)
   - **Expand one level**
     Opens the next level of child items. (Expand one level performs the same function as double-clicking the plus (+) icon.)

     NOTE

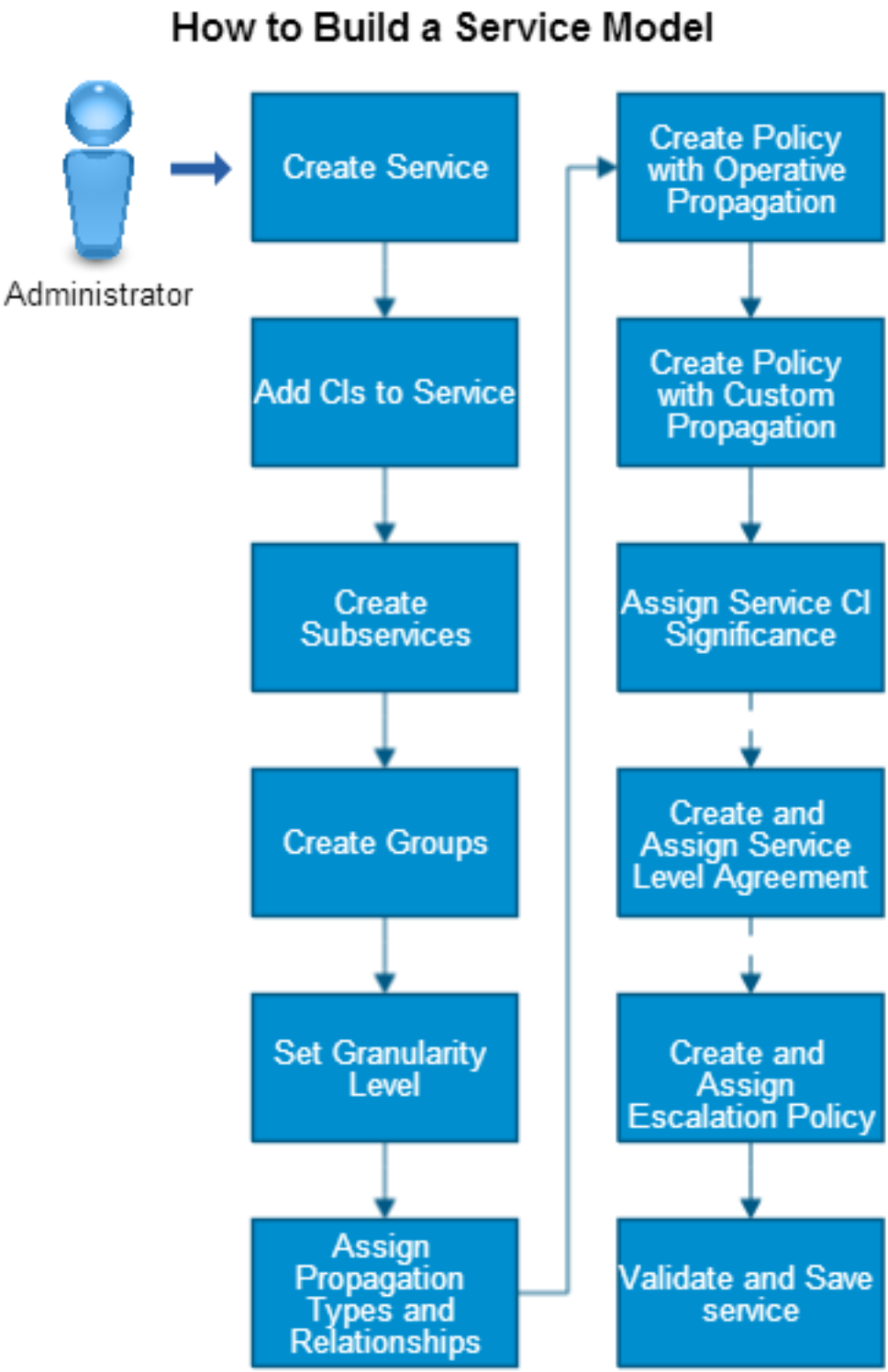
     - You can undo (Ctrl + Z) or redo (Ctrl + Y) a collapse or expansion.
     - You can save a collapsed or expanded view in the Service Modeler. You can save it in the Contents pane when you select the option button Adjust and click the Save icon. When the View option button is selected, you cannot save the layout.

How to Build Service Models

As an administrator, you build service models in CA SOI. Building service models is a multi-tiered process that lets you define the associated CIs, relationships, and propagation policies, and optional service-level agreements and escalation policies. You can add any CI that an integrated product manages to a service in CA SOI. You can also set the granularity level, which determines whether CA SOI aggregates alerts from child CIs not included in the model.

Use this scenario to guide you through the process:

Figure 29: how to build a service model



1. Create a Service.

2. Add CIs to the Service.
3. Create Subservices.
4. Create Groups.

   NOTE
   Subservices and groups act together to deliver specific aspects of the service.
5. Set the Granularity Level.
6. Assign Propagation Types and Relationships.
7. Create a Propagation Policy with Operative Propagation.
8. Create a Propagation Policy with Custom Propagation.
9. Assign the CI Significance in the Service.
10. (Optional) Create and Assign a Service-level Agreement.
11. (Optional) Create and Assign an Escalation Policy.
12. Validate and Save the Service.

NOTE
You can perform several operations while modeling a service to customize the service display. For more information about these operations, see How to Customize Service Model Display. You can also associate services with customers to determine how services impact the associated customers. For more information about customers, see the How to Create and Manage Customers.

Create and Configure the Service Model

Contents

As an administrator, you create the service model, then assign the configuration items (CIs), and optionally create subservices.

Create a Service

You create and edit services in the Service Modeler.

Follow these steps:

1. Open the Operations Console and select Tools, Create New Service.
   The Service Modeler opens. An icon labeled New Service appears on the Topology tab. A high-level list of classes appears on the Objects tab in the left pane.
2. (Optional) Click the New Service icon and move it to a different location on the Topology grid.
   This may be necessary for services that will contain many resources.
3. Double-click the icon and rename the service.
4. Click Save, OK.
   The service is saved.

Add CIs to the Service

You add CIs to services to define the service content. You can add any CI that is managed by a running connector to a service. The Service Modeler provides all CIs available for inclusion in services in the Objects tab. If you do not see an expected CI on the Objects tab, verify that the source domain manager is actively managing the resource.

NOTE
You can also add resources not managed by a running connector using the Sample connector. This method lets you test connector functionality, and potentially spur discovery and management of created entities in domain managers.

The Service Modeler provides simple drag-and-drop functionality for quickly adding CIs to a service. When you add a CI, it automatically establishes a relationship with its parent CI, whether that is the top-level service CI, a group, or another CI.

Consider the following as you add CIs to a service:

- If you are adding CIs that you want to be a part of a subservice, you must add them directly in the subservice. You can add CIs to a group when you create the group in the main service.
- Consider whether to make your service model bottom-up or top-down. For bottom-up service models based on specific domains, you can add services imported from a domain manager to the service model as subservices.
- Add only the necessary level of detail. Federated modeling intelligence lets you create simpler models (for example, a ComputerSystem CI without its associated ports, CPU, memory, and so on). You also may be interested in service impact only up to a specific CI level, or you may want to start with a simple service model and build its level of detail over time.
- You can customize the modeler display to best suit your service size, hierarchy, and other factors. For more information, see How to Customize Service Model Display.

Follow these steps:

1. Do any of the following to locate CIs in the Service Modeler:
   - Select View, Locator.
     The Locator dialog opens. You can search for CIs based on property values. Right-click a CI in the search results and select Add to Modeler to add the CI to the service.
   - Enter text by which to filter the available CIs in the Filter field.
     The Objects tab displays only the CIs that meet the filter criteria in the containing folders.
   - Select the browsing criteria in the Browse By drop-down list.
     The Objects tab sorts CIs by the criteria you selected. You can browse by the following classifications:
     - **Top-level Classes**
       Displays a list of folders sorted by high-level classes such as ComputerSystem, Group, Router, Service, and so on. The CIs in these folders may include child components in subfolders.
     - **All Classes**
       Displays a list of folders sorted by class name.
     - **Data Source**
       Displays a list of folders sorted by the source connector name. If a CI exists in three different integrated domain managers, it appears in all three data source folders.
       Each connector entry contains a five-digit ID number defined by the USM schema. For information about viewing the connector identification number, see View Connector Status in a High Availability Environment. For more information about the connector IDs, see Connector Identification Numbers.
     - **Monitored Services**
       Displays a list of existing services. You can drag entire services into the service as subservices or expand the service to find included CIs.
     - **Monitored Groups**
       Displays a list of existing groups. You can drag entire groups into the service or expand the group to find included CIs.

     The number displayed next to each folder only reflects the top-level CIs in the folder. It does not include CIs contained in subfolders.
2. Select a CI to include in the service.
   The following tabs below the Objects tab display information about the CI:
   - **Information**
     Displays a subset of basic CI properties.
   - **USM Properties**
     Displays the full list of USM properties for the CI.
   - **USM Notebook**

Displays a comparison of the USM properties taken from all managed domains and the USM reconciled sheet, which reconciles the properties from all managed domains into a single set of properties.

3. Do any of the following to add the CI to the service:
   - Drag the selected CI in the service topology.
     The CI is added to the service. If you place the CI in a blank area of the topology pane, it becomes associated to the top-level service CI. To associate the CI with a different CI, drag it on top of that CI.
   - Right-click the CI and select Add with sub-components.
     The CI and all child CIs are added to the service model associated with the last CI selected on the topology pane. By default, child CIs are not added to the service when you drag a parent item into the topology. If you always want to add child CIs when you drag a CI into the topology, you can change this preference in the Operations Console.
   - Select multiple CIs with Shift+click or Ctrl+click and drag them in the topology pane in one of the following ways:
     a. Do not lift the left mouse button after making the last selection. Keep the button pressed and drag the CIs.
     b. Drag the selected items with the right mouse button.

     The CIs are added to the service model associated with the top-level service CI or the CI on which you dragged the items.

   All CIs added to a service obtain the default Has Member relationship and aggregate propagation with connected CIs, unless you add them to a group with automatic policy maintenance enabled.

Create Subservices

Services can have any number of subservices. Subservices let you group CIs and relationships in a modular fashion. Create a subservice for any set of CIs that can operate as a standalone service and that you may use in other services. For example, an Email service may contain subservices for key functions such as Exchange, Active Directory, and so on.

You can reuse created subservices in multiple services, and you can use an existing service as a subservice in other services. Subservices are important entities in top-down service models, where modeled services accurately represent the overall business service topology. Most top-down service models contain multiple subservices.

Follow these steps:

1. Open the Operations Console, right-click the service in the Navigation pane that needs one or more new subservices, and select Create Sub-Service.
2. Add CIs to the service.
3. Follow the rest of the applicable steps in How to Build a Service Model.
   The subservice is created. When you reopen the parent service, the subservice appears connected to the top-level service CI with a default relationship of Has Member and propagation type of aggregate.

Follow these steps: (existing service as a subservice of another service)

1. Open the Operations Console, right-click the top-level service to which you want to add an existing service as a subservice, and select Tools, Edit Service.
2. Select Monitored Services from the 'Browse by' drop-down list.
3. Select the service to include as a subservice and drag it into the Topology pane.
   The service is added as a subservice connected to the top-level service CI (if you dragged to an empty spot in the Topology pane or directly onto the service CI) or to the CI on which you dragged the service with a default Has Member relationship and aggregate propagation.

   NOTE
   If you add a service CI defined within a domain manager as a subservice, CA SOI does not automatically include all CIs associated with that service in the domain manager unless that service has been fully imported.

Create Groups

Groups are intermediate CI objects that collect CIs and relate them to each other by some role or function.

Groups are often used to leverage the same custom propagation policy for a set of CIs, or to funnel several aggregate propagation types into one top-level CI. Group CIs do not commonly have a severity because they are not managed by a domain manager, but the impact of the CIs they contain is considered in how the group CI impacts other CIs and the service.

Follow these steps:

1. Create a service or modify a service.
   The Service Modeler opens.
2. Do one of the following to create a group:
   – Select one or more CIs, right-click anywhere on the Topology pane, and select Group, Create.
     The group is created as a parent of the selected CIs. The group establishes a default relationship of Has Member and propagation of aggregate with each connected CI.

     NOTE
     The main service cannot be included in a group.
   – Verify that no CIs are selected, right-click a blank area in the Topology pane, and select Group, Create.
     The group is created without relationships to any CIs, including the service CI.
   – Right-click an unselected CI and select Group, Create.
     The group is created as a child of the right-clicked CI. Use this option to create a group as a direct child of the service CI.
3. Double-click the group and give it a unique name.
4. (Optional) Drag CIs not already included in the service model onto the group CI.
   The CIs are added to the group.
5. (Optional) Select one or more CIs that are already included in the service model, right-click the unselected group, and select Group, and one of the following actions:
   – **Add**
     Creates Has Member relationships and aggregate propagation types from the group to all selected CIs. No existing relationships with other parent CIs are removed. Use this operation to maintain any existing associations the CIs may have in the service model while also adding them to the group.
   – **Move**
     Creates Has Member relationships and aggregate propagation types from the group to all selected CIs. Existing relationships with other parent CIs are removed. Use this operation to move CIs from one place in the service to the group.

Set the Granularity Level

You can change the granularity level to determine if you need to include all resources in the model for their alerts to be shown as impacting the service.

Consider the following:

- You can only change the model granularity of a service you are currently editing.
- You can change the default service model granularity for subsequently created services on the Global Settings page of the Administration UI.

Follow these steps:

1. Right-click a service in the Service Modeler.
2. Select Change Granularity and one of the following.

- **Normal** - A normal granularity service model does not aggregate alerts from child CIs not included in the service model.
- **Low** - A low granularity service model aggregates alerts from child CIs not included in the model.

  NOTE
  You cannot include an unmodeled CI in any service if the CI is to impact a modeled CI in low granularity mode. This is because the alerts for that CI are attached to an existing model and the low granularity level looks for unmanaged alerts.

3. If a confirmation dialog opens, click Yes.

A low granularity service detects whether unmodeled child CIs exist that are related to modeled CIs and immediately begins propagating impact from those CIs. You can tell when a service is low granularity by the following indicators:

- An L character appears in the Information column of the Services tab.
- Granularity displays as LOW in the service model tooltip on the Topology tab.
- Granularity displays as Low in the Information tab when you select the service.

When CA SOI detects that a low granularity service contains CIs with unmodeled child CIs and associated alerts that are propagating impact, a

icon appears next to the parent CI to indicate this fact. The color of the parent CI itself does not reflect the impact of the alerts; instead, the impact is propagated upwards on the CI's parent, and ultimately, the service.

Create Propagation Policy and Assign Types

Contents

As an administrator, you create an operative or custom propagation policy and assign the propagation types, significance, and relationships.

Assign Propagation Types and Relationships

Each relationship type (and instance) has one specific propagation type. Selecting a relationship type implies a specific propagation type. When you create a service model, the default Has Member relationship (with aggregate propagation) is established between all services and CIs. If you want to change the relationship type, CA SOI lets you invoke the change relationship action from the toolbar or the shortcut menu.

Follow these steps:

1. Click the Relationship tool

   on the Service Modeler toolbar.
   A drop-down list appears with all propagation types available for selection.
2. Select a propagation type.
   The drop-down list expands to display all relationships that are mapped to the propagation type that you selected.
3. Select a relationship.
   The pointer in the Topology view changes to a wand.
4. Click the parent object in the propagation and relationship you want to change, and drag to the child object.
   The Relationship Edit dialog opens if you are replacing an existing relationship.
5. Click OK on the Relationship Edit dialog if necessary.

NOTE
If you need to straighten the line between the two objects, click the *Straighten selected edges* tool

.

The new relationship and propagation appear connecting the related items as follows:

- If you have selected the *Advanced* 'Chart display complexity level' option (default), a letter indicates the propagation type (A for aggregate, B for bound, C for custom, O for operative) followed by a colon (:) and the relationship significance value.
- If you have selected the *Advanced with names* 'Chart display complexity level' option, the relationship name appears followed by a colon (:) and the relationship significance value.
- For either option, the line connecting each set of related CIs is color-coded to indicate the propagation type (blue for aggregate, burgundy for bound, orange for custom, and pink for operative).

6. Continue changing to the same propagation and relationship, if necessary. For a different propagation and relationship, click the Relationship tool again and select from the drop-down list.
7. Click another tool when you are finished.
   The Relationship tool is deactivated and the previous relationships are replaced.
8. Click Save, OK to exit Service Modeler.

To change relationships and propagation using the shortcut menu

1. Click a child object in the Service Modeler to select it. To select multiple children of the same parent, hold down the Ctrl key and click them.

   NOTE
   Do not select the parent object.

   The child objects are surrounded by small boxes.
2. Right-click the parent, select Establish Relationships, and select a propagation type and a mapped relationship.
   The Relationship Edit dialog opens if you are replacing an existing relationship.

   NOTE
   Alternatively, you can also use the context menu of the relationship to change one or more relationship types. To do so, select and right-click the relationship, then select Change Relationship, *propagation type*, *mapped relationship* from the context menu.
3. Click OK on the Relationship Edit dialog if necessary.

   NOTE
   If you need to straighten the line between the two objects, click the *Straighten selected edges* tool

   .

   The new relationship and propagation appear connecting the related items as follows:

   - If you have selected the *Advanced* 'Chart display complexity level' option (default), a letter indicates the propagation type (A for aggregate, B for bound, C for custom, O for operative). The tooltip for this letter displays the full propagation and relationship type followed by a colon (:) and the relationship significance value.
   - If you have select the *Advanced with names* 'Chart display complexity level' option, the relationship name appears. The tooltip for this name displays the full propagation and relationship type followed by a colon (:) and the relationship significance value.
   - For either option, the line connecting each set of related CIs is color-coded to indicate the propagation type (blue for aggregate, burgundy for bound, orange for custom, and pink for operative).
   - The relationship line tooltip displays the propagation type, relationship type, significance, and the names of the connected CIs.
4. Click Save, OK to exit Service Modeler.

Create a Propagation Policy with Operative Propagation

Operative propagation requires you to specify an impact threshold that causes only impact values greater than or equal to the threshold to propagate to the parent item. Operative propagation policy is useful in situations when a dependent CI is only affected by a child item when the child item's impact becomes severe.

By default, every relationship with operative propagation only propagates impact when the impact of the child CI is equal to or greater than 20. You can change the default operative propagation policy for each relationship that maps to operative propagation. The Service Modeler must be closed to change the default operative policy.

In the Service Modeler, you define operative policies that are specialized for individual relationships.

Follow these steps:

1. Create a new service or modify an existing service.
   The Service Modeler opens.
2. Do one of the following:
   – Select and right-click a relationship that has operative propagation and select Operative Policies, Edit. You can select multiple relationships for the same source CI.

     NOTE
     You must select the relationship so that the relationship line widens for the Edit option to be available when you right-click.
   – Select and right-click a CI that is the source of a relationship that has operative propagation and select Define Policies, Operative Policies, Edit.

   The Operative Policy Editor dialog opens. The table displays whether the policy is new or previously defined, the affected source CI, target CI, relationship, and current threshold.
3. Click Set in the Threshold column, enter an impact threshold between 1-40, and press Enter.
   The threshold changes.
4. (Optional) Do any of the following if necessary:
   – Click Initial to return the Threshold to its initial setting when you opened the dialog.
   – Click Default to return the Threshold to its default setting.
   – Keep the 'When saving, remove default policy equivalents' check box selected to disregard a custom operative policy equivalent to the defined default policy for that relationship and use the default rather than creating the custom policy.

     NOTE
     CIs in maintenance mode are excluded from custom policy calculations related to average or percentage.

   The threshold or setting changes accordingly.
5. Click OK.
   The policy is associated with the specified relationship. Save the service to save any created policies.
6. Click Save, OK to exit Service Modeler.

Any custom operative policies that you define appear in the Policies tab in the pane below the service topology. You can right-click existing custom policies to edit or delete them. You can edit and delete multiple selected operative policies at the same time.

Create a Propagation Policy with Custom Propagation

Custom propagation requires you to specify when and how to change the severity of a parent item. The severity of the parent item changes based on the impact value of the children items.

Custom propagation policy is useful in situations when the impact of dependent CIs is complex. For example, a web server farm may have several servers that perform the same role. This redundancy means that a fault condition on one device may not impact the availability of the service. In fact, several servers may need to fail before users experience service degradation.

A sample rule is "If 50% of the servers are down, propagate an impact of Moderately Degraded. If 75% are down, propagate an impact of Severely Degraded." The default custom propagation policy contains the following rules:

- The first rule sets the parent item impact to Slightly Degraded when any of the included items have an impact greater than or equal to Moderately Degraded (20).
- The second rule sets the parent item impact to Moderately Degraded when any of the included items have an impact greater than or equal to Severely Degraded (30).

> **NOTE**
> When you assign custom propagation policy to a CI and the threshold is reached, the target CI attains the specified severity value and an infrastructure alert with the appropriate severity. However, because the alert applies to the CI as it relates to the parent service, the CI icon does not change color as expected, and the severity propagates to a separate impact calculation on the service, which is reflected in the service icon color.

You can change the default custom propagation policy. The Service Modeler must be closed to change the default custom policy.

> **NOTE**
> CIs in maintenance mode are excluded from custom policy calculations related to average or percentage.

In the Service Modeler, you define custom policies that are specialized for individual relationships. A service can have multiple custom propagation policies, and a custom propagation policy between CIs can have up to four rules for propagating impact.

Follow these steps:

1. Create a new service or modify an existing service.
   The Service Modeler opens.
2. (Optional) Create a group if you have multiple configuration items that will have the same policy.
3. Right-click a parent object or group that has custom propagation and select Define Policies, Custom Policies, Edit.
   The Policy Editor dialog opens.
4. (Optional) Select the Automatically Maintained check box to automatically change the relationship to Is Affected By or Is Evolution Of and apply the custom propagation policy when new CIs are added to a group that uses custom propagation. Auto-maintenance applies to relationships added by both manually editing a service and through automation by service discovery or service import.

   > **NOTE**
   > You can set Automatic Policy Maintenance for the Modeler operation to on by default in the Set Preferences dialog.

   Select one of the following from the Policy Type drop-down list:

   - **Average**
     Sets the impact of the parent item based on the average impact values of CIs associated with the policy.
   - **Percentage**
     Sets the impact of the parent item based on a percentage of CIs that have the impact specified in the rule.
   - **Any**
     Sets the impact of the parent item when any CIs associated with the policy have the impact specified in the rule.
   - **All**
     Sets the impact of the parent item when all CIs have the impact specified in the rule.

   Default: Any

   The text of the rules on the dialog changes to reflect the selected policy type.

   Complete the following fields and settings for each rule that you want to create:

   - **% of items**

(Percentage type only) Defines the percentage of items that must exceed the impact threshold to meet the rule criteria.

- **Threshold**
  Defines the impact threshold. The appropriate amount of CIs (as defined by the policy type) must meet or exceed the impact threshold to meet the rule criteria. Define a number between 0 and 40. The impact numbers translate to the following impacts:
  - 0: Operational
  - 1-10: Slightly Degraded
  - 11-20: Moderately Degraded
  - 21-30: Severely Degraded
  - 31-40: Down
- **Set Severity**
  Defines the severity to assign to the parent item if the rule criteria are met.

The text for each rule changes to reflect the new settings. Define as many of the four available rules as the policy requires.

5. Select the entities to which to apply the policy. Use the arrows to move CIs from the Available Configuration Items pane.
   All items included in the policy appear in the 'Configuration Items Included In Policy' pane.
6. (Optional) Click the 'Create a new policy' tab and create a different policy for any items that were not included in the current policy.
7. Click OK.
   The policy or policies are associated with the specified items. You must save the service to save any created policies.
8. Click Save, OK to exit Service Modeler.

Any custom policies that you define appear in the Policies tab in the pane below the service topology. You can right-click existing custom policies to edit or delete them. You can edit one custom policy at a time and delete multiple custom policies at the same time.

> **NOTE**
> Only custom policies that you have created appear in the Policies tab, not the defaults.

Assign the CI Significance in the Service

Significance indicates the importance of a CI to related items in a service, and therefore influences the impact of a CI condition on those related items. It is a value from 1 through 10, where 1 is the least significant and 10 is the most significant. You can set global significance for CI types, and you can set significance for individual CIs and relationships in a service.

Consider the following:

- When you set global significance, it applies to new CIs that you create after the change. Existing CIs retain the previous significance for the type.
- Before you change the significance for a service, verify that all relationships are established. If you change a relationship later, the significance reverts to the default for the CI type.
- When you change significance for an individual CI in a service, the CI's significance value only changes within that service. If the CI belongs to other services, it retains its previous significance value in those services. This behavior helps ensure that you can maintain different significance values for the same CI if it is more or less important to other services.

Follow these steps: (global type significance)

1. Start the Operations Console, and select Tools, Default Significance.
   The Global Significance Editor dialog opens, showing the default significance for each CI type.
2. Adjust the slider for the types whose significance you want to change, and click OK.

The global significance changes.

Follow these steps: (individual CI significance)

1. Open the Service Modeler.
2. Right-click a CI or relationship on the Topology tab, select Assign Significance, Set from the shortcut menu, and select a number from 1 to 10.
   The previous significance is replaced. The relationship text in the Modeler displays the significance of the relationship between the connected CIs.
3. Click Save, OK to exit Service Modeler.

Create and Assign a Service-Level Agreement

A *service-level agreement* (SLA) is a contract that specifies the service expectations of internal or external customers. An example is the downtime that is acceptable for various resources.

An SLA can, for example, help ensure that an online shopping site is available and delivering the level of service that internet shoppers expect. If performance is poor, some transactions can be lost, thus reducing profits and discouraging repeat customers.

You can define SLAs when you define or edit service models using the SLA tab. CA SOI uses the measurable provisions that you specify in the SLA to monitor the real-time health of each associated service, and records outage time when the service is down. The recorded time is compared to the SLA thresholds to determine the status of the SLA for a given time period.

For more information about defining, managing, and visualizing SLAs, see How to Create and Work with Service-Level Agreements.

Create and Assign an Escalation Policy

After you define service models and their associated CIs, relationships, and propagation policies, you can add escalation policy.

Escalation policy specifies the automated actions to take in response to fault conditions.

Some of the conditions that require an automated response are as follows:

- Alert impact on the service
- Time since acknowledgment
- Alert attribute values
- Time spent in an alert queue

The following are common actions:

- Sending an email message to an operator or service owner
- Opening a help desk ticket
- Invoking a script or application to help diagnose and remedy the fault condition

The following kinds of escalation polices are available:

- **Nonglobal**
  Escalates alerts in one or more specified services or alert queues that meet the policy criteria. For example, you can create nonglobal escalation policy for a payroll service owner who requires a notification when an alert is raised against the payroll service. When the policy considers an alert for escalation, it evaluates the alert against the nonglobal policy first, before any global policy.
- **Global**

Escalates all alerts that meet the policy criteria. For example, you can create global escalation policy for an IT manager who requires notification when *any* service alert is raised in CA SOI.

You can define escalation policy in the Operations Console or when creating or editing a service in the Service Modeler using the Alert Escalation tab.

For more information about defining escalation policy and escalation policy actions, see How to Create Escalation Policy.

Validate and Save the Service

Contents

You save a service to apply any changes that you make in a Modeler session. The Modeler validates the contents of the service automatically each time you save. You can also validate a service at any time to verify that any operation does not invalidate the service model.

Follow these steps:

1. (Optional) Click Perform Service Validation

   at any time.

   NOTE
   This operation does not save the service. When you do save, validation is performed automatically.

   One of the following occurs:
   – The Validation Results dialog opens with a message stating that no problems were detected in the service topology model.

     NOTE
     You can set a preference not to display this dialog when the validation is successful. For more information about how to set a preference, see Operations Console Customization.
   – The Service Topology Validation dialog opens with a list of problems in the service model construction that would cause service impact calculation errors, such as a state propagation loop. For more information about the types of potential errors, see How Service Validation Works.

     NOTE
     The topology view highlights the objects (that constitute a validation problem) when a specific problem is selected in the list that displays all the validation problems in the service model.
2. Fix service validation errors if necessary.
3. Click Save, OK in the Service Modeler when you are ready to save the service.
   One of the following occurs:
   – The Validation Results dialog opens stating that problems with the topology have prevented the save operation. When you click OK on this dialog, the Service Topology Validation dialog opens and displays validation errors. You must fix service validation errors before you can save the service.
   – The Save Service dialog opens stating that the service changes were saved successfully.

     NOTE
     You can set a preference to hide this dialog when the save is successful. For more information about how to set preferences, see Operations Console Customization.
   – A dialog opens stating that the service contains noncritical errors. You can select to save the service, but it is placed in the offline/test mode until you fix the errors.

How Service Validation Works

Each step in the service validation process tests one aspect of the service topology. The tests target the structure of the service to verify that a service model is valid and will not cause impact propagation calculation errors.

The Service Modeler performs validation automatically before each save attempt. You can also validate on demand by clicking Perform Service Validation

.

Each validation test is assigned a rank, which determines its impact on the subsequent service save process. The ranks and their effect are as follows:

- **UNKNOWN**
  Indicates that the test cannot determine the effect on the service.
- **LOW**
  Indicates that the model is inefficient, is missing items, or could use cosmetic improvements.
- **MEDIUM**
  Indicates that the model structure may affect the operation of the state engine.
- **HIGH**
  Indicates that the service cannot be put online in its current structure.
- **CRITICAL**
  Indicates that the service cannot be saved.

The tests run in the following order:

1. **CRITICAL:** The topology graph must be connected. The topology must not contain unconnected subgraphs; it must be possible to move between any nodes of the topology using associations, or all CIs must have relationships with other CIs.
2. **HIGH:** The topology graph must not have multiple roots. The state or impact of all CIs should propagate to one node or CI.
3. **CRITICAL:** The root node or CI must not have dependents. The root node is the final target for state or impact propagation.
4. **HIGH:** The topology graph (or subgraphs) must not have cycles formed by relationships. State propagation through relationships must not form loops.
5. **CRITICAL:** Component services of the tested service must not have antecedent CIs. Specifically, no antecedents are prescribed by the tested service's topology; the component services may be complex.
6. **LOW:** All graph nodes must have labels.

Results display in the Service Validation Results dialog if errors occurred. The table in this dialog has the following columns:

- **Rank**
  Defines the error rank.
- **Type**
  Defines the error type. Types include EDGE and PATH.
- **Root**
  Defines whether the problem affects the root service CI.
- **Object**
  Defines the affected parent object.
- **Description**
  Describes the nature of the problem.

Click each result for the Modeler to highlight the objects and relationship involved in the error. Use this visual information and the Description to resolve problems before saving the service model.

Service Modeling Examples and Scenarios

This section includes service modeling examples and scenarios. Examples of real-world scenarios and service models that illustrate the service modeling process are provided.

This section contains the following topics:

- Scenario 1 - Finance Service
- Scenario 2 - Shopping Cart Service
- Scenario 3 - Dynamic Service Policy
- Scenario 4 - Automatic Relationships Policy
- Scenario 5 - Unmanaged Relationships Policy

Service Modeling Example 1 - Finance Service

Contents

This scenario involves a large company with a vast infrastructure that must stay online to maintain the continuity of business processes. The company's IT manager is tasked with creating a service model representing the infrastructure components that comprise the company's finance department.

Finance Service Resources

The finance department resources must remain available so that the company maintains an accurate record of financial activity. Resources that affect the finance department include the following:

- A finance server hosts critical finance applications
- Database servers and databases maintain financial records
- A subnetwork under which the finance resources run

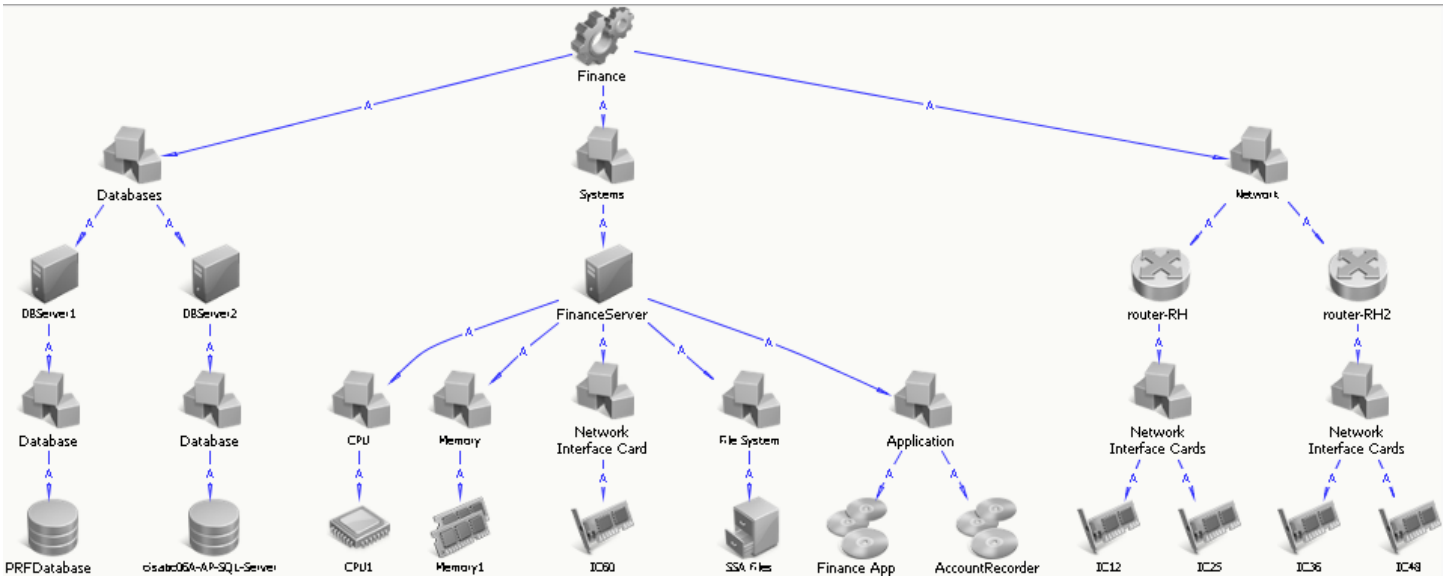
The following domain managers manage these resources:

- CA Insight DPM manages the finance database.
- CA Spectrum IM and CA eHealth manage the finance subnetwork.
- CA Application Performance Management manages the finance applications.

All important finance resources are currently managed, but no domain manager provides a consolidated view of all resources affecting the finance department infrastructure. Therefore, the domain managers cannot accurately represent service status and the root cause of problems affecting the service. A service model in CA Spectrum SA can leverage the intelligence of these domain managers to accomplish what a single domain manager cannot.

Finance Service Model

The IT manager models the Finance service in CA Spectrum SA as shown in the following graphic:

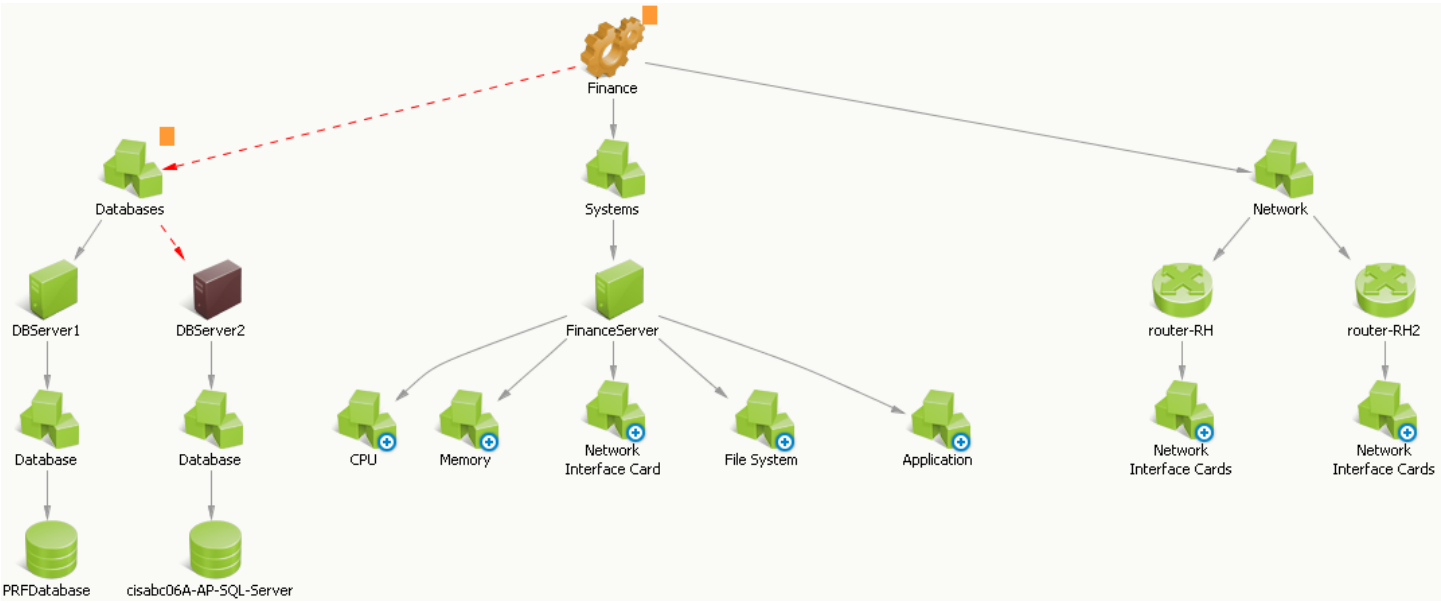


The Finance service has the following characteristics:

- It is a bottom-up service model arranged by domain. Logical groups represent the network, systems, and database resources.
- The service model uses logical groups to assemble CIs of the same type (for example, Application and Network Interface Cards) and create a logical, easy to understand representation of all resources.
- The service is modeled organically with no services imported from the source domain managers. If any of the source domain managers contained a service representation similar to one of the top-level groups, the IT manager could have imported that service and added it to the Finance service as a subservice.
- All CIs and groups use aggregate propagation, so that impact propagates directly from related items.
- Employees enter financial records using the Finance App and AccountRecorder applications. If these applications fail, the entire finance department cannot function. Therefore, the IT manager increased the significance of the application CIs from the default of 4 to 8. Because the FinanceServer hosts the applications, the significance of the FinanceServer CI has also been increased from the default of 5 to 9. These changes adjust the significance to reflect that the finance applications are the most important aspects of the Finance service, other than the supporting network.

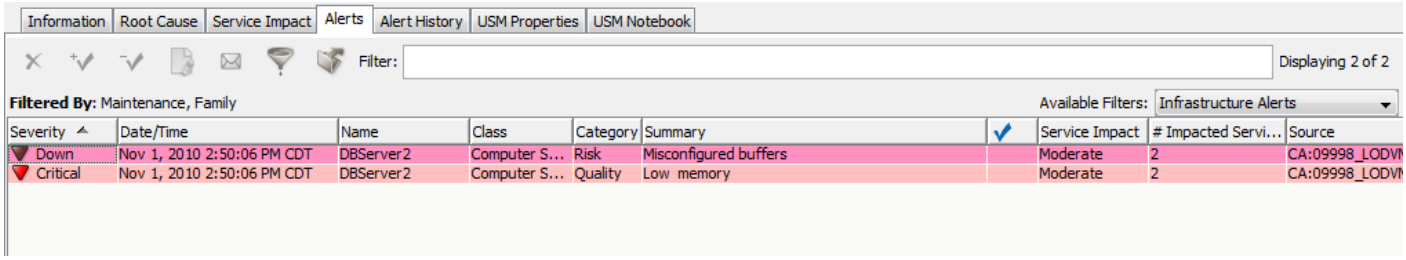
Example Finance Service Escalation Flow

The modeled Finance service provides the necessary information to quickly detect and resolve issues that cause service degradation. Consider this simple service condition, displayed from the Topology view of the Operations Console as shown in the following graphic:



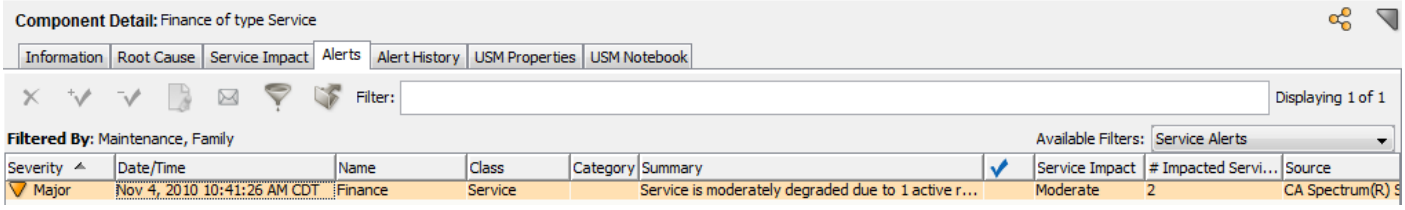
The conditions appear and are resolved as follows:

1. Two alerts for DBServer2 appear in the Operations Console as shown in the following graphic:

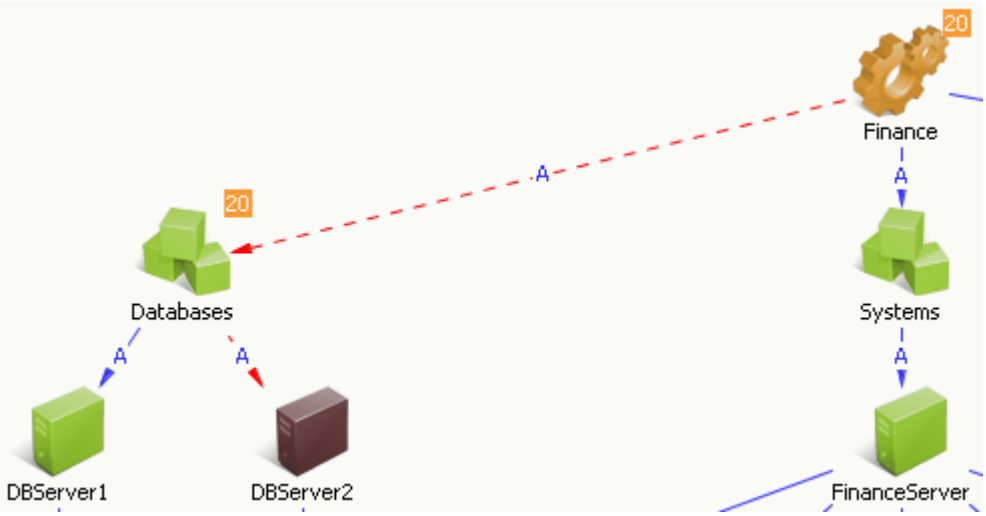


   These infrastructure alerts originated from the source domain manager (in this case, CA Insight DPM) and indicate Down and Critical alerts on the DBServer2 CI. If configured, escalation policies could perform specified actions on the alert, such as opening a help desk ticket or sending an email to a technician.

2. The SA Manager calculates the impact on the Finance service, changes the Finance service status, and generates a service alert as shown in the following graphic:

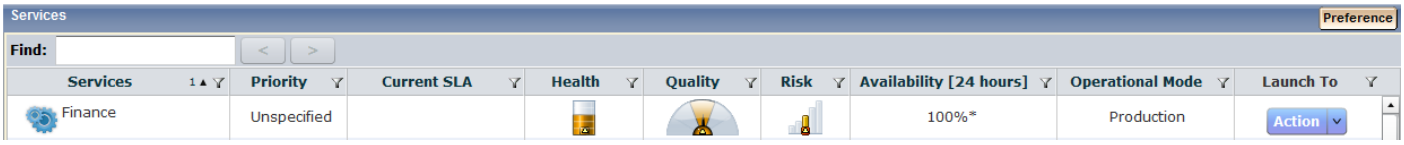


   The Major severity results from the service impact value of 20 as shown in the following graphic:



CA Spectrum SA calculates the impact based on the severity of the root cause alert and the significance of the alerted CI. In this case, the root cause alert is the Down alert because it is a higher severity than the Critical alert, which is confirmed in the Root Cause tab. If other areas of the service had a similar severity with a higher CI significance (FinanceApp, for example), the service impact would be greater, and this condition would instead aggregate to the service level as the root cause condition.

3. Service stakeholders can see a graphical view of all service conditions from the Dashboard as shown in the following graphic:



Notice that service quality and risk are both affected, because the Down alert belongs to the Risk category, and the Critical alert belongs to the Quality category. Health is a reflection of the worst state held by quality or risk.

4. To resolve the problem, the assigned technician right-clicks the alert to drill down into the source domain manager (in this case, CA Insight DPM) to learn more about the problems, if necessary.
5. The assigned technician reconfigures the database server buffers, resolves the lack of free memory, and clears the infrastructure alerts.
6. The alerts disappear from the Operations Console and the service condition returns to normal.

Service Modeling Example 2- Shopping Cart Service

Contents

This scenario involves a retail company that contracts a managed service provider to host its online ordering application and associated hardware resources. The managed service provider must create a Shopping Cart service to represent the infrastructure components of the retail company's online ordering application.

Shopping Cart Service Resources

The online ordering resources must remain available at all times and maintain specific quality levels, so that the retail company guarantees optimal customer experience and the managed service provider meets contractual obligations. Resources that affect the Shopping Cart service include the following:

- A web server farm that hosts the online ordering applications
- A database cluster that maintains shopping cart records

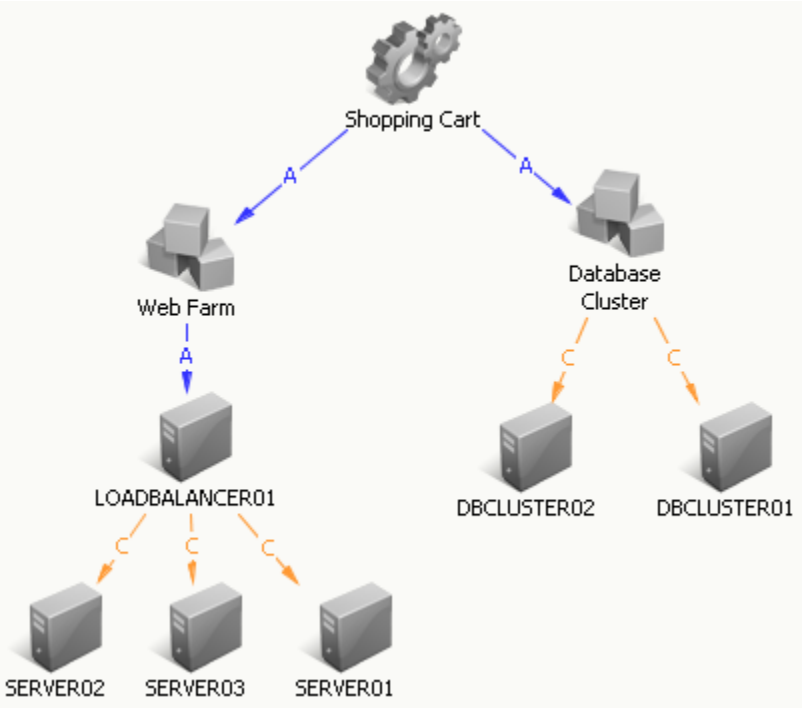
The following domain managers manage these resources:

- CA Spectrum IM, and CA eHealth manage the availability and health of the web server farm.
- CA Application Performance Management manages the performance of the underlying applications.
- CA CMDB and CA Application Performance Management maintain a record of the service and its underlying components.
- CA Application Configuration Manager monitors the configuration compliance of all components.
- CA Service Desk is the integrated help desk application.
- CA Insight DPM manages the availability and health of the database cluster.

The important online ordering resources are currently managed, and CA CMDB maintains a record of the complete service. However, CA Spectrum SA can combine the service view with the management information, detect the root cause of problems, and provide a quick path to resolution through links to the source domain managers and the integrated help desk application.

Shopping Cart Service Model

The managed service provider imports the Shopping Cart service from CA CMDB. The service appears in CA Spectrum SA as shown in the following graphic:



The Shopping Cart service has the following characteristics:

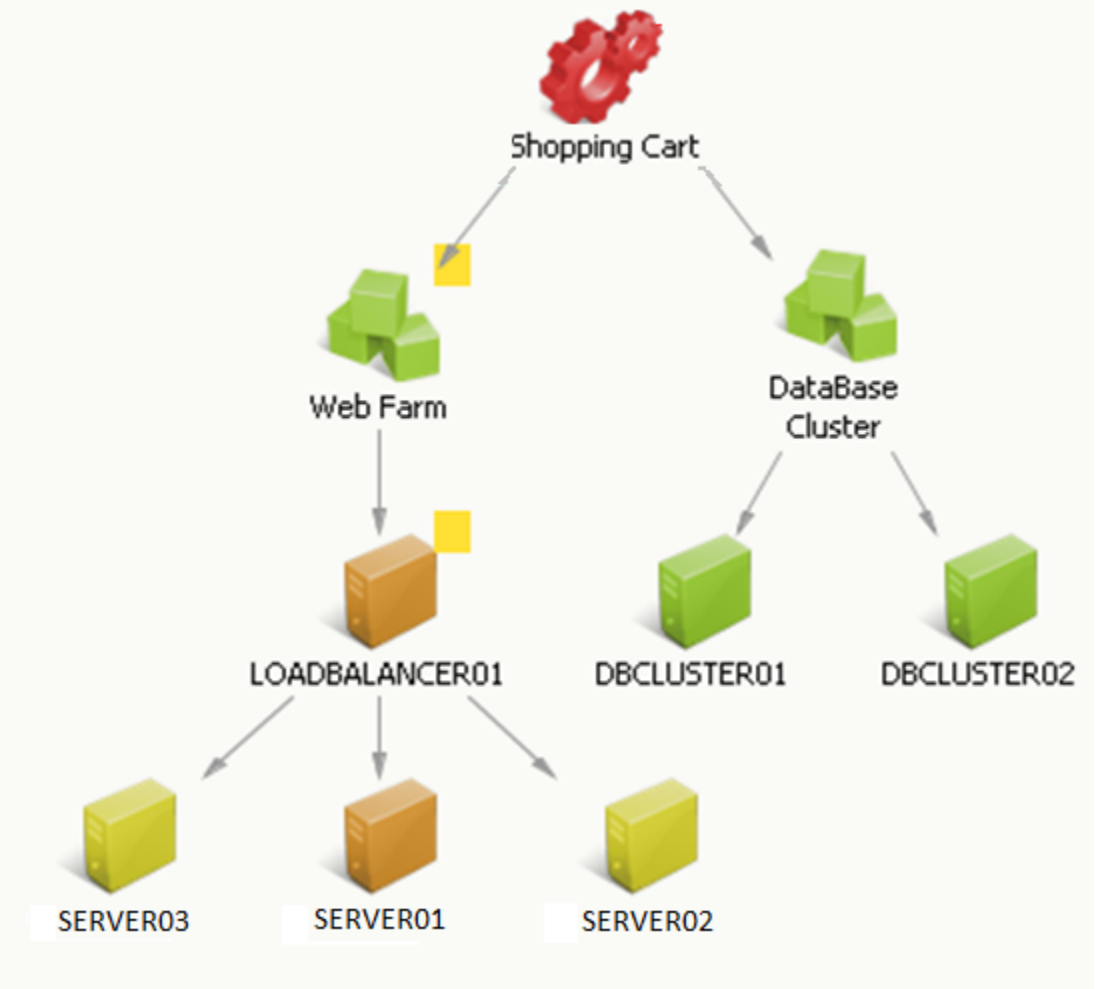
- Logical groups represent the two main resource domains: the web server farm and the database cluster. The groups use aggregate propagation to the service as shown by the letter A on the arrows connecting the CIs.
- The CIs in the web server farm and database cluster use custom propagation as shown by the letter C on the arrows connecting the CIs. Each group of CIs has an associated custom propagation policy that specifies the conditions under which to propagate impact to the related CI (LOADBALANCER01) or group (Database Cluster). Custom propagation

policy takes into account that if one server in a cluster or web farm fails, this should not have a severe impact on the service, as long as the other servers are running at peak performance.

- The managed service provider has created escalation policy that opens a CA Service Desk ticket when a service alert occurs.
- CA Application Performance Management and CA Application Configuration Manager track important service-level metrics and compliance information for the service and associated CIs. If necessary, the managed service provider could also create an SLA for the service that tracks performance against service health, quality, risk, or availability.

Example Shopping Cart Service Escalation Flow

The Shopping Cart service in CA Spectrum SA provides the necessary information to quickly detect and resolve issues that cause service degradation. Consider these simple service conditions, displayed from the Topology view of the Operations Console as shown in the following graphic:



These conditions appear and are resolved as follows:

1. Alerts appear in the Operations Console for the Web Farm servers as shown in the following graphic:

Severity	Name	Ticket ID	Category	Description
Major	SERVER01		Risk	HIGH CPU UTILIZATION
Minor	SERVER02		Risk	Low CPU Utilization. The utilization of 5% for CPU instance 0x015777 named 'CPU_T
Minor	SERVER03		Risk	Low CPU Utilization. The utilization of 7% for CPU instance 0x064321 named 'CPU_T
Major	LOADBALANCER		Risk	Configuration for device: Loadbalancer01 is outside of standard compliance

SERVER01 has a major severity alert for high CPU utilization, while SERVER02 and SERVER03 have minor severity alerts for unusually low CPU utilization. LOADBALANCER01 has a major severity alert from CA Application Configuration Manager for a compliance violation.

2. The impact of the web farm alerts does not propagate, because it is not high enough to meet the conditions defined in the custom propagation policy. However, these alerts do affect the service risk value.
3. An alert appears for the Shopping Cart service CI as shown in the following graphic:

Critical	Shopping Cart		Quality	Transaction times have breached a critical threshold on Shopping Cart Service

This alert from CA Application Performance Management indicates a critical threshold breach in the transaction time for online ordering.

4. The critical alert causes a similar service alert as shown in the following graphic:

Critical	Shopping Cart	60	Service is severely degraded due to one active critical alarm

This service alert changes the Shopping Cart service condition to critical.

5. Escalation policy automatically opens a ticket in CA Service Desk in response to the service alert. The assigned technician can directly access the ticket in CA Service Desk by clicking the link on the number in the Ticket ID column.
6. The assigned technician consults CA Spectrum SA, CA Service Desk, and the source domain manager to resolve the problem before the managed service provider violates contractual obligations.

Service Modeling Example 3 - Dynamic Service Policy

Contents

This scenario involves a system administrator who wants to automatically place relevant CIs system-wide under services. The system administrator has two sets of conditions and will create a policy for each:

- All hardware in a specific region
- All virtual systems that exist and those added in the future

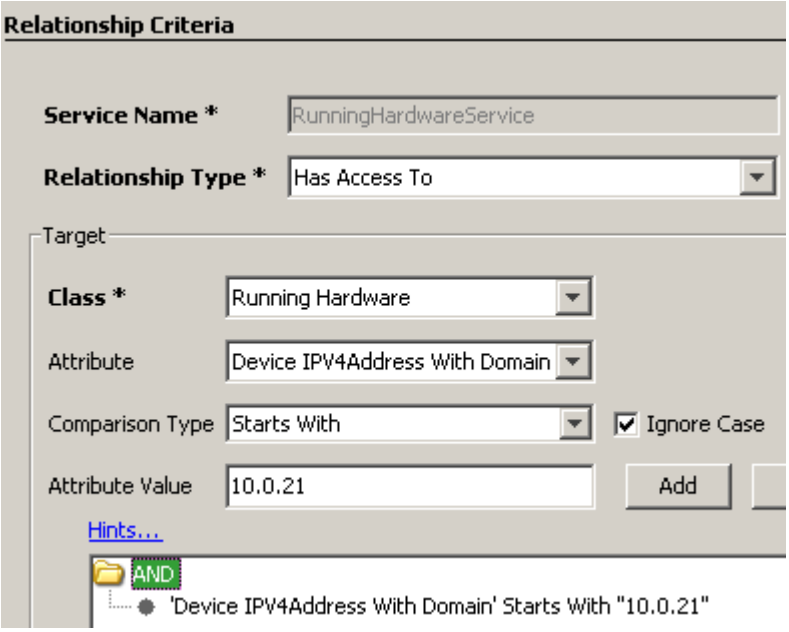
Policy - All Running Hardware with Specific Starting IP Address

Because one of the company's enterprise regions is located in a specific subnet, the system administrator wants to create dynamic service policy to collect the hardware resources from that region and add to a service. This provides the system administrator with a consolidated view of that region. The system administrator could then model the relationships among the resources or set up automatic relationships.

Because all hardware resources in the region begin with the same IPV4 address (10.0.21.*x*), the system administrator creates a dynamic service policy for all IPV4 addresses starting with 10.0.21.

The system administrator creates a new dynamic service policy using the Service Discovery Policy Editor wizard:

- **Define Service**
  The system administrator completes the Define Service page as follows:
  Service Name: RunningHardwareService
  Relationship Type: Has Access To
  The system administrator then completes the Target fields:
  Class: Running Hardware
  Attribute: Device IPV4Address With Domain
  Note that all USM properties available are available except the following:
  – domain properties (MdrProduct, MdrProductInstance, MdrElementID)
  – date and time properties
  Comparison Type: Starts With
  Attribute Value: 10.0.21
  The system administrator clicks Add and the criteria appear as shown in the following graphic:



If the system administrator wanted to put a dynamic service CI with a different class, the system administrator can define another policy which has the same service name and relationship type, but a different class.

- **Confirm**
  The dynamic services policy appears on the Confirm page as follows:

```
Dynamic Service Policy 'RunningHardwareService'
WITH
    relationship 'Has Access To'
FOR
    Running Hardware with properties
    (
        'Device IPV4Address With Domain' starts with (ignore case) "10.0.21."
    )
```

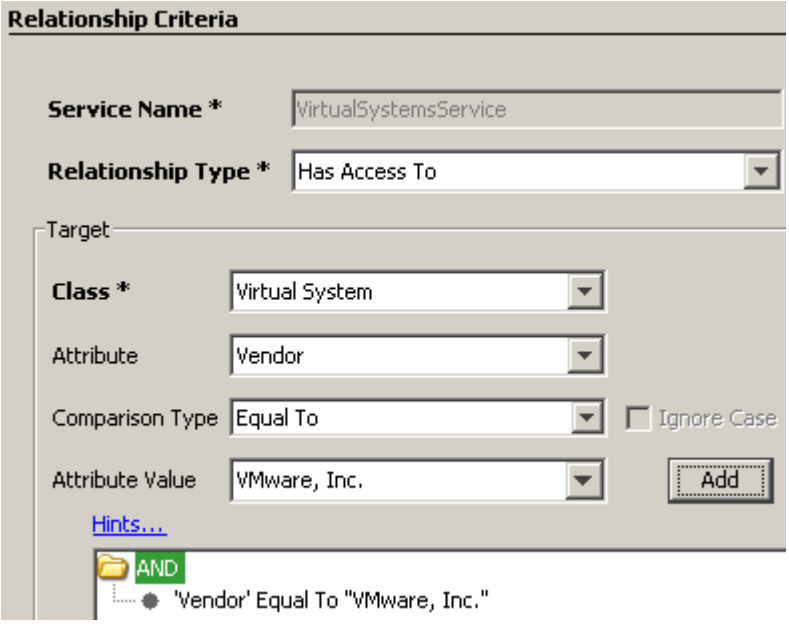
Policy - All Virtual Systems by a Specific Vendor

The company is converting its data center to virtual resources over time. The system administrator wants a service container for all existing virtual machines and virtual machines that come online in the future.

The system administrator creates a dynamic service policy that automatically puts all virtual systems by vendor VMware, Inc. under a service.

The system administrator creates a new dynamic service policy using the Service Discovery Policy Editor wizard:

- **Define Service**
  The system administrator completes the Define Service page as follows:
  Service Name: VirtualSystemsService
  Relationship Type: Has Access To
  The system administrator then completes the Target fields:
  Class: Virtual System
  Attribute: Vendor
  Comparison Type: Equal To
  Attribute Value: VMware, Inc.
  The system administrator clicks Add and the criteria appear as shown in the following graphic:



- **Confirm**
  The dynamic services policy appears on the Confirm page as follows:

```
Dynamic Service Policy 'VirtualSystemsService'
WITH
    relationship 'Has Access To'
FOR
    Virtual System with properties
    (
        'Vendor' equals "VMware, Inc."
    )
```

Service Modeling Example 4 - Automatic Relationships Policy

Contents

This scenario involves a system administrator whose company is in the early stages of adopting a service-oriented infrastructure and wants to define general rules to automate universal granular relationships that the service administrator would otherwise have to create manually.

The system administrator has two sets of criteria for creating relationships and will create a policy for each:

- All computer systems and software that runs on them
- All database instances and their tablespaces

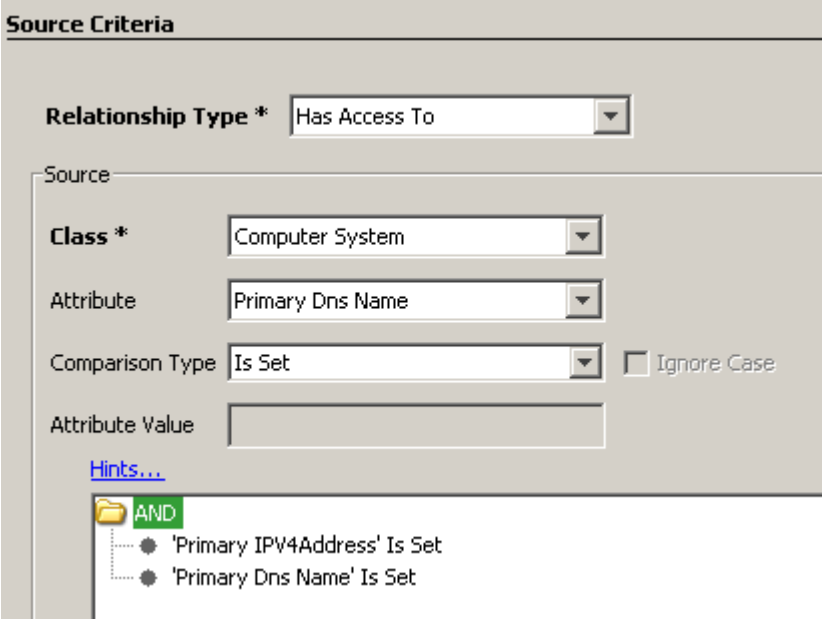
Policy - All Computer Systems and Running Software

The company is in the process of hiring more employees and so they are constantly adding new computer systems bundled with various software packages.

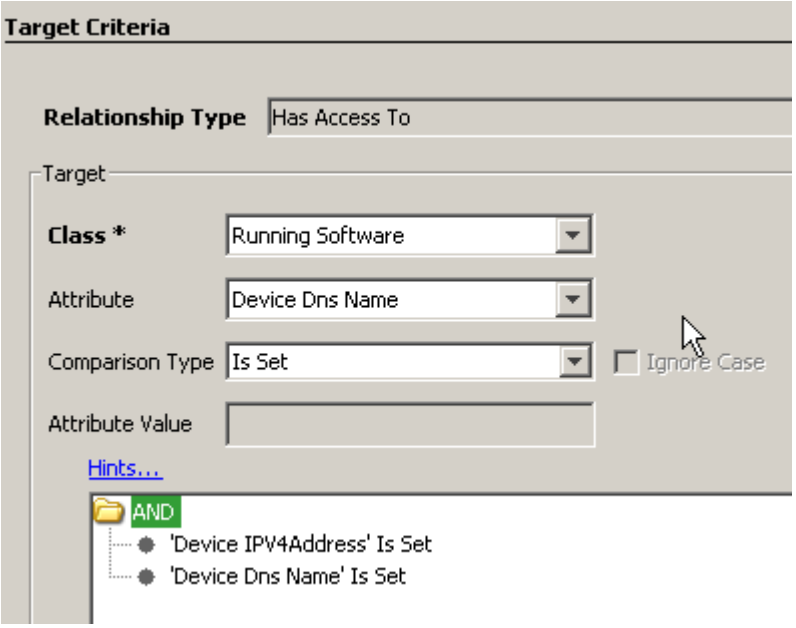
The system administrator wants to automatically create relationships between computer systems and the software that runs on them. When new computer systems come online, the system administrator wants the relationship creation automated also.

The system administrator creates a new automatic relationship policy using the Service Discovery Policy Editor wizard:

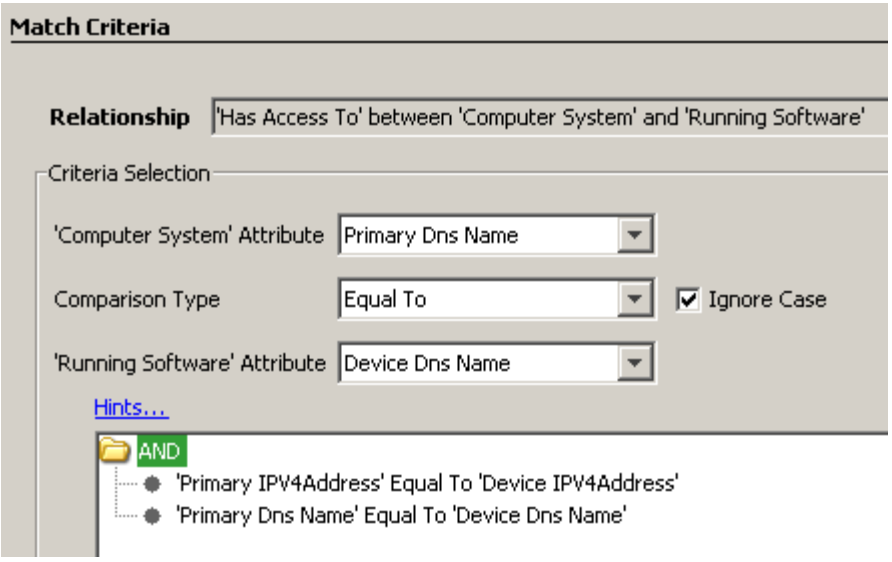
- **Source Criteria**
  The system administrator completes the Source Criteria page with the following values:
  Relationship Type: Has Access To
  Class: ComputerSystem
  The system administrator creates the first source criteria:
  Attribute: Primary IPV4Address
  Comparison Type: Is Set
  The system administrator clicks Add then creates the second source criteria:
  Attribute: Primary Dns Name
  Comparison Type: Is Set
  The system administrator clicks Add and the resulting source criteria appear as shown in the following graphic:



- **Target Criteria**
  The system administrator clicks Next and completes the Target Criteria page as follows:
  Class: RunningSoftware
  The system administrator creates the first target criteria:
  Attribute: Device IPV4Address
  Comparison Type: Is Set
  The system administrator clicks Add then creates the second target criteria:
  Attribute: Device Dns Name
  Comparison Type: Is Set
  The system administrator clicks Add and the resulting target criteria appear as shown in the following graphic:



- **Match Criteria**
  The system administrator creates the first criteria:
  'Computer System Attribute: Primary IPV4Address
  Comparison Type: Equal To
  'Running Software' Attribute: Device IPV4Address
  The system administrator clicks Add and creates the second criteria:
  'Computer System Attribute: Primary Dns Name
  Comparison Type: Equal To
  'Running Software' Attribute: Device Dns Name
  The system administrator clicks Add and the resulting match criteria appear as shown in the following graphic:



- **Relationship Scope**
  The system administrator selects All services so that all services with matching ComputerSystem CIs are considered for creating the relationship.
- **Confirm**
  The automatic relationship policy appears on the Confirm page as follows:

Automatic Relationship Policy 'Has Access To'

```
Automatic Relationship Policy 'Has Access To'
BETWEEN
Computer System with properties
(
'Primary IPV4Address' is set AND
'Primary Dns Name' is set
)
AND
Running Software with properties
(
'Device IPV4Address' is set AND
'Device Dns Name' is set
)
WHEN
(
'Computer System.Primary IPV4Address' equals (ignore case) 'Running Software.Device IPV4Address' AND
'Computer System.Primary Dns Name' equals (ignore case) 'Running Software.Device Dns Name'
)
SCOPED TO
(
All services
)
```

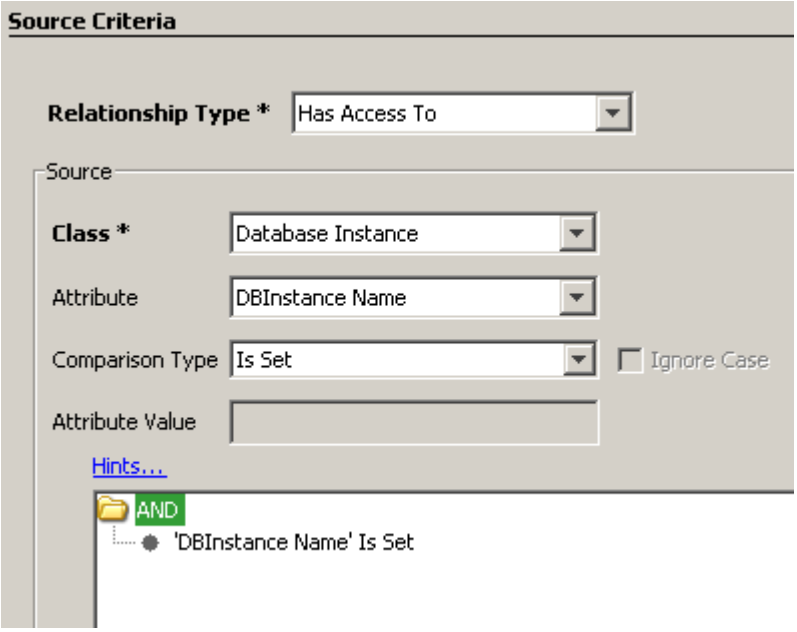
Policy - All Database Instances and Tablespaces

The company is adding new enterprise software that requires adding multiple databases. The company is also planning on adding additional databases in the future.

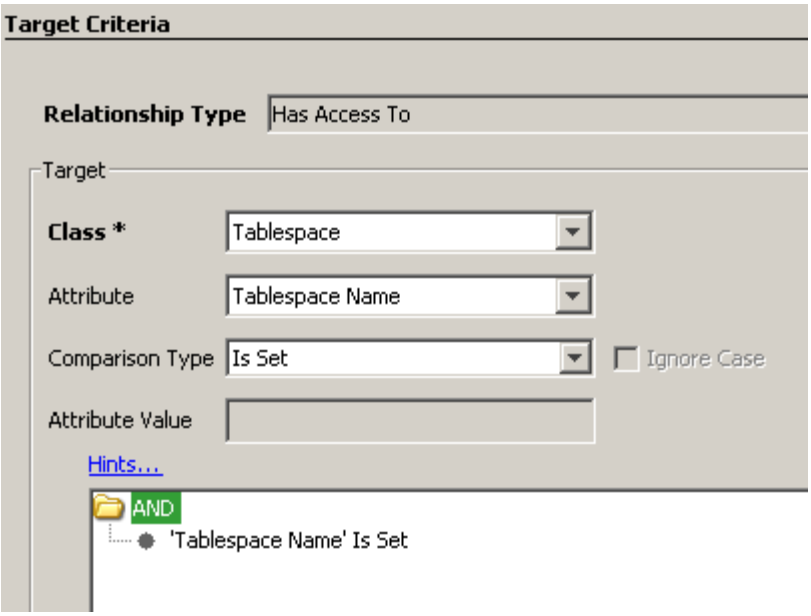
The system administrator wants to automatically create relationships between database instances and their respective tablespaces. When a new database comes online, the system administrator wants the relationship creation automated also.

The system administrator creates a new automatic relationship policy using the Service Discovery Policy Editor wizard:

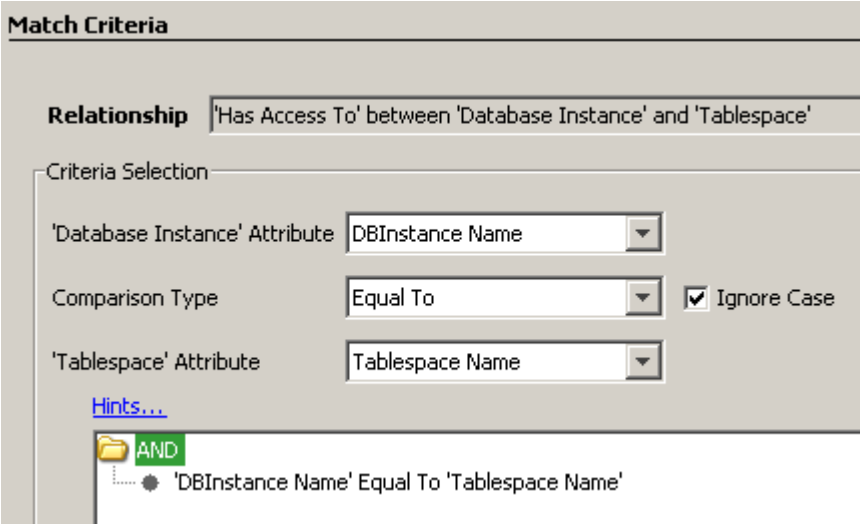
- **Source Criteria**
  The system administrator completes the Source Criteria page with the following values:
  Relationship Type: Has Access To
  Class: Database Instance
  Attribute: DBInstanceName
  Comparison Type: Is Set
  The system administrator clicks Add and the resulting source criteria appear as shown in the following graphic:



- **Target Criteria**
  Class: Tablespace
  Attribute: Tablespace Name
  Comparison Type: Is Set
  The system administrator clicks Add and the resulting source target criteria appears as shown in the following graphic:



- **Match Criteria**
  'Database Instance" Attribute: DBInstance Name
  Comparison Type: Equal To
  'Tablespace' Attribute: Tablespace Name
  The system administrator clicks Add and the resulting match criteria appears as shown in the following graphic:



- **Relationship Scope**
  The system administrator selects All services so that all services are considered for relationship creation.
- **Confirm**
  The automatic relationship policy appears on the Confirm page as follows:

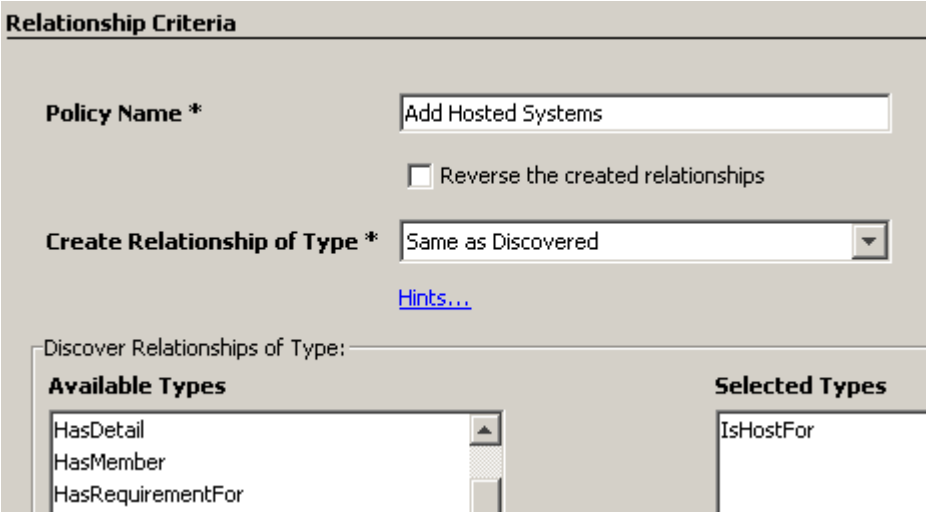
```
Automatic Relationship Policy 'Has Access To'
BETWEEN
    Database Instance with properties
    (
        'DBInstance Name' is set
    )
AND
    Tablespace with properties
    (
        'Tablespace Name' is set
    )
WHEN
    (
        'Database Instance.DBInstance Name' equals (ignore case) 'Tablespace.Tablespace Name'
    )
SCOPED TO
    (
        All services
    )
```

Service Modeling Example 5 - Unmanaged Relationships Policy

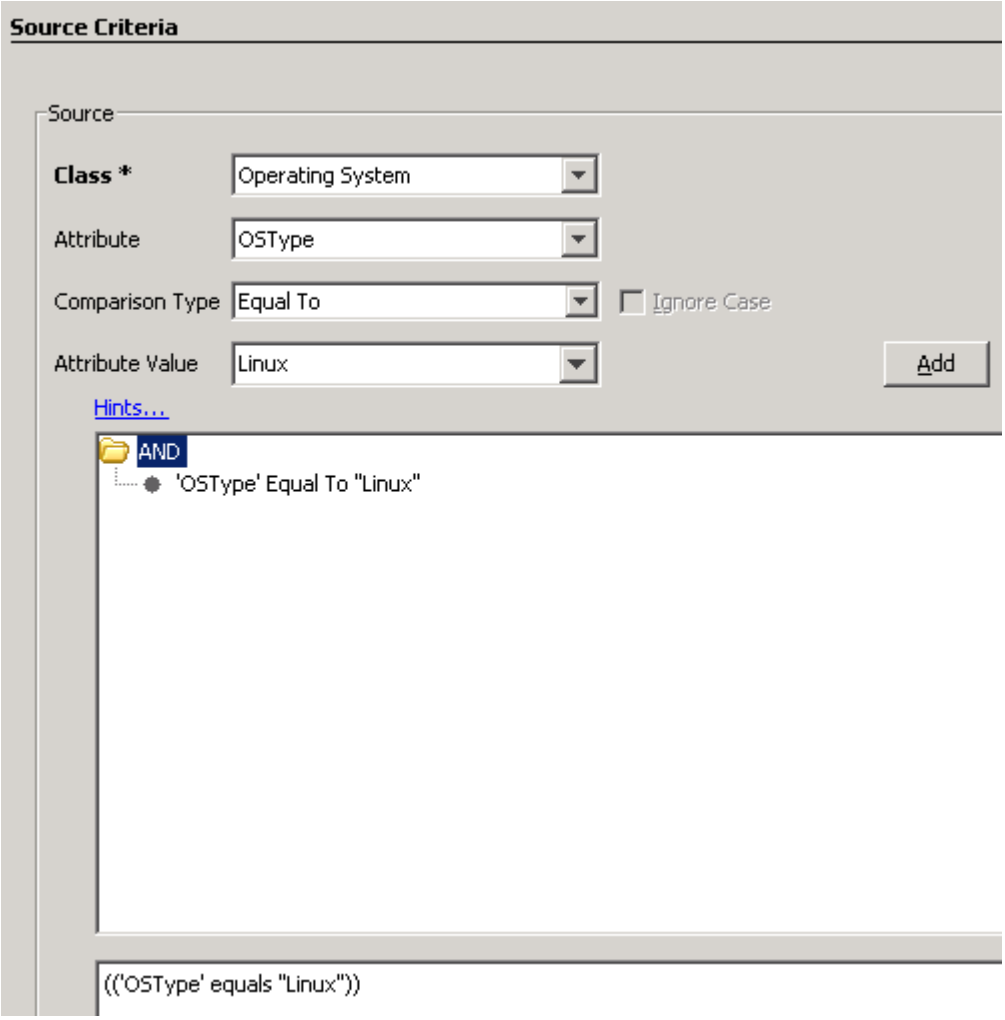
This scenario, *Add Hosted Systems*, involves a system administrator whose company's IT infrastructure includes a domain manager that publishes information about operating systems. The operating systems are Linux-based operating systems. They are either *hosting* operating systems (which implies they run a virtual computer or computers) or *hosted* operating systems (which implies they run on a virtual system). The domain manager also publishes IsHostFor relationship between *hosting* and *hosted* operating systems. The system administrator wants to create an unmanaged relationship policy. The policy ensures adding a *hosting* operating system to a service also adds all *hosted* operating systems.

The system administrator creates a new unmanaged relationship policy using the Service Discovery Policy Editor wizard:

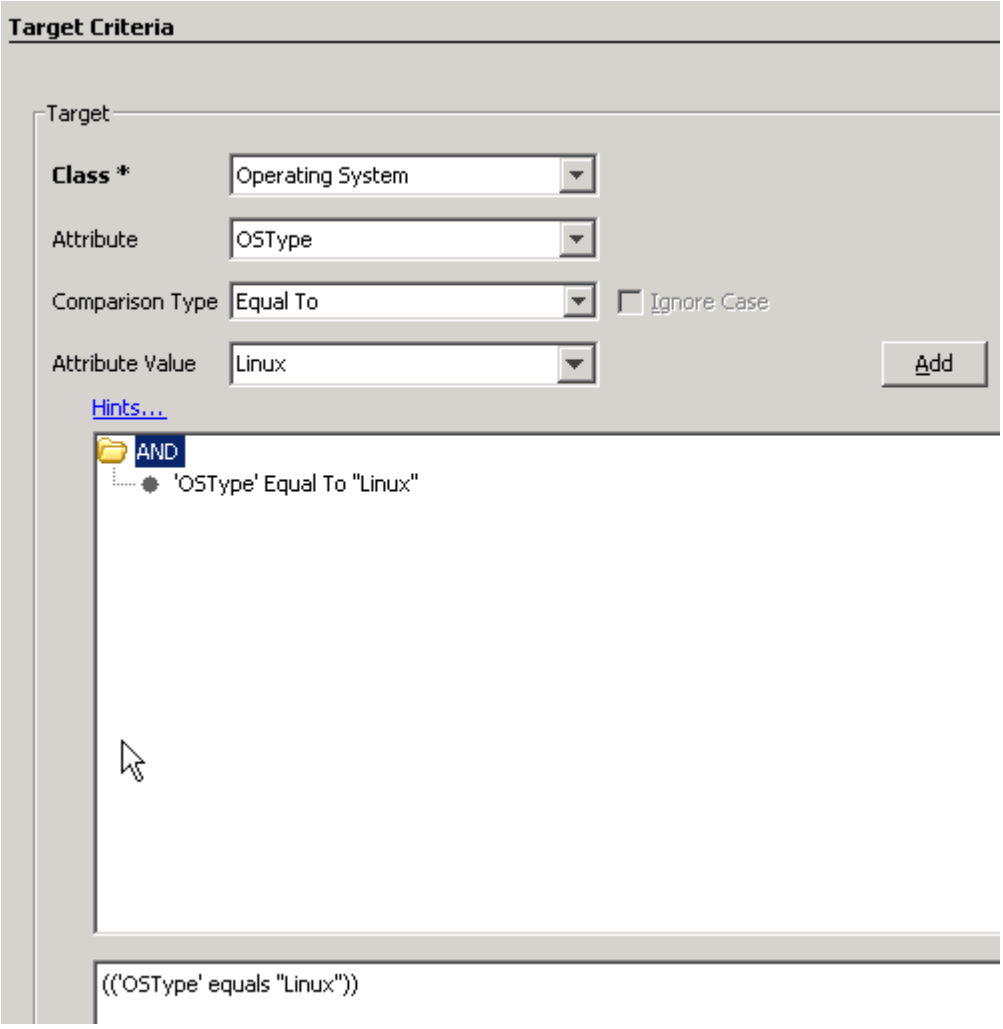
- **Relationship Criteria**
  The system administrator completes the Relationship Criteria page with the following values:
  Policy Name: Add Hosted Systems
  Create Relationship of Type: Same as Discovered
  Discover Relationships of Type: IsHostFor
  The system administrator clicks Add and the resulting relationship criteria appear as shown in the following graphic:



- **Source Criteria**
  The system administrator completes the Source Criteria page with the following values:
  Class: Operating System
  Attribute: OSType
  Comparison Type: Equal To
  Attribute Value: Linux
  The system administrator clicks Add and the resulting source criteria appear as shown in the following graphic:



- **Target Criteria**
  The system administrator completes the Target Criteria page with the following values:
  Class: Operating System
  Attribute: OSType
  Comparison Type: Equal To
  Attribute Value: Linux
  The system administrator clicks Add and the resulting target criteria appear as shown in the following graphic:



NOTE
This scenario does not require any match criteria.

- **Relationship Scope**
  The system administrator selects All services so that all services with matching OperatingSystem CIs are considered for creating the relationship.
- **Confirm**
  The unmanaged relationship policy appears on the Confirm page as follows:

```
Unmanaged Relationship Policy 'Add Hosted Systems' creates
relationships with
SEMANTIC
 Same as Discovered
BETWEEN
    Operating System with properties
    (
        'OSType' equals "Linux"
    )
AND
    Operating System with properties
    (
        'OSType' equals "Linux"
```

```
    )
SCOPED TO
    (
        All services
    )
BASED ON SEMANTICS
    (
       IsHostFor
     )
```

How to Customize Service Model Display

As an administrator, you can customize the zoom level, model layout, background type and image in the Operations Console to optimize the service model layout:

- Control Zoom Level
- Adjust Model Layout
- Control Line Layout and Appearance
- Adjust Background Type
- Add a Background Image

Control Zoom Level

Several tools exist in the Service Modeler to control the zoom level of the Topology view. You may require different zoom levels in large service models when you need to edit specific areas or view the entire model in one display.

To control zoom level, click one of the following:

-
  (Interactive Zoom Tool)
  Enlarges or reduces the topology when you click the tool and drag it on the screen.
-
  (Marquee Zoom Tool)
  Increases the magnification in a specific region when you click the tool and select a region in the topology.
-
  (Zoom Level Control)
  Specifies the amount of magnification. Select Auto fit to automatically fit the entire model within the display.

You can also do the following when any tool is selected:

- Hold the Ctrl key and scroll the mouse wheel to quickly zoom in or out.
- Hold Ctrl + Shift and scroll the mouse wheel to more finely adjust the zoom level and resolution.

Adjust Model Layout

Several tools exist to optimize the layout of the service model and how it is organized. Different types of service models require different layouts to best visualize the relationships and hierarchy. You can adjust the layout type to best fit the structure of your service model, and you can adjust the layout mode to specify the level of automated adjustment to perform to the model layout (according to the selected layout type) when you add items.

To adjust the overall service model layout type, click the Apply Automatic Layout

icon and select one of the following:

- **Circular**
  Emphasizes clusters that are present in a network's topology.
- **Grid**
  Arranges objects in horizontal rows and vertical columns.
- **Hierarchical**
  Emphasizes relationships among objects by placing them at different levels, like an organizational chart at a company. This is the default when you build services from scratch.
- **Orthogonal**
  Minimizes bend points by arranging objects horizontally and vertically, at 90 degree angles.
- **Symmetric**
  Emphasizes symmetries that are present in a service's topology. This is the default for newly discovered services. It is also the fastest and yields the smallest topologies.

The layout adjusts to the selected type.

To adjust the layout mode, right-click an empty area in the Topology view, select Layout Mode, and select one of the following:

> **NOTE**
> Any layout mode adjustment always adheres to the selected layout type.

- **Automatic**
  Performs an automatic layout adjustment on the whole model chart after every new item that you add to a model.
- **Incremental**
  Performs an automatic layout adjustment after every new item that you add to a model while minimizing the amount of movement.
- **Manual**
  Performs no automated layout adjustment. When you drag and drop an item into the model, the item is placed at the drop point, and relationships establish accordingly. When you add an item using the right-click menu or drop an item directly on a parent item, the Modeler uses a fixed location relative to the target parent item.

Control Line Layout and Appearance

You can control how the relationship lines appear that connect resources in a service model. Bent lines are the default, and can improve the presentation. Straight lines can improve the mobility of the nodes in the Topology. You can specify the line type only if the layout mode is Automatic or Incremental.

> **NOTE**
> Bent and straight lines do not apply to the Topology tab in the Operations Console.

To select bent or straight lines in the service topology, do one of the following:

- Right-click an empty area in the Topology view, select Layout Mode, and select or clear Bend Edges.
  The line type for the model is set.
- Select a bent relationship line and click the Straighten Selected Edges tool

  .

  The previously bent relationship line straightens.
- Select View Preferences in the Operations Console, expand Modeler, Edit Mode, and Layout in the Set Preferences dialog, select Edge Bends, and select Yes or No in the Edge Bends drop-down list.

The default line type is set.

- Right-click an empty area in the Topology view and select Route Edges.
  Edge bend points are modified so that the lines run orthogonally. Item positions do not change.
- To manipulate bend points, select the bend point (not the relationship line) and drag to modify the bend.

You can also control what information displays for relationships by doing any of the following:

- Click the Chart Complexity Level

  tool and select one of the following:
  - **Advanced**
    Displays a letter in each relationship line that indicates the propagation type.
  - **Advanced with names**
    Displays the full relationship name in the relationship line.

  For both options, the lines are color-coded to indicate the propagation type:
  - Aggregate: Blue
  - Bound: Burgundy
  - Custom: Orange
  - Operative: Magenta
- Click the Change relationship visibility tool

  and select one of the following:
  - **Specific Propagation Type**
    Displays only the relationships of the selected propagation type. This may be useful to pinpoint all relationships of a certain propagation, such as custom, to fine-tune their policy.
  - **Show All**
    Displays all relationship lines.
  - **Hide All**
    Hides all relationship lines.

Adjust Background Type

You can adjust the background type and pattern of the Topology view.

To adjust the topology background type, right-click an empty area in the topology and select Chart Grid and one of the following:

- **Type**
  Defines the pattern type of the topology background. Select from the following options:
  - **None**
    Displays and maintains no pattern in the topology background.
  - **Line**
    Displays lines that define the grid areas in the topology. This option presents grid areas as squares with divisions within the squares controlled by the Size setting.
  - **Point**
    Displays individual points that define the grid areas in the topology. This option presents grid areas as squares, but does not connect the areas with lines. You control the distance between the points with the Size setting.
  - **Invisible**
    Maintains chart areas that are not visible in the Modeler.
- **Size**

Defines the number of units that populate each area of the grid. This option only has obvious results for the Line and Point grid types.

Add a Background Image

You can add a background image to a service model to enhance the model presentation. For example, if your service can be modeled based on the location of CIs in a specific geographical area, you can add a map of the area as a background image and model the service on the map.

Follow these steps:

1. Copy the image to <SOI_HOME>\jsw\bin\sam_topo_images on the SA Manager system.
2. Open the Operations Console and open the Service Modeler to create or edit a service in which to place the image.
3. Right-click the background of the service topology on the Service Modeler and select Background Image, Select.
   The Image File Selection dialog opens.
4. Select the background image from the directory in Step 1 and click Select.
   The image appears in the background of the service topology.
5. Click OK.
   The changes are saved, and the service's topology includes the background image when you view it on the Operations Console.

Editing and Managing Services

Contents

As an administrator, you can modify and manage your service models. After you create a service, you can edit the model in the Service Modeler or manage or add properties to the service in the Operations Console. You can also associate your services to customers and manage them from the perspective of customers.

Modify a Service Model

You can modify any created or imported service at any time.

Follow these steps:

1. Open the Operations Console, right-click a service in the Navigation pane, and select Edit Service.
   The Service Modeler window opens to the same display as when you create a new service.
2. Make changes to the service model using the applicable steps in How to Build a Service Model.
3. Click Save, OK.

Perform Copy, Cut, Paste, Delete, or Remove Operations on a Service in the Console

You can perform the following operations on a service or subservice in the Navigation pane to adjust the service hierarchy:

- **Cut**
  Removes a subservice from the Navigation pane and places it in the paste buffer.
- **Remove**
  Detaches a subservice from its parent service. The subservice moves up one level in the tree.
- **Copy Service**
  Places a service in the paste buffer and leaves it in the Navigation pane.

  NOTE
  You can copy monitored services only.
- **Paste**

Adds a cut or copied service as a subservice to another service.

> **NOTE**
> If you copy and paste a top-level service, the original service is removed from its original top level and pasted as a subservice. However, if you paste a copied subservice, the original subservice appears in its original location and the new location.

- **Delete**
  Removes a service from CA SOI permanently. Any subservices are not deleted, but are promoted to the next highest level in the tree in the Navigation pane.

> **NOTE**
> You can also perform all service hierarchy operations from the List tab of the Contents pane for service CIs.

To cut and paste a service

1. Open the Operations Console and right-click a subservice in the Navigation pane, and select Cut.
   A confirmation dialog opens.
2. Click Yes.
   The subservice is placed in the paste buffer and removed from its parent service. If the service has no other parent, it appears in the top-level Services node. When you paste it in another service, it disappears from the top-level Services folder.
3. Right-click another service under which you want to relocate the cut service, and select Paste.
   A Paste dialog opens if the subservice has relationship properties that you customized when it was a part of the previous parent service.
4. Click yes to preserve the previous relationships or no to erase the previous relationships.
   The service becomes a subservice of the selected parent service.

To remove a subservice

1. Open the Operations Console, right-click a sub-service in the Navigation pane, and select Remove.
   A confirmation dialog opens.
2. Click Yes.
   The subservice is detached from the parent service. If a removed subservice does not have other parents, it moves under the top-level Services node in the tree.

To copy and paste a service

1. Open the Operations Console, right-click a service in the Navigation pane, and select Copy.
   The service is placed in the paste buffer and remains in the Navigation pane.

   > **NOTE**
   > The contents of the buffer are erased if a Paste operation does not immediately follow.
2. Right-click another service under which you want the copied service, and select Paste.
   The service becomes a subservice of the selected parent service.

To delete a service

1. Open the Operations Console, right-click a service in the Navigation pane, and select Delete.
   A confirmation dialog opens.
2. Click Yes.
   The service is deleted from CA SOI.

Set Service Priority

Priority indicates the importance of a service to the business.

Priority applies to the Dashboard, where you can sort services by this value. Priority is not used in any impact calculations, and all services have no priority value by default.

Follow these steps:

1. Open the Operations Console and select a service.
   The alerts for that service appear in the Contents pane, and general service information appears in the Information tab of the Component Detail pane.
2. Click *set* next to Priority in the General Information area of the Information tab, and select one of the following priorities from the drop-down list:
   - None (0)
   - Low (7)
   - Medium (8)
   - High (9)
   - Critical (10)

Set the Service Location for Google Earth

You can associate a service with a location and then use Google Earth to view it based on the specified location.

If you enter a city, Google Earth shows the service in the center of the city. If you enter a street address (for example, 710 Ashbury, San Francisco, CA) Google Earth shows the service at that location if Google Maps can validate the address. You can include additional information (for example, Building 12 or Floor 3) as long as Google Maps can resolve it as a valid address.

You can view Google Earth with mapped CA SOI services from the Dashboard.

Follow these steps:

1. Open the Operations Console, and click a service in the Navigation pane on the left side.
   The alerts for that service appear in the Contents pane, and general service information appears in the Information tab of the Component Detail pane.
2. Click *set* next to Location in the General Information section of the Information tab.
   A text box opens.
3. Enter a location and press Enter. The location can be as broad as a country, or as specific as street address with city and state.
   The location value appears on the Information tab.

Control User Group Service Access

You can grant or remove user group access to individual services from the Operations Console.

The default user groups have access to all services unless you remove all access from the Users tab. You can only change individual service access for groups that do not have access to all services.

To add user group service access

1. Select a service in the Operations Console.
   The alerts for that service appear in the Contents pane, and general service information appears in the Information tab of the Component Detail pane.
2. Scroll to the Security section.
   This section displays the user groups that can currently access the service.
3. Click Add

   .

   The Select Users/Groups dialog opens and displays all user groups that do not currently have access to the service.
4. Select a user group and click OK.
   The user group appears in the Security table, and its users can now view the service in the Operations Console.

To remove user group service access

1. Select a service in the Operations Console.
   The alerts for that service appear in the Contents pane, and general service information appears in the Information tab of the Component Detail pane.
2. Scroll to the Security section.
   This section displays the user groups that can currently access the service.
3. Select a user group from the table and click Remove

   The Remove dialog opens.
4. Click Yes to confirm the removal.

   NOTE
   A dialog opens if a user group has access to all services by default. You must change this setting in the Users tab before you remove access for a specific service.

   The user group disappears from the table and can no longer see the service in the Operations Console.

Change Multiple CI Relationships Simultaneously

You can select multiple CI relationships and change them all simultaneously in the Service Modeler. This way, you can apply the same relationship type to multiple CIs simultaneously; you do not need to individually change each relationship.

Follow these steps:

1. Open the Operations Console and click the Services tab.
   A list of services displays.
2. Select the service for which you want to change multiple CI relationships.
3. Right-click the service and select Edit Service from the context menu.
   The Service Modeler window opens and displays the service topology in the Topology pane.
4. Select all CIs (excluding the parent CI) for which you want to change the relationship.
5. Right-click the parent CI and select Establish Relationships, *<relationship type>* from the context menu.
   The Relationship Edit dialog opens.
6. Click Yes.
   A confirmation message appears.
7. Click OK.
   The message window closes.
8. Click Save, then click OK in the Service Modeler.
   All relationships for the selected CIs are changed and the Service Modeler closes.

Importing Services

Contents

As an administrator, you can use CA SOI to import services from one or more domain managers that CA SOI monitors. For example, CA Spectrum or CA CMDB services can become CA SOI services. Importing services lets you quickly populate CA SOI with service models that leverage existing technology. After the import, you can update the services to add more properties or CIs, or you can leave them as is.

You can import services in two ways:

- Import Services Automatically
- Import Services Manually

If you remove a service from a domain manager, it is not automatically deleted from CA SOI. If you no longer want to manage the service in CA SOI, delete the service manually in CA SOI. If auto-import is on, services that are deleted from CA SOI but not from the domain manager import again. You can also manually reimport any service that you delete from CA SOI that still exists in the domain manager.

Import Services Automatically

Automatic import type imports all services from the connector when CA SOI starts. Automatic import is useful when domain managers update regularly. However, this type of import is not advisable when domain managers are stable, because automatic import uses a considerable amount of resources.

Follow these steps:

1. Open the Operations Console, and select Tools, Import Services.
   The Configure Data Sources dialog opens and displays currently running connectors.

   NOTE
   Each connector entry contains a five-digit identification number defined by the USM schema. For a list of connectors and their ID numbers, see Connector Identification Numbers.
2. Select the connectors whose services you want to load automatically. They would have *No* in the Auto Import column.
3. Click Auto on the toolbar.
   No changes to *Yes* in the Auto Import column.
4. Click Save or OK.
   All services from the connector are automatically imported, and a new import occurs each time CA SOI starts.

Import Services Manually

Manual import type imports specific services that you select. Manual import is recommended if new services were added to the source application during the current session. Manual is the default import type.

Follow these steps:

1. Open the Operations Console, and select Tools, Import Services.
   The Configure Data Sources dialog opens and displays currently running connectors.

   NOTE
   Each connector entry contains a five-digit ID number defined by the USM schema. For a list of connectors and their ID numbers, see Connector Identification Numbers.
2. Select the connector whose services you want to load manually and click Import.

   NOTE
   You cannot import services from connectors that have *Offline* in the Connector Status column.

   The Import Services dialog opens and lists available services in the selected connector. Services that have already been imported contain a check mark in the Exist in SOI column.
3. (Optional) Enter text in the Filter field to limit the number of services displayed.
   The number of listed services decreases.
4. Complete one of the following actions:
   – Click Add All Services to move all services.
   – Select the services you want to import, and click the right arrow.
   The services move to the right pane of the dialog.
5. Click OK.
   The Import Services dialog closes. The Import Status column of the Configure Data Sources dialog displays *Waiting* while the console connects to the source. The same column displays *Importing* while the import is in progress. When the import finishes, the column value changes to *Idle*, and the services appear in the Navigation pane.

Processing Updates for an Imported Service

After you import a service model, connectors send updates as they occur in the domain manager. The following actions occur:

- CIs are added to a service when they are added to the corresponding domain manager service.
- CIs are removed from a service when all relationships that connect CIs to other CIs in the service are removed.

Launch the Source of Imported Services

You can launch the source domain managers of services imported to CA SOI.

Follow these steps:

1. Open the Operations Console, and right-click a service in the Navigation Pane.
   A shortcut menu opens.
2. Select Launch, and select the domain manager to open.
   The source domain manager interface starts.

Service Discovery

Service Discovery is a CA SOI connector that publishes its data to CA SOI as an ordinary connector running in CA SOI Integration Services. You can use a wizard to define policies that automatically create and maintain services and relationships between source and target CIs according to specified criteria. Service Discovery searches the Persistent Store for all the data that is collected from connectors for CIs that match the policy criteria you define. Service Discovery then creates services and relationships using those CIs.

Service Discovery runs as a part of Integration Services as a connector. The Service Discovery connector appears on the Administration Pages panel among other connectors and the connector can be started or stopped from here.

You can install the Service Discovery connector on the same computer as CA SOI Integration Services and the SA Manager or on a different computer. It is possible to have only one Service Discovery connector in the whole CA SOI solution.

Service Discovery is useful when:

- You want to maintain a collection of CIs that share common qualities.
- Some areas of your enterprise are constantly changing, and you want to maintain a service dynamically without having constantly to add CIs manually.
- Certain relationships occur repeatedly in your enterprise, and manually establishing those relationships would take a significant amount of time.

The following policy types are available:

> **NOTE**
> Service Discovery policies support only USM-Core CIs.

- Dynamic service policy
  Automatically discovers CIs matching specified USM types and attribute criteria and places them under a specified service.
- Relationship policy
  Automatically creates relationships that are based on source and target CI types and match criteria.
- Unmanaged relationship policy
  Discovers unmanaged relationships and creates managed relationships (with the same source and target CIs) based on the specified filtering criteria match and whether the source CI is part of the service.

NOTE
A subservice does not automatically inherit the scope of the parent service. If you want to scope an Automatic Relationship or Unmanaged Relationship policy to a parent service and any of the parent's child services, you must explicitly add the child services to the scope list in the policy.

How to Create Dynamic Service Policies

Contents

As an administrator, you can create dynamic service policies that automatically include resources in a new dynamic service based on the criteria that you define. You can use regular expressions and test your expressions with the Regex Tester. The Service Discovery Policy Editor lets you create and manage Service Discovery policies based on the following:

- **Relationship**
  Defines the relationship to establish between CIs that meet the policy criteria and the service CI.
- **Type**
  Defines the USM type on which to apply the policy.

For example, you can create a Service Discovery policy to match all ComputerSystem CIs within a certain IP address range and add those CIs to a service. When you open the Service Discovery Policy Editor, you can create a policy for one relationship and type. To define multiple policies for one dynamic service, you must go through the wizard multiple times.

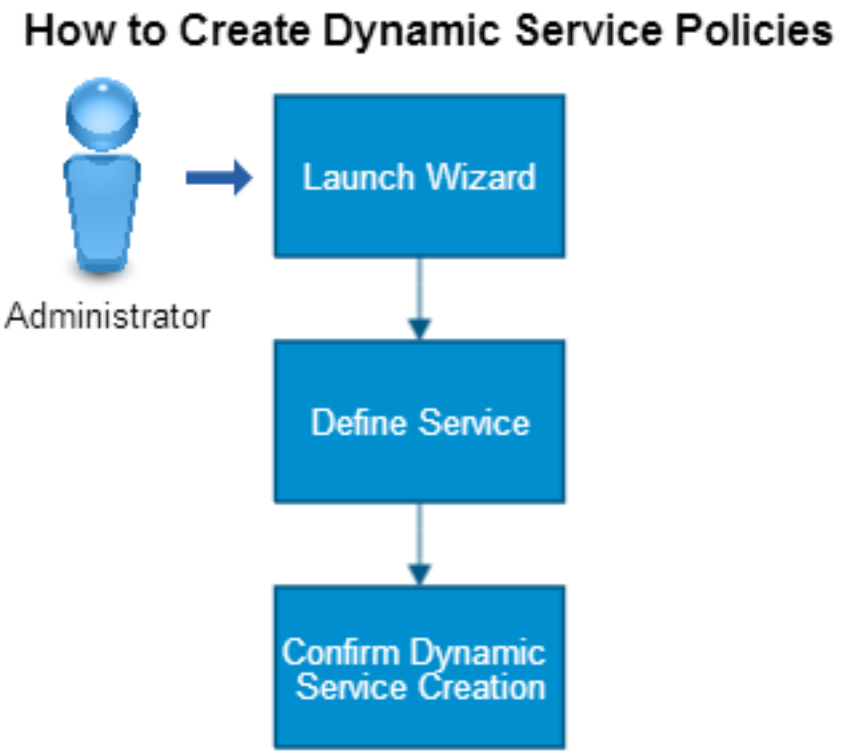
The Service Discovery engine evaluates the Persistent Store for CIs that match the Service CI relationship and attribute criteria. Every CI that matches is then added to the service. The evaluation is continuous, so the service is updated dynamically when matching changes in the Persistent Store occur.

You can create one policy at a time for a relationship. Multiple policies for each relationship are supported, but you must create them in separate operations. Multiple dynamic service policies can contribute to the same service as long as they share the name of the service.

NOTE
Service Discovery policies support only USM-Core CIs.

Use this scenario to guide you through the process:

Figure 30: how to create dynamic service policies



1. Launch the Wizard.
2. Define the Service.
3. Confirm the Dynamic Service Creation.

You can also read this scenario that provides an example for creating dynamic service policies.

Launch the Wizard

You use the Service Discovery Policy Editor to create and manage dynamic services and relationship criteria.

> **NOTE**
> Page fields marked with an asterisk (*) are required.

Follow these steps:

1. Open the Operations Console.
2. Select Tools, Service Discovery Policies.
   The Service Discovery Policy Editor opens.

   > **NOTE**
   > System-wide, only one user can access the Service Discovery Policy Editor at a time.

   The Policies tab provides a tree that displays the Dynamic Services, Automatic Relationships, and Unmanaged Relationships available.
3. Right-click Dynamic Services in the Policies tab and select Create.

Define the Service

The wizard guides you through the process of service creation.

Follow these steps:

1. Specify a name for the service in the Service Name field.
2. The Define Service page lets you name the new dynamic service and assign user groups to the service. Assigning user groups for the service lets Service Discovery automatically assign the same user access to the services when it (Service Discovery) creates them. This way, you enhance the security of the service, as only authorized users are able to access the service.
   Click Next.
3. On the Relationship Criteria page, select a USM relationship type from the Relationship Type drop-down list. This relationship type defines the relationship between the service CI and CIs that meet the policy criteria.
   The Target pane lets you build criteria using Boolean expressions.
4. Select a USM type from the Class drop-down list in the Target pane.
   Consider the following:
   – The Attribute drop-down list displays a subset of the USM properties available for the selected type. Certain attributes such as MdrProduct attributes and date/time data types are not supported and do not appear in the drop-down.
   – When you select a USM type, all available derived types in the USM hierarchy are also included. For example, when you select the ComputerSystem type, its child type VirtualSystem is automatically included in the policy.
   – For USM property definitions, see How to Access the USM Schema Documentation.
5. (Optional) Select a USM attribute, comparison type, and attribute value for the expression. If you want to add all CIs of the selected type without adding attribute criteria, skip to Step 6.

   NOTE
   You can use regular expressions by selecting either Matches regex or Does not match regex from the Comparison Type drop-down list. Regular expressions are not available for all attributes. Click Test Regex to open the Regex Tester and test the regular expression against a string. For information about using regular expressions in other CA SOI features, see Regular Expressions.

   WARNING
   If you do not add attribute criteria, you risk discovering a large number of CIs, which can compromise system performance.

   NOTE
   For attribute values, mouse over the field and a tooltip displays the data type required.
6. Click Add.
   The criteria are added to the expression, which appears in the logic tree and as an expression in the field at the bottom of the page.
7. (Optional) Repeat Steps 3 - 6 to add additional criteria and logic.

   NOTE
   For more information about using advanced logic with multiple target criteria, click the Hints link above the logic tree.
8. Click Next.
   The Confirm page opens. A Topology Warning dialog may appear if the Service Discovery editor detects that the policy might cause topology errors or unmanageable services. Evaluate the policy before proceeding to verify whether changes are required. For more information, see Topology Warnings.

Regular Expressions Considerations

Consider the following situations when using Regular Expressions in CA SOI:

Regular Expressions in CA SOI act as a *find,* not as a *match.*

A *find* searches for the pattern across the strings, including the substrings.

A *match* searches for the pattern in the strings only.

Example:

With the following strings: "cart" and "artistic":

- A *find* for the "art" pattern finds "art" in the substrings of both the "cart" and "artistic" strings.
- A *match* for the "art" pattern does not match in "cart" or "artistic" strings because a *match* does not search the substrings.

To perform a *match* in CA SOI, enclose the pattern with "^" and "$", such as "^art$". You can use the Regex Tester to verify your expressions before implementing them.

Use the Regular Expression (Regex) Tester

You can use the Regular Expression (Regex) Tester to validate a regular expression before using the expression in CA SOI.

Follow these steps:

1. In a dialog that supports regex, click Test Regex.
   The Regex Tester dialog appears.
   The Operation field describes the conditions:
   – Case sensitivity
   – If the pattern is to match or not match
   If you entered an expression in the Attribute Value field, the expression appears in the Regex Pattern for editing.
2. Enter (or edit) the regular expression in the Regex Pattern field.
   The Valid? field indicates if the expression you entered is a valid regular expression. The field displays Yes or No.
3. Enter a test string in the Test Text field.
   The Found?/Not Found? field indicates if the regular expression finds or does not find the Test Text string, based on the Operation conditions. The field displays True or False.
4. Perform one of the following actions:
   – Click Use Pattern to close the RegEx Tester dialog and transfer the Regex Pattern to the Attribute Value field.
   – Click Cancel to close the RegEx Tester dialog and leave the Attribute Value field unchanged.

Confirm the Dynamic Service Creation

The Confirm page displays the policy expression for the dynamic service policy criteria you created, displays warnings about policies that could create topology errors or large services, and lets you confirm creation of the new dynamic service policy. The page also displays the user groups that you assign to the service.

Follow these steps:

1. Click Finish to confirm creation of the new dynamic service policy.
   The service name appears under the Dynamic Services folder. Expand the service name to view the relationship and type defined in each policy for the service. You can define additional policies for the service through the right-click menu using the same or new relationships.

   NOTE
   If you receive a topology warning, you should evaluate to determine whether changes are required. For more information, see Topology Warnings.
2. Click Save or OK when the dynamic service policies are complete.
   The Service Discovery engine begins scanning the Persistent Store for CIs that match the policy criteria.

How to Create Automatic Relationship Policies

Contents

As an administrator, you use the Service Discovery Policy Editor to create and manage policies that automatically create relationships based on policy criteria.

Automatic relationship policies are based on the following:

- **Relationship**
  Defines the type of relationship to create.
- **Source CI**
  Defines the CI type to use as the source of the relationship with optional attribute-based criteria.
- **Target CI**
  Defines the CI type to use as the target of the relationship with optional attribute-based criteria.
- **Match Criteria**
  Defines how the source and target CIs must relate (based on attribute values) for the policy criteria to match and create a relationship between the CIs.
- **Scope**
  Defines under which services the relationships are created.

The automatic relationship policy creates a new relationship in the service that automatically adds the target CI to the service. Only the source CI needs to pre-exist in the service.

A CI can become part of the service through various ways; for example:

- Imported service already includes the CI in it.
- CI is manually added to the service.
- Another Service Discovery policy (for example, a dynamic service policy) adds the CI to services.

> **NOTE**
> When a policy specifies a class name (for example, class=ComputerSystem), the policy applies to the CIs of that class and also to its subclasses (as defined by the USM Schema Type hierarchy). For example, in the case of ComputerSystem, the policy would also apply to VirtualSystems.

When the specified source CI appears in a service and the policy criteria are met, Service Discovery adds a relationship to all the specified target CI in the service. You can scope the policies to a subset of service models or apply them to all service models in which the source CI exists.

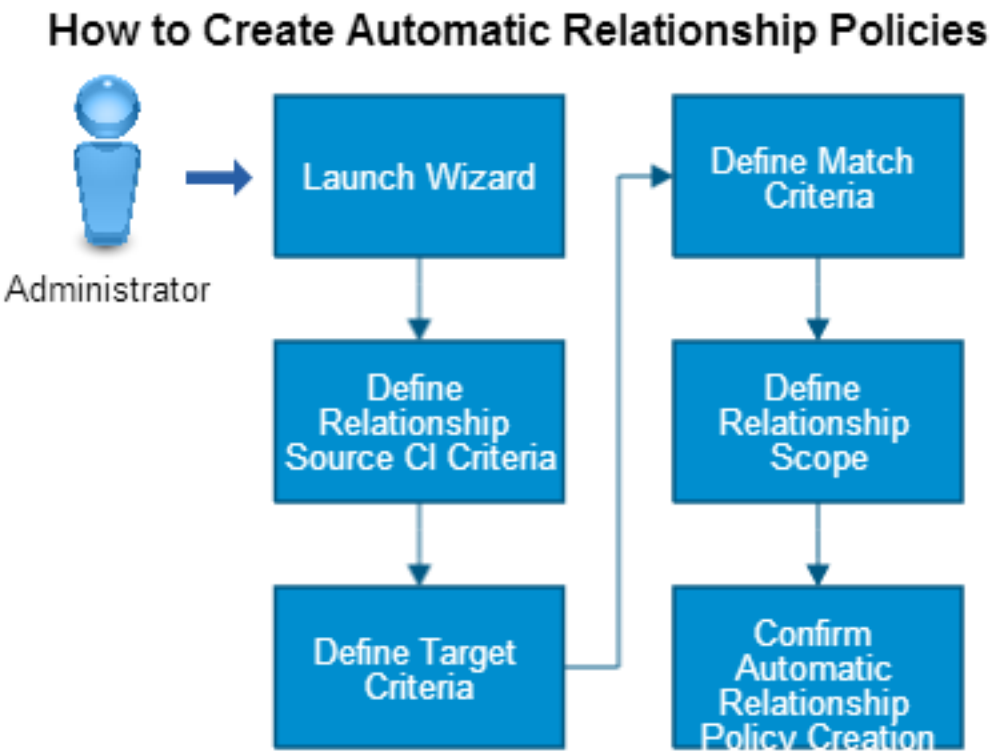
For example, you can create an automatic relationship policy that creates relationships between ComputerSystem CIs and all RunningSoftware contained on the ComputerSystem based on matching ComputerSystem's and RunningSoftware's IP addresses.

You can create one policy at a time for a relationship. Multiple policies for each relationship are supported, but you must create them in separate operations.

> **NOTE**
> Service Discovery policies support only USM-Core CIs.

Use this scenario to guide you through the process:

Figure 31: how to create automatic policies



1. Launch the Wizard.
2. Define the Relationship and Source CI Criteria.
3. Define the Target CI Criteria.
4. Define the Match Criteria.
5. Define the Relationship Scope.
6. Confirm the Automatic Relationship Policies Creation.

You can also read a scenario that provides an example for creating automatic relationship policies.

Launch the Wizard

You use the Service Discovery Policy Editor to create and manage automatic relationship policies.

Follow these steps:

1. Open the Operations Console.
2. Select Tools, Service Discovery Policies.
   The Service Discovery Policy Editor opens.

   NOTE
   System-wide, only one user can access the Service Discovery Policy Editor at a time.

   The Policies tab provides a tree that displays the Dynamic Services, Automatic Relationships, and Unmanaged Relationships available.

3. Right-click Automatic Relationships in the Policies tab and select Create.
   The wizard opens on the Source Criteria page.

Define the Relationship and Source CI Criteria

The Source Criteria page lets you define the relationship type and the source CI criteria.

Follow these steps:

> **NOTE**
> Page fields marked with an asterisk (*) are required.

1. Select a USM relationship type from the Relationship Type drop-down list.
   Once you select a relationship type, the Relationship Type field becomes read-only in subsequent pages and the relationship expression builds as you select the target and match expression.
2. Select the class (USM type) of the source CI from the Class drop-down list in the Source pane.
   Consider the following:
   – The Attribute drop-down list displays a subset of the USM properties available for the selected type. Certain attributes such as MdrProduct attributes and date/time data types are not supported and do not appear in the drop-down.
   – When you select a USM type, all available derived types in the USM hierarchy are also included. For example, when you select the ComputerSystem type, its child type VirtualSystem is automatically included in the policy.
   – For USM property definitions, see the USM Schema Documentation.
3. (Optional) Select a USM attribute, comparison type, and attribute value for the expression. If you want to add all CIs of the selected type without adding attribute criteria, skip to Step 6.

   > **NOTE**
   > For attribute values, mouseover the field and a tooltip displays the data type required.
4. Click Add.
   The criteria are added to the expression, which appears in the logic tree and as an expression in the field at the bottom of the page.
5. (Optional) Repeat Steps 2 - 4 to add additional criteria and logic.
   Note: For more information about using advanced logic with multiple source criteria, click the Hints link.
6. Click Next.

Define the Target CI Criteria

The Target Criteria page lets you define criteria for the target CI in the relationship.

Follow these steps:

> **NOTE**
> Page fields marked with an asterisk (*) are required.

1. Select the USM type of the target CI from the Class drop-down list in the Target pane.
   Consider the following:
   – The Attribute drop-down list displays a subset of the USM properties available for the selected type. Also, certain attributes such as MdrProduct attributes and date/time data types are not supported and do not appear in the drop-down.
   – When you select a USM type, all available derived types in the USM hierarchy are also included. For example, when you select the ComputerSystem type, its child type VirtualSystem is automatically included in the policy.
   – For USM property definitions, see the USM Schema Documentation.
2. (Optional) Select a USM attribute, comparison type, and attribute value for the expression. If you want to add all CIs of the selected type without adding attribute criteria, skip to Step 5.

NOTE
For attribute values, mouseover the field and a tooltip displays the data type required.

3. Click Add.
   The criteria are added to the expression, which appear in the logic tree and as an expression in the field at the bottom of the page. CIs of the specific type must meet the attribute-based criteria to match.
4. (Optional) Repeat Steps 2 - 3 to add additional criteria and logic. CIs of the specified type must meet the attribute-based criteria to match.

   NOTE
   For more information about using advanced logic with multiple target criteria, click the Hints link.
5. Click Next.

Define the Match Criteria

The Match Criteria page lets you define a comparison between attributes of your source and target.

Follow these steps:

NOTE
Page fields marked with an asterisk (*) are required.

1. Select a source attribute, comparison type, and target attribute from the drop-down lists as follows:
   - *'Source_CI'* Attribute
     Defines the USM attribute of the source CI that must have some relation to a target CI attribute
   - **Comparison Type**
     Defines how the specified source and target CI attributes must relate. The options in this drop-down list change based on the data type of the selected attributes.
   - *'Target_CI'* Attribute
     Defines the USM attribute of the target CI that must have some relation to a source CI attribute.

     NOTE
     For USM property definitions, see the USM Schema Documentation.
2. Click Add.
   The criteria are added to the expression, which appears in the logic tree and as an expression in the field at the bottom of the page.

   NOTE
   If you define criteria that conflict with the potential values of the selected attributes (for example, if you select Equal To for two enumerated properties that do not share values), an error message opens.
3. (Optional) Repeat Steps 1 - 3 to add additional conditions and logic.

   NOTE
   For more information about using advanced logic with multiple match criteria, click the Hints link.
4. Click Next.

Define the Relationship Scope

The Relationship Scope page lets you define the scope of the relationship, which defines which services are affected when creating the relationships.

Follow these steps:

1. Select a scope:
   - **All services**
     Creates the relationship in all services of which the source CI is a part.
   - **Specific services**

Creates the relationship in the specified services of which the source CI is a part.

2. (Specific services only) Do any of the following:
   - Click Add and use the Locator tool to select additional services by optionally filtering services, selecting the services you want to include, clicking Add, and then clicking OK.
   - In the Locator tool, select a service from the scope list and click Remove to remove the service from the scope list.
   - In the Locator tool, click Clear to clear the service scope list.

     NOTE
     If you add a dynamic service to the scope that you later rename, you must manually add the renamed dynamic service to the scope again.
3. Click Next.
   The Confirm page opens. A Topology Warning dialog may appear if the Service Discovery editor detects that the policy might cause topology errors or unmanageable services. Evaluate the policy before proceeding to verify whether changes are required. For more information, see Topology Warnings.

Confirm the Automatic Relationship Policies Creation

The Confirm page displays the policy expression for the automatic relationship criteria you created.

Follow these steps:

1. Click Finish to confirm creation of the new automatic relationship policy.
   The new policy appears under the Automatic Relationships folder. Expand the semantic type to view the relationship policy. You can easily define additional policies for the semantic by right-clicking the semantic and selecting Add.

   NOTE
   If you receive a topology warning, check whether changes are required. For more information, see Topology Warnings.
2. Click OK or Save when the automatic relationship policies are complete.
   The Service Discovery engine begins scanning the Persistent Store for CIs that match the policy criteria.

How to Create Unmanaged Relationship Policies

Contents

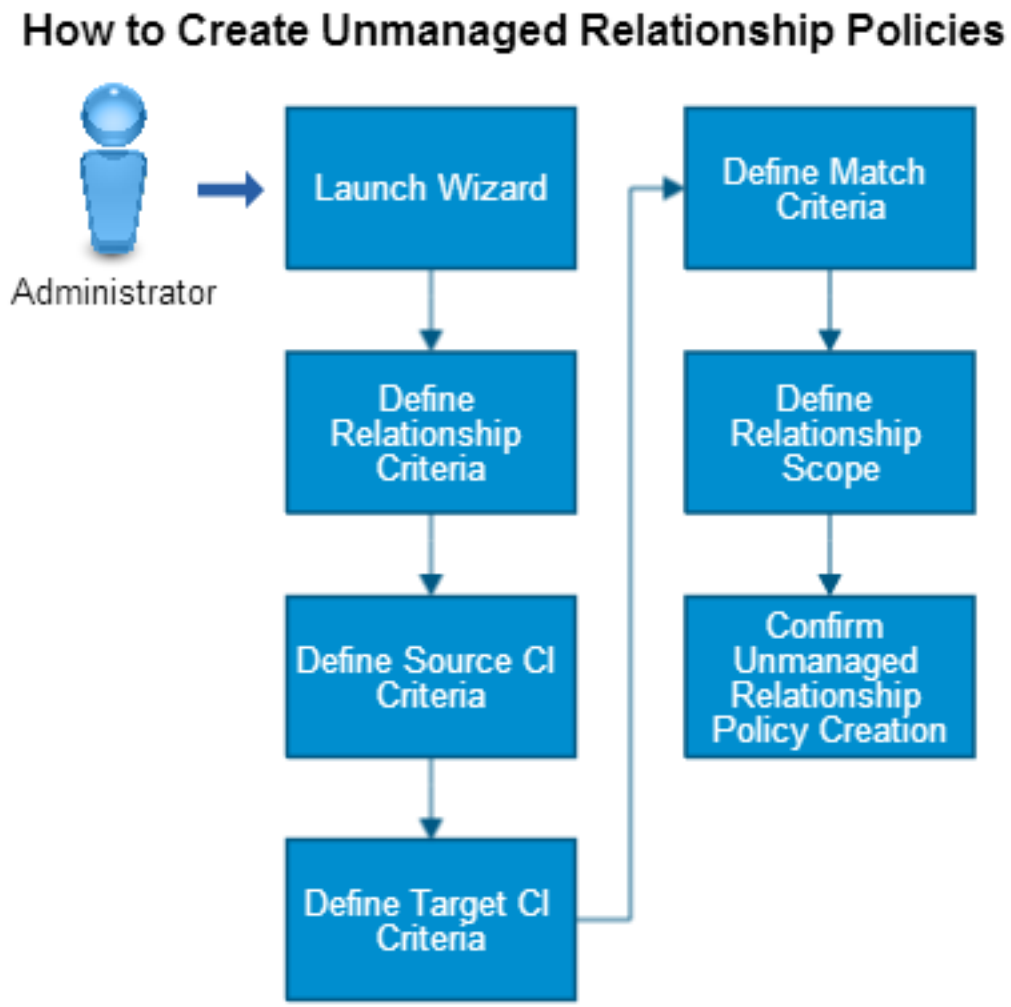
As an administrator, you use the Service Discovery Policy Editor to create an unmanaged relationship policy. An unmanaged relationship is a relationship that exists between two CIs in the CA Catalyst Persistent Store but is not reflected in a CA SOI service model. The unmanaged relationship policy discovers and evaluates unmanaged relationships based on the filtering criteria and whether a source CI is part of a specific or any service. If the filtering criteria match exists, the policy creates a same, but managed, relationship. The newly created managed relationship has the same source and target CIs. This relationship is scoped to all services the source CI is part of. The semantic of the new relationship can be same or different from the original unmanaged relationship depending on the policy definition.

NOTE
If the source CI is part of multiple services, the policy creates as many relationships between source and target CIs, with each relationship assigned to the appropriate service.

NOTE
Service Discovery policies support only USM-Core CIs.

Use this scenario to guide you through the process:

Figure 32: how to create unmanaged policies



1. Launch the Wizard.
2. Define the Relationship Criteria.
3. Define the Source CI Criteria.
4. Define the Target CI Criteria.
5. Define the Match Criteria.
6. Define the Relationship Scope.
7. Confirm the Unmanaged Relationship Policies Creation.

You can also read a scenario that provides an example for creating unmanaged relationship policies.

Launch the Wizard

Use the Service Discovery Policy Editor to create unmanaged relationship policies.

Follow these steps:

1. Open the Operations Console.
2. Select Tools, Service Discovery Policies.
   The Service Discovery Policy Editor opens.

   NOTE
   System-wide, only one user can access the Service Discovery Policy Editor at a time.

   The Policies tab provides a tree that displays the Dynamic Services, Automatic Relationships, and Unmanaged Relationships.
3. Right-click Unmanaged Relationships in the Policies tab and select Create from the context menu.
   The wizard opens on the Relationship Criteria page.

Define the Relationship Criteria

The Relationship Criteria page lets you define the name of the unmanaged relationship policy and filtering criteria that you want to use to discover an unmanaged relationship. The page also lets you specify the type of managed relationship that you want to create after discovering the unmanaged relationship.

Follow these steps:

NOTE
Page fields marked with an asterisk (*) are required.

1. Specify a name for the policy in the Policy Name field. Ensure that the policy name is different from the existing policy names.
2. Select the Reverse the created relationships option if you want the source and target CIs to swap their positions in the created managed relationship. This implies that the source CI in the unmanaged relationship becomes the target CI in the created managed relationship; similarly, the target CI becomes the source CI.
   For example, consider a scenario where a connector publishes unmanaged relationships *IsHostedBy*. These relationships have virtual systems as source CIs and VMWare hosts as target CIs. The service you are populating must have reversed semantic *IsHostFor*, where the sources are VMWare hosts and targets are virtual systems. In such scenarios, enable the Reverse the created relationships option to reverse the source CI and target CI positions in the managed relationships.

   NOTE
   In a typical scenario, an unmanaged relationship policy creates a managed copy of the discovered unmanaged relationship. The created managed relationship has the same semantic (if the *Same as Discovered* option is used in Step 3), source CI, and target CI as the original unmanaged relationship. However, if you want to swap the source and target CIs positions in the created managed relationship, use this option.
3. Select a USM relationship type from the Create Relationship of Type drop-down list.
   This value specifies the type of managed relationship that you want to create for the discovered unmanaged relationship.

   WARNING
   Select *Same as Discovered* if you want to create a managed relationship of the same type as the discovered relationship.

   For USM property definitions, see the USM Schema Documentation.
4. Select the relationship type that you want to discover for identifying the unmanaged relationships.

   NOTE
   Select at least one option; you cannot leave this field blank.
5. Use the Add Selected arrow key to move the selected type from Available Types to Selected Types.
6. Click Next.

Define the Source CI Criteria

The Source Criteria page lets you define the source CI criteria for the unmanaged relationship.

Follow these steps:

> **NOTE**
> Page fields marked with an asterisk (*) are required.

1. Select the USM class (type) of the source CI from the Class drop-down list in the Source pane.

   > **NOTE**
   > Selecting *Entity* as the USM type matches all CI types.

   Consider the following:
   - The Attribute drop-down list displays a subset of the USM properties available for the selected type. Certain attributes such as MdrProduct attributes and date/time data types are not supported and do not appear in the drop-down.
   - When you select a USM type, all available derived types in the USM hierarchy are also included. For example, when you select the ComputerSystem type, its child type VirtualSystem is automatically included in the policy.
   - For USM property definitions, see the USM Schema Documentation.
2. (Optional) Select a USM attribute, comparison type, and attribute value for the expression. If you want to add all CIs of the selected type without adding attribute criteria, skip to Step 5.
3. Click Add.
   The criteria are added to the expression, which appears in the logic tree and as an expression in the field at the bottom of the page.
4. (Optional) Repeat the steps to add additional criteria and logic.

   > **NOTE**
   > For more information about using advanced logic with multiple source criteria, click the Hints link.
5. Click Next.
   The Target Criteria page opens.

Define the Target CI Criteria

The Target Criteria page lets you define the filtering criteria for the target CI in the relationship.

Follow these steps:

> **NOTE**
> Page fields marked with an asterisk (*) are required.

1. Select the USM type of the target CI from the Class drop-down list in the Target pane.

   > **NOTE**
   > Selecting *Entity* as the USM type matches all CI types.

   Consider the following:
   - The Attribute drop-down list displays a subset of the USM properties available for the selected type. Also, certain attributes such as MdrProduct attributes and date/time data types are not supported and do not appear in the drop-down.
   - When you select a USM type, all available derived types in the USM hierarchy are also included. For example, when you select the ComputerSystem type, its child type VirtualSystem is automatically included in the policy.
   - For USM property definitions, see the USM Schema Documentation.
2. (Optional) Select a USM attribute, comparison type, and attribute value for the expression. If you want to add all CIs of the selected type without adding attribute criteria, skip to Step 5.
3. Click Add.

The criteria are added to the expression, which appears in the logic tree and as an expression in the field at the bottom of the page. CIs of the specific type must meet the attribute-based criteria to match.

4. (Optional) Repeat the steps to add additional criteria and logic. CIs of the specified type must meet the attribute-based criteria to match.

   NOTE
   For more information about using advanced logic with multiple target criteria, click the Hints link.

5. Click Next.

Define the Match Criteria

The Match Criteria page lets you define a comparison between attributes of your source and target CIs.

Follow these steps:

1. Select a source attribute, comparison type, and target attribute from the drop-down lists as follows:
   - *'Source_CI'* Attribute
     Defines the USM attribute of the source CI that must have some relation to a target CI attribute.
   - **Comparison Type**
     Defines how the specified source and target CI attributes must relate. The options in this drop-down list change based on the data type of the selected attributes.
   - *'Target_CI'* Attribute
     Defines the USM attribute of the target CI that must have some relation to a source CI attribute.

   NOTE
   For USM property definitions, see the USM Schema Documentation.

2. Click Add.
   The criteria are added to the expression, which appears in the logic tree and as an expression in the field at the bottom of the page.

   NOTE
   If you define criteria that conflict with the potential values of the selected attributes (for example, if you select Equal To for two enumerated properties that do not share values), an error message opens.

3. (Optional) Repeat the steps to add additional conditions and logic.

   NOTE
   For more information about using advanced logic with multiple match criteria, click the Hints link.

4. Click Next.

Define the Relationship Scope

The Relationship Scope page lets you define the scope of the relationship, which defines what services are affected when creating the relationships.

Follow these steps:

1. Select a scope:
   - **All services**
     Creates the relationship in all services of which the source CI is a part.
   - **Specific services**
     Creates the relationship in the specified services of which the source CI is a part.
2. (Specific services only) Do any of the following:
   - Click Add and use the Locator tool to select additional services by optionally filtering services, selecting the services you want to include, clicking Add, and then clicking OK.
   - In the Locator tool, select a service from the scope list and click Remove to remove the service from the scope list.
   - In the Locator tool, click Clear to clear the service scope list.

3. Click Next.
   The Confirm page opens. A Topology Warning dialog may appear if the Service Discovery editor detects that the policy might cause topology errors or unmanageable services. Evaluate the policy before proceeding to verify whether changes are required. For more information, see Topology Warnings.

Confirm the Unmanaged Relationship Policies Creation

The Confirm page displays the unmanaged relationship policy criteria that you created.

Follow these steps:

1. Click Finish to confirm creation of the new unmanaged relationship policy.
   The new policy appears under the Unmanaged Relationships folder. Expand the semantic type to view the relationship policy. You can easily define additional policies for the semantic by right-clicking the semantic and selecting Add.

   NOTE
   If you receive a topology warning, check whether changes are required. For more information, see Topology Warnings.
2. Click OK or Save when the unmanaged relationship policies are complete.
   The Service Discovery engine begins scanning the Persistence Store for relationships matching the policy criteria.

Topology Warnings

Topology warnings indicate that a Service Discovery policy could lead to one of the following topology issues:

- **Orientation**
  Occurs when a relationship points from a CI to a top-level service node. All relationships must orient from the root service to its child items. For example, this error could occur if you create an automatic relationship policy with a service as the target CI.
- **Double relationships**
  Occurs when more than one relationship using different propagation types exists between two CIs or if equivalent policies are applied to the same service. For example, this error could occur if you create a dynamic service with multiple policies using the same target class and different propagation type, or if you create an automatic relationship with multiple policies using the same source and target class with a different propagation type.
- **Cycles**
  Occurs when an item becomes related to itself, two items have separate relationships going in each direction, or when more than two items have a circular relationship structure. For example, this error could occur if you do the following:
  - Create a dynamic service policy with a service CI as the target
  - Create an automatic relationship policy with the same source and target USM type
  - Create an automatic relationship with multiple policies that use the same USM types as source and target. For example, one policy with ComputerSystem as source and BackgroundProcess as target, and another policy with BackgroundProcess as source and ComputerSystem as target.
  - Any automatic relationship has the same source CI as another policys target CI
- **Unmanageable services**
  Occurs when a policy creates the potential for an overwhelming number of CIs to be added to a service. For example, this error could occur if you create a dynamic service policy with only a USM type but no target filter criteria, or you create a dynamic service policy using a type that has several children in the USM type hierarchy. For instance, creating a dynamic service policy based on the USM type entity with no target filter conditions would add CIs of all USM types to a service and potentially impact SA Manager performance.

Topology warnings appear in the following locations:

- Confirm page when saving policy
- Details tab in a TOPOLOGY WARNINGS section when you select a policy from the Policies tab.
- Details tab when you select Dynamic Services, Automatic Relationships, Unmanaged Relationships in the Policies tab.
- Policies tab as red service and policy icons
- Policies tab tooltips

You can switch warning display settings in the Set Preferences dialog.

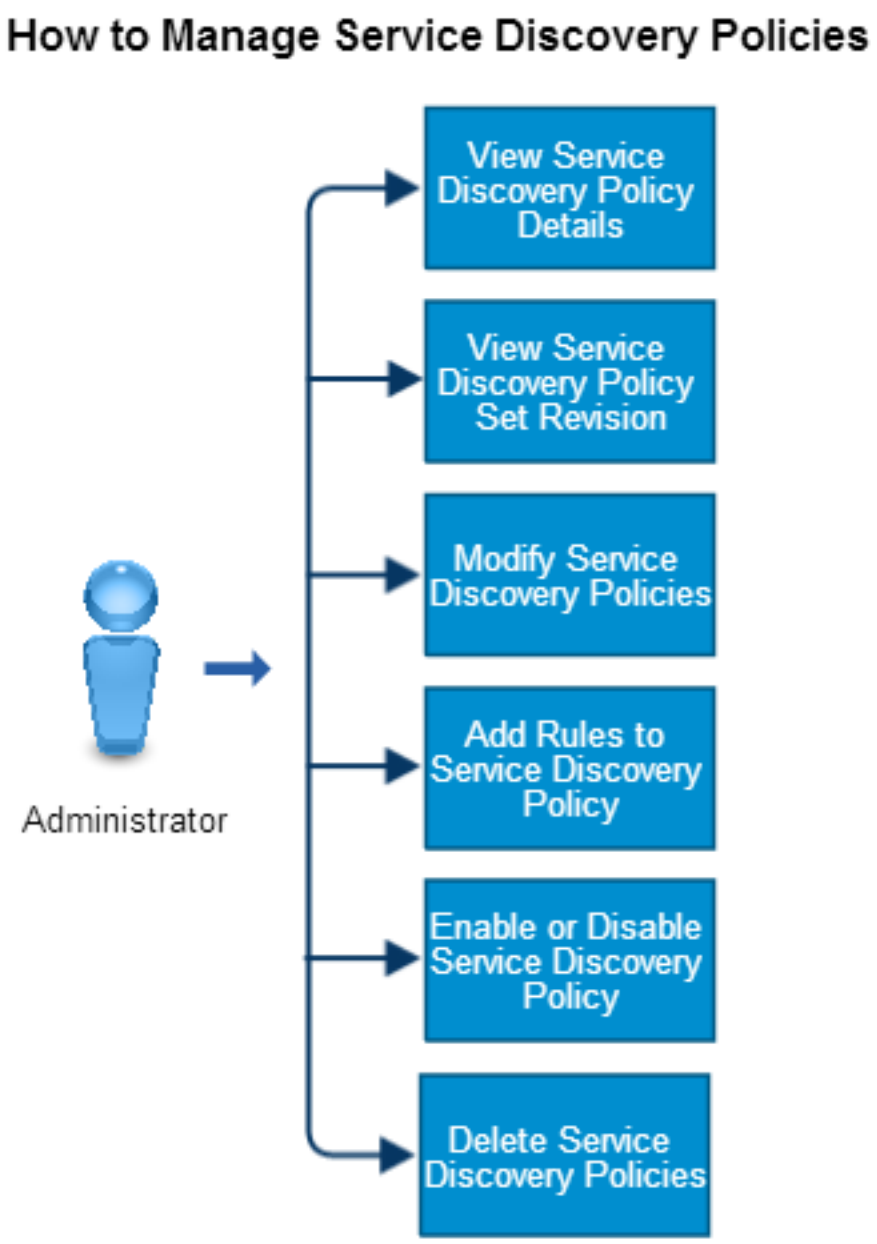
Service Discovery does not prevent the creation of policies that contain warnings. A warning does not mean that the potential problem will definitely occur. If you receive a warning, review the Service Discovery policy to help ensure the policy does not present problems.

How to Manage Service Discovery Policies

As an administrator, you can view, modify, enable, and delete Service Discovery policies. This ability helps you incorporate changed requirements in the policies when required. For example, you can modify policies to add rules to them or delete or disable them if they are not required in the infrastructure.

Use the following scenario to guide you through the process:

Figure 33: how to manage service discovery policies



- View Service Discovery Policy Details.
- View Service Discovery Policy Set Revision Information.
- Modify Service Discovery Policies.
- Add Rules to a Service Discovery Policy.
- Enable or Disable Service Discovery Policies
- Delete Service Discovery Policies.

You can perform these tasks in any sequence.

View Service Discovery Policy Details

You can view all Service Discovery policies on the Service Discovery Policy Editor.

Follow these steps:

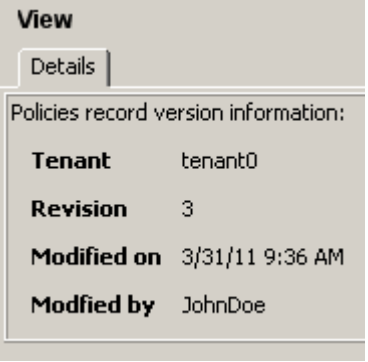
1. Open the Operations Console.
2. Select Tools, Service Discovery Policies.
3. Click a policy in the Policies tab.
   The Details tab displays the policy details and also shows topology warnings that can negatively impact the service topology.

View Service Discovery Policy Set Revision Information

You can view the revision information for the Service Discovery policy set, which informs you of the number of revisions to the policy set, who modified it, and when.

Follow these steps:

1. Select Tools, Service Discovery Policies from the Operations Console.
2. Click Service Discovery Policies in the Policies tab.
   The policy revision information appears in the Details tab.



   - **Tenant**
     This field is for a future release of CA SOI.
   - **Revision**
     Indicates the current revision number of the policy set. Performing a Redo using redo_rules.bat or a Save on the Service Discovery Policy Editor increases the revision number by one. Performing an Undo using undo_rules.bat decreases the revision number by one.
   - **Modified on**
     Indicates the last date and time when the policy set and revision number changed.
   - **Modified by**
     Indicates either a user ID or cmdLine if the last revision was performed using a command line operation.

Modify Service Discovery Policies

You can modify any Service Discovery policy.

> **NOTE**
> If you rename a dynamic service that is part of the service scope for an automatic relationship policy, you must manually add the renamed dynamic service to the scope again.

Follow these steps:

1. Open the Operations Console.
2. Select Tools, Service Discovery Policies.
3. Right-click a policy in the Policies tab and select Modify.
   The Service Discovery Policy Editor opens on the Relationship Criteria page (for a dynamic services policy), Source Criteria page (for an automatic relationships policy), and Relationship Criteria page (for an unmanaged relationships policy).
4. Update the conditions as necessary and complete the wizard pages.
5. Save the policies when prompted.

Add Rules to a Service Discovery Policy

When you create a policy, you can create one rule at a time for a USM type or relationship type. You can add rules to the same policy if necessary in separate operations.

Follow these steps:

1. Do one of the following in the Policies tab of the Service Discovery Policy Editor:
   – Right-click the dynamic service name and select Add to add a rule to the dynamic service based on a different relationship type from the existing rules.
   – Right-click the relationship folder in a dynamic service and select Add to add a rule to the dynamic service using the selected relationship and based on a different USM type from the existing rules.
   – Right-click the automatic relationship name and select Add to add a rule for that relationship type.
   The first page of the wizard appears for creating the appropriate rule.
2. Create the new rule.
   The rule appears in the appropriate place in the associated policy hierarchy in the Policies tab.

Enable or Disable Service Discovery Policies

You can enable or disable specific Service Discovery policies based on your requirements and control the policy execution schedule.

When you disable a policy, it (disabled policy) does not delete any of the relationships that it has already discovered; this is the default behavior. You can configure this default behavior by configuring the value of the *treatDisabledPoliciesAsDeleted* parameter.

This Service Discovery plugin parameter takes *true* or *false* as its value. The default value is *false*, which implies that the relationships that the policy has already discovered are not deleted when the policy is disabled. The value *true* implies that all relationships that the policy has already discovered are deleted when the policy is disabled.

You can update the value of the parameter in the Service Discovery plugin properties file SOI_HOME\ServiceDiscovery \connectivityContext.xml. Restart Integration Services to make the change effective.

Follow these steps:

1. Open the Operations Console.
2. Select Tools, Service Discovery Policies.
3. Select Dynamic Services, Automatic Relationships, or Unmanaged Relationships (as appropriate) in the Policies tab.
   A list of related policies appears in the Details tab. For example, if you select Dynamic Services, all dynamic service policies are displayed in the Details tab.
4. Select the appropriate Service Discovery policy that you want to enable or disable in the Details tab.
5. Click the Enables selected policies icon (plus sign with a tick mark) or Disables selected policies icon (minus sign with a tick mark) to enable or disable the policy.
   The Enabled column displays the updated status. *Yes* implies that the policy is enabled and *No* signifies that it is disabled.

NOTE
By default, the policy is enabled when you create it. However, if require a policy to be disabled from the beginning, disable it after creation, but before saving. This way the policy only starts discovering when you enable it manually later.

Delete Service Discovery Policies

You can delete Service Discovery policies to which you have access privileges.

You can select multiple Service Discovery policies and delete them all at the same time from the Service Discovery Policy Editor.

Follow these steps:

1. Open the Operations Console.
2. Select Tools, Service Discovery Policies.
3. Select Dynamic Services, Automatic Relationships, or Unmanaged Relationships (as appropriate) in the Policies tab. A list of related policies appears in the Details tab. For example, if you select Dynamic Services, all dynamic service policies are displayed in the Details tab.
4. Right-click the policy that you want to delete, and select Delete.
5. Confirm the deletion.

Service Discovery Connector Configuration

In the <SOI_HOME>/ServiceDiscovery directory, you can find the conectivityContext.xml file which contains connector configuration. The file contains three bean definitions:

- **persistenceDB**
  Specifies connectivity to Catalysts Persistent Store database. You can specify driver, URL, database user name, and encrypted password.
- **ssaDB**
  Specifies connectivity to the CA SOI database, which is usually the same as the Persistent Store DB. You can specify driver, URL, database user name, and encrypted password.
- **treatDisabledPoliciesAsDeleted**
  Specifies how the disabled policies are treated. It set to true, then service discovery deletes all relationships that have been created by the disabled policies so far (while they were enabled). If set to true, discovered relationships are not deleted, but new relationships based on the disabled policies are created.
- **tickCounts**
  Specifies the frequency of executing each evaluation algorithm. The 'tick' unit is equal to 5 seconds, thus tick count 6 means 30 seconds.
  - **dynamicGroupTickCount**
    Frequency for Dynamic Service policy evaluation. This parameter evaluates only recently added CIs (uses timestamps).
  - **unmanagedRelationshipTickCount**
    Frequency for Unmanaged Relationship policies evaluation. This parameter evaluates only recently added CIs (uses timestamps).
  - **autoRelationshipTickCount**
    Frequency for Automatic Relationship policies evaluation. This parameter evaluates only recently added CIs (uses timestamps).
    This value is 0 by default, because inMemory evaluation of the automatic relationship is the default algorithm.
  - **inMemoryAutoRelationshipTickCount**

Frequency for Automatic Relationship policies evaluation. This is a new, faster evaluator, which requires SQL Server 2008 or newer.

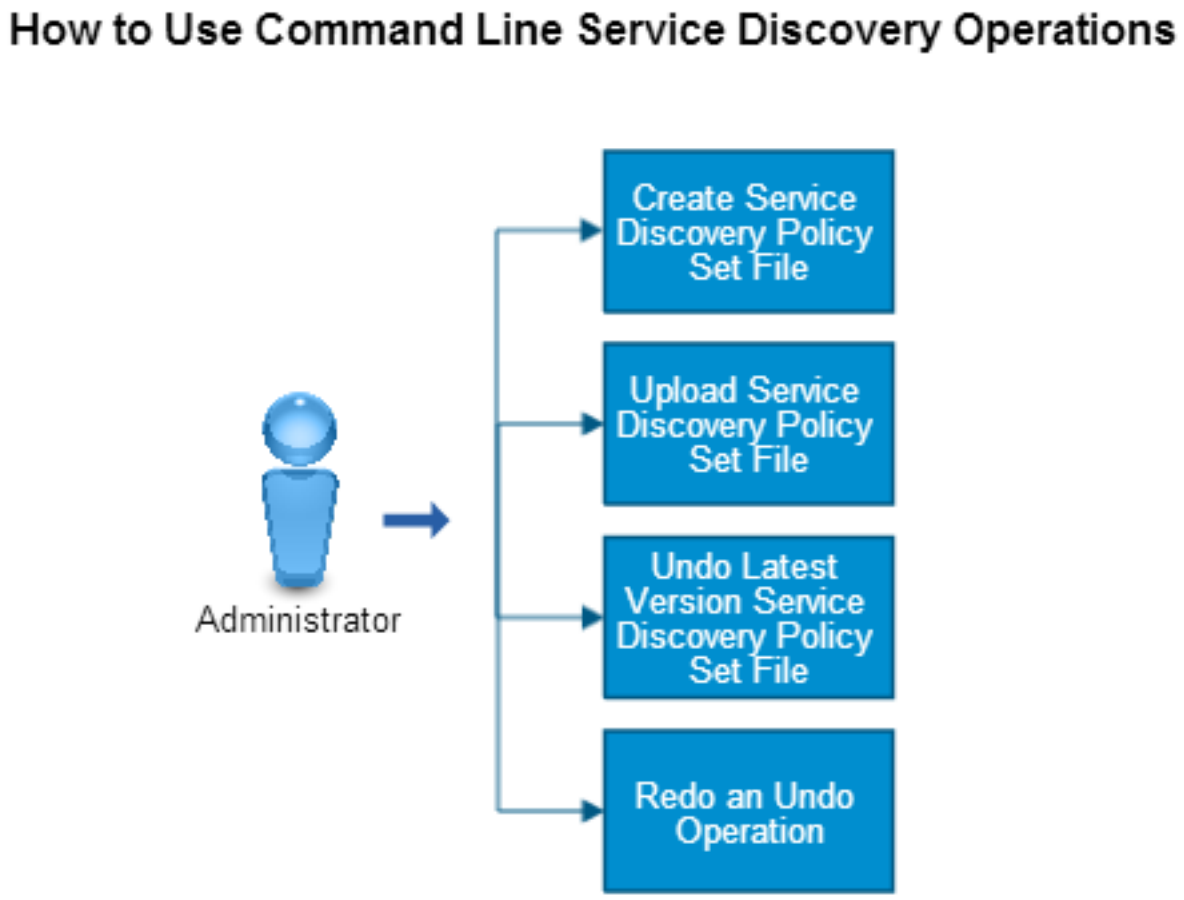
- **reevaluationTickCount**
  Frequency for reevaluating existing relationships created by Service Discovery. Deletes already created service discovery relationships which do not match any of the policy.
- **deleteUnusedSdSheetsTickCount**
  Frequency for looking for service discovery projections in the notebooks which are not used by any service discovery relationship.
- **autoRelationshipReevalTickCount**
  Frequency for Automatic Relationship policies evaluation when a timestamp is not used. It is actually a reset of the evaluation of the automatic relationships.

How to Use Command Line Service Discovery Operations

As an administrator, you can use the command line Service Discovery operations to create and upload service discovery files as well as undo and redo operations. These operations are useful for backing up and restoring service policies.

Use this scenario to guide you through the process:

Figure 34: how to use command line service discovery operations



- Create a Service Discovery Policy Set File.
- Upload a Service Discovery Policy Set File.
- Undo the Latest Version of a Service Discovery Policy Set.
- Redo an Undo Operation.

All .bat files are located in the <SOI_HOME>\ServiceDiscovery directory on the system where Service Discovery is installed.

Create a Service Discovery Policy Set File

The read_rules_from_db.bat file lets you create an .xml file that contains the current service policy set definitions from the database. You can use this file as a backup to restore at a later time.

> **NOTE**
> Do not edit the .xml file manually. Use the Service Discovery Policy Editor to change your policies.

Follow these steps:

1. Open a command prompt on the SA Manager system.
2. Navigate to <SOI_HOME>\ServiceDiscovery, and run the following command:

   ```
   read_rules_from_db.bat <filename>.xml
   ```

   – **filename**
   Specifies the output .xml file you want to create. If a file by the same name already exists, you are prompted to use a different filename and you must run the .bat file again with the new filename. You can use a relative or absolute path.

   Debug information displays as read_rules_from_db.bat creates the file.
   File creation completes and the .xml file appears either in the directory you specified or the ServiceDiscovery directory if you did not specify a path.

Upload a Service Discovery Policy Set File

The upload_rules_to_db.bat file lets you upload service policy definitions back into the database from an .xml file you created with read_rules_from_db.bat.

When you upload the .xml file, upload_rules_to_db.bat validates the following:

- Class names in policies are valid USM class names
- Properties in the criteria are valid properties of the class
- Operators (StartWith, Contains, Equals, and so on) are valid operator names
- All required properties, such as relationship types, are provided

Uploading a Service Discovery policy set increases the Revision number by one.

> **NOTE**
> Do not edit the .xml file manually. Use the Service Discovery Policy Editor to change your policies.

Follow these steps:

1. Open a command prompt on the SA Manager system.
2. Navigate to <SOI_HOME>\ServiceDiscovery, and run the following command:

   ```
   upload_rules_to_db.bat <filename>.xml
   ```

   – **filename**
   Specifies the input .xml file created with read_rules_from_db.bat that you want to restore. The upload_rules_to_db.bat file does several semantic checks to avoid loading an invalid file.

   Debug information displays as upload_rules_to_db.bat uploads the .xml file to the database.
   The upload completes and you are returned to the command line.

3. Close and reopen the Service Discovery Policy Editor so that the policies are refreshed in the Policies tab.

Undo the Latest Version of a Service Discovery Policy Set

The undo_rules.bat file lets you undo the last save of a policy set, whether the save was performed through the Service Discovery Policy Editor or as the result of uploading service policy definitions using upload_rules_to_db.bat. You can undo up to ten saves. Service discovery remembers 10 latest save operations, so the undo can be done 10 times.

Performing an undo decreases the Revision number by one.

Follow these steps:

1. Open a command prompt on the SA Manager system.
2. Navigate to <SOI_HOME>\ServiceDiscovery, and run the following command:
   ```
   undo_rules.bat
   ```
   Debug information displays as undo_rules.bat performs the undo.
   If the .bat file detects that there is no older version to undo, the operation ends.
   The undo completes and you are returned to the command line.

Redo an Undo Operation

The redo_rules.bat file lets you redo an undo you performed using the undo_rules.bat file.

You can perform as many redo operations as undo operations that have been performed. Therefore, if you performed five undo operations, you can perform up to five redo operations. If you attempt to perform more redo operations redo_rules.bat notifies you that the current policy set is the latest.

Performing a redo increases the Revision number by one.

Follow these steps:

1. Open a command prompt on the SA Manager system.
2. Navigate to <SOI_HOME>\ServiceDiscovery, and run the following command:
   ```
   redo_rules.bat
   ```
   Debug information displays as redo_rules.bat performs the redo.
   If the .bat file detects that the current policy set is the latest, you receive a message and the operation stops.
   The redo completes and you are returned to the command line.

How to Create Generic Service Relationship Policy

Contents

As an administrator, you use the Service Discovery Policy Editor to create and manage policies that automatically create relationships based on policy criteria.

Generic service relationship policies are based on the following:

- **Relationship**
  Defines the type of relationship to create.
- **Source CI**
  Defines the CI type to use as the source of the relationship with optional attribute-based criteria. Only Service CI is available as a source.
- **Target CI**
  Defines the CI type to use as the target of the relationship with optional attribute-based criteria.
- **Match Criteria**

Defines how the source and target CIs must relate (based on attribute values) for the policy criteria to match and create a relationship between the CIs.

- **Scope**
  Defines under which services the relationships are created. Scope applies to all services matching the criteria.

The automatic relationship policy creates a new relationship in the service that automatically adds the target CI to the service. Only the source CI needs to pre-exist in the service.

A CI can become part of the service through various ways; for example:

- Imported service already includes the CI in it.
- CI is manually added to the service.
- Another Service Discovery policy (for example, a dynamic service policy) adds the CI to services.

> **NOTE**
> When a policy specifies a class name (for example, class=ComputerSystem), the policy applies to the CIs of that class and also to its subclasses (as defined by the USM Schema Type hierarchy). For example, in the case of ComputerSystem, the policy would also apply to VirtualSystems.

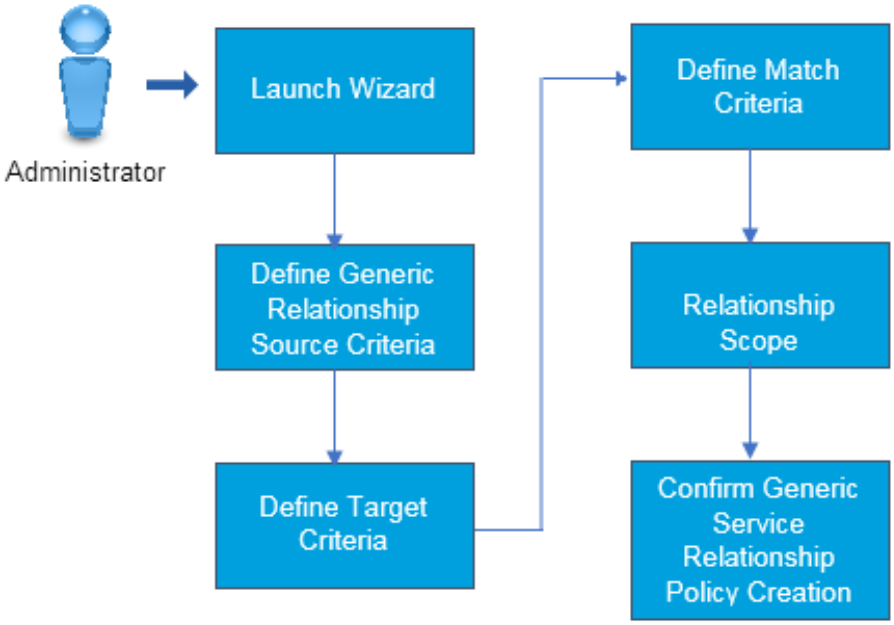
When the specified source CI (Service in this case) exists and the policy criteria are met, Service Discovery adds a relationship to all the specified target CIs in the service. Scope of the policy applies to all existing services.

For example, you can create a generic service relationship policy that creates relationships between Service and all ComputerSystem based on matching label of Service and ComputerSystem's.

You can create one policy at a time for a relationship. Multiple policies for each relationship are supported, but you must create them in separate operations.

> **NOTE**
> Service Discovery policies support only USM-Core CIs.

Use this scenario to guide you through the process:



1. Launch the Wizard.
2. Define the Relationship and Source CI Criteria.

3. Define the Target CI Criteria.
4. Define the Match Criteria.
5. Define the Relationship Scope.
6. Confirm the Automatic Relationship Policies Creation.

You can also read a scenario that provides an example for creating automatic relationship policies.

Launch the Wizard

You use the Service Discovery Policy Editor to create and manage automatic relationship policies.

Follow these steps:

1. Open the Operations Console.
2. Select Tools, Service Discovery Policies.
   The Service Discovery Policy Editor opens.

   > **NOTE**
   > System-wide, only one user can access the Service Discovery Policy Editor at a time.

   The Policies tab provides a tree that displays the Dynamic Services, Automatic Relationships, Generic Service Relationships, and Unmanaged Relationships available.
3. Right-click Generic Service Relationships in the Policies tab and select Create.
   The wizard opens on the Source Criteria page.

Define the Relationship and Source CI Criteria

The Source Criteria page lets you define the relationship type and the source CI criteria.

Follow these steps:

> **NOTE**
> Page fields marked with an asterisk (*) are required.

1. Select a USM relationship type from the Relationship Type drop-down list.
   Once you select a relationship type, the Relationship Type field becomes read-only in subsequent pages and the relationship expression builds as you select the target and match expression.
2. Select the class (USM type) of the source CI from the Class drop-down list (only services) in the Source pane.
   Consider the following:
   – The Attribute drop-down list displays a subset of the USM properties available for the selected type.
   – For USM property definitions, see the USM Schema Documentation.
3. (Optional) Select a USM attribute, comparison type, and attribute value for the expression. If you want to add all CIs of the selected type without adding attribute criteria, skip to Step 6.

   > **NOTE**
   > For attribute values, mouseover the field and a tooltip displays the data type required.
4. Click Add.
   The criteria are added to the expression, which appears in the logic tree and as an expression in the field at the bottom of the page.
5. (Optional) Repeat Steps 2 - 4 to add additional criteria and logic.
   Note: For more information about using advanced logic with multiple source criteria, click the Hints link.
6. Click Next.

Define the Target CI Criteria

The Target Criteria page lets you define criteria for the target CI in the relationship.

Follow these steps:

1. Select the USM type of the target CI from the Class drop-down list in the Target pane.
   Consider the following:
   – The Attribute drop-down list displays a subset of the USM properties available for the selected type. Also, certain attributes such as MdrProduct attributes and date/time data types are not supported and do not appear in the drop-down.
   – When you select a USM type, all available derived types in the USM hierarchy are also included. For example, when you select the ComputerSystem type, its child type VirtualSystem is automatically included in the policy.
   – For USM property definitions, see the USM Schema Documentation.
2. (Optional) Select a USM attribute, comparison type, and attribute value for the expression. If you want to add all CIs of the selected type without adding attribute criteria, skip to Step 5.

   NOTE

   For attribute values, mouseover the field and a tooltip displays the data type required.
3. Click Add.
   The criteria are added to the expression, which appears in the logic tree and as an expression in the field at the bottom of the page. CIs of the specific type must meet the attribute-based criteria to match.
4. (Optional) Repeat Steps 2 - 3 to add additional criteria and logic. CIs of the specified type must meet the attribute-based criteria to match.

   NOTE
   For more information about using advanced logic with multiple target criteria, click the Hints link.
5. Click Next.

Define the Match Criteria

The Match Criteria page lets you define a comparison between the attributes of your source and target.

Follow these steps:

NOTE
Page fields marked with an asterisk (*) are required.

1. Select a source attribute, comparison type, and target attribute from the drop-down lists as follows:
   – *'Source_CI'* Attribute
   Defines the USM attribute of the service that must have some relation to a target CI attribute
   – **Comparison Type**
   Defines how the specified source and target CI attributes must relate. The options in this drop-down list change based on the data type of the selected attributes.
   – *'Target_CI'* Attribute
   Defines the USM attribute of the target CI that must have some relation to a source CI attribute.

   NOTE
   For USM property definitions, see the USM Schema Documentation.
2. Click Add.
   The criteria are added to the expression, which appears in the logic tree and as an expression in the field at the bottom of the page.

   NOTE
   If you define criteria that conflict with the potential values of the selected attributes (for example, if you select Equal To for two enumerated properties that do not share values), an error message opens.
3. (Optional) Repeat Steps 1 - 3 to add additional conditions and logic.

NOTE
For more information about using advanced logic with multiple match criteria, click the Hints link.

4. Click Next.

Define the Relationship Scope

The Relationship Scope page lets you define the scope of the relationship, which defines which services are affected when creating the relationships.

Follow these steps:

1. By default scope is selected
   – **All services**
     Creates the relationship in all services with matching criteria. This is the only option that is available and is selected by default.
2. Click Next.
   The Confirm page opens. A Topology Warning dialog may appear if the Service Discovery editor detects that the policy might cause topology errors or unmanageable services. Evaluate the policy before proceeding to verify whether changes are required. For more information, see Topology Warnings.

Confirm the Automatic Relationship Policies Creation

The Confirm page displays the policy expression for the automatic relationship criteria you created.

Follow these steps:

1. Click Finish to confirm creation of the new automatic relationship policy.
   The new policy appears under the Automatic Relationships folder. Expand the semantic type to view the relationship policy. You can easily define additional policies for the semantic by right-clicking the semantic and selecting Add.

   NOTE
   If you receive a topology warning, check whether changes are required. For more information, see Topology Warnings.
2. Click OK or Save when the automatic relationship policies are complete.
   The Service Discovery engine begins scanning the Persistent Store for CIs that match the policy criteria.

Generic Service Relationship policy

How to Create and Work with Service-Level Agreements

Contents

As an administrator, you can define, view, edit, attach or detach an SLA with service models.

A *service-level agreement* (SLA) is a contract that specifies the service expectations of internal or external customers. An example is the downtime that is acceptable for various resources.

You can monitor service performance based on specified thresholds and receive a notification when the SLA approaches or breaches the thresholds.

SLAs let you track service metrics over a specified interval based on specific thresholds. They can help ensure adherence to any quality or availability requirements that an organization must maintain.

SLAs in CA SOI are based on the following attributes:

- Health, Quality, Risk, or Availability metric
- Violation threshold
- SLA time period
- Business hours

Each SLA is based on one of the core CA SOI metrics. An SLA calculates violations that are based on a threshold, spans a defined time period, and is optionally only monitored during defined business hours.

CA SOI uses the measurable provisions that you specify in the SLA to monitor the real-time health of each associated service, and records outage time when the service is down. The recorded time is compared to the SLA thresholds to determine the status of the SLA for a given time period.

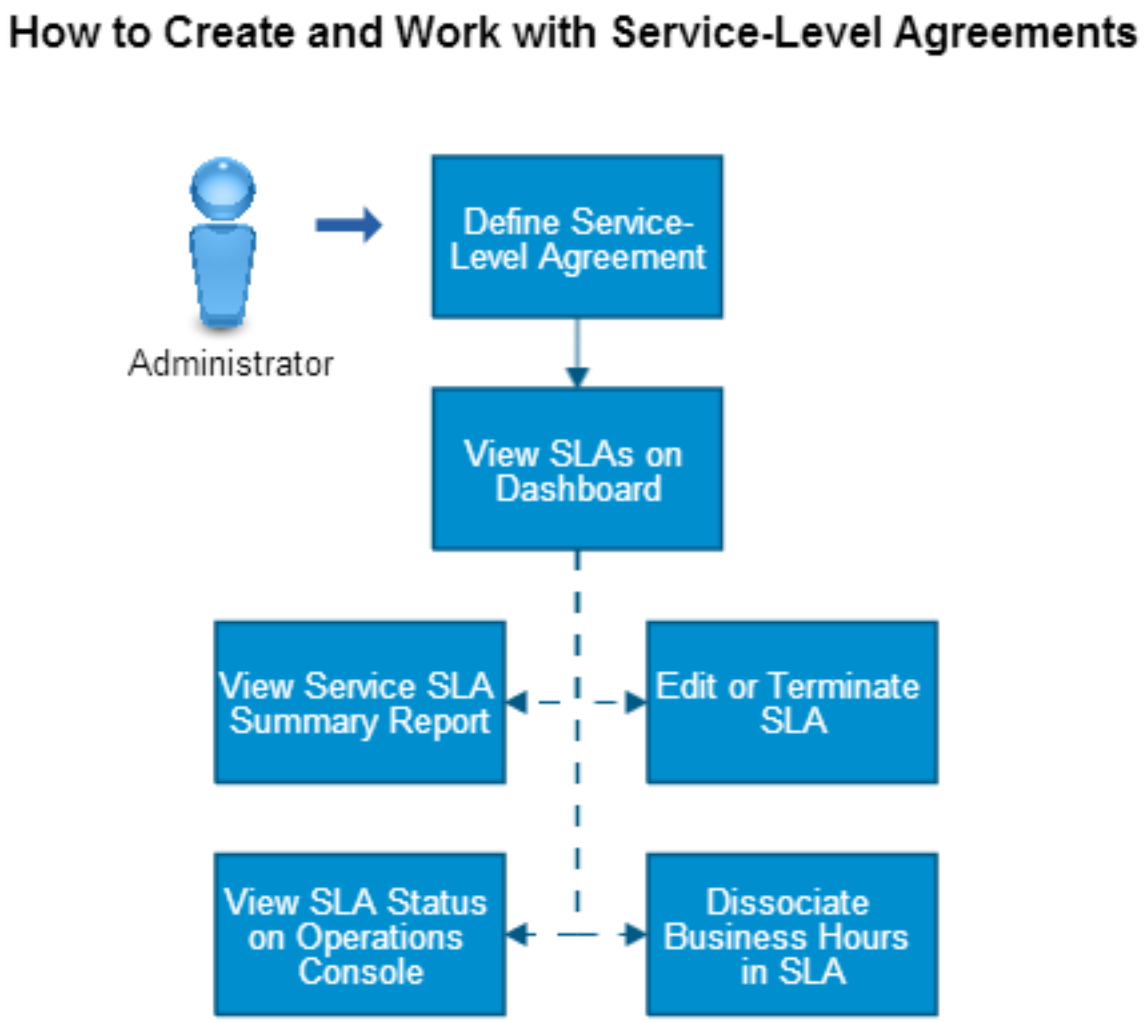
For example, you can define an SLA that tracks the quality of a customer-facing website. If the quality drops below the threshold level, an outage is recorded. If the quality stays below the threshold level past the defined threshold time, a violation is recorded. You can track the SLA status through the Operations Console, Dashboard, and reports to ensure that the website meets the customer quality expectation level.

For an in-depth example, see SLA Example.

When you create an SLA for a service, the SLA evaluation begins after the SLA period start date and when the defined business hours begin, if applicable. The SLA tracks outages and violations; an outage occurs when the service crosses the threshold defined in the SLA. A violation occurs when the service crosses the threshold for the amount of time defined in the SLA.

Use this scenario to guide you through the process:

Figure 35: how to work with service-level agreements



1. Define the SLA.
2. View the SLAs on the Dashboard.
3. (Optional) View a Service SLA Summary report.
4. (Optional) View SLA Status on the Operations Console.
5. (Optional) Edit or Terminate an SLA.
6. (Optional) Disassociate the business hours in an SLA.

Define a Service-Level Agreement

You define a service-level agreement when creating or editing a service. Each SLA is specific to a service, and a service can only have one SLA.

Follow these steps:

1. Open the Operations Console, and complete one of the following actions:
   – Select Tools, Create New Service.
   – Right-click a service in the Navigation pane and select Edit Service.
   The Service Modeler opens.
2. Click the SLA tab.
3. Enter a name in the Service Level Objective (SLO) Name field, an optional description in the Description field, and select the Enabled check box to immediately enable the SLA after you save it.
4. Perform the following steps:
   a. Indicate whether to base the SLA on Health, Quality, Risk, or Availability.
   b. Select the threshold level. For example, for Health the threshold levels are Minor, Major, Critical, Down, and Unknown. If you select Critical, any period of time when health becomes Critical, Down, or Unknown, it is considered a service outage and is one factor used to determine whether the SLA is violated.
   c. Select Percent or Seconds and enter a value to determine how long the threshold level must be breached to trigger a violation.

      NOTE
      The Seconds value must be greater than 0 and less than 1000000.
5. Click Select for the SLO Period.
6. Perform one of the following steps:
   – Click Create to create a new schedule for the SLA time period. Continue with Step 7.
   – Select an existing schedule and click OK. Continue with Step 9.
   A description of the selected period appears in the SLO Period pane.
7. (New SLA period only) Complete the Create Period dialog and click OK.
   – **Start Date**
   Specifies the date on which to begin calculating the service SLA status. Depending on your selections in the Recurrence pane, the period may not start on the exact start date. For example, if you define a start date of September 15th with a monthly recurrence that starts on the 17th of every month, the period does not start until September 17th.
   – **Recurrence**
   Specifies how long the SLA period lasts, and when it expires. Select from the following options:
   - **Daily**
     Specifies that the period recurs after a specified number of days. For example, you can create a period that renews every three days. A daily period starts for the first time on the specified start date.
   - **Weekly**
     Specifies that the period recurs after a specified number of weeks and begins on a specific day. For example, you can create a period that starts on the first Monday after the specified start date and renews every two weeks.
   - **Monthly**
     Specifies that the period recurs after a specified number of months and begins on a specific day of the month. For example, you can create a period that starts on the fifth day of the month after the specified start date and renews every three months.
   - **Yearly**
     Specifies that the period recurs *after* a specified number of years and begins on a specific month and day. For example, you can create a period that starts on January 5 (the first occurrence of this day *after* the start date) and renews every two years.
     Note: If the Start Date and the Recurrence date are the same, the recurrence begins the *following* year. If you want a yearly SLO Period recurrence to begin this year, then set a Recurrence date at least one day after the Start Date. For example, if the Start Date is Jan 1, 2013 and the Recurrence date is Jan 1, the first recurrence is Jan 1, 2014. However, if you set the Recurrence Date to Jan 2, the first recurrence happens on Jan 2, 2013.

NOTE
For all recurrence options, select either No Expiration for the period to recur indefinitely or enter a Schedule End Date when SLA evaluation stops.

8. (New SLA period only) Select the created SLA period and click OK.
   A description of the selected period appears in the SLO Period pane.
9. (Optional) Select the Other option in the Business Hours pane to specify business hours to restrict SLA evaluation, and perform one of the following actions:
   – Click Create. Continue with Step 10.
   – Select an available schedule, if any, and click

.

   Continue with Step 11.
10. (New business hours only) Enter the following information in the Create Business Hours dialog and click OK:
   – **Start Date**
     Specifies when the business hours schedule starts. Depending on your selections in the Recurrence pane, the period may not start on the exact start date. For example, if you define a start date of September 15th with a monthly recurrence that starts every week on Monday, the period does not start until the first Monday after September 15th.
   – **Time**
     Specifies the daily business hours time interval. These values must be rounded to 30 minute intervals.
   – **Recurrence**
     Specifies how long the business hours period lasts, and when it expires. Select from the following options:
     - **Daily**
       Specifies that the period spans every weekday, every weekend day, or every day of the week. For example, you can create a business hours schedule from 9:00-5:00 on Monday through Friday.
     - **Weekly**
       Specifies that the period starts every week on a specific day.

     For all recurrence options, you must either select No Expiration for the period to recur indefinitely or enter a Schedule End Date when the business hours period stops.

     NOTE
     You do not have to synchronize the SLA period and business hours recurrence types or recurrence days. For example, you can use a monthly SLA period with weekly business hours. If you define an SLA period that starts on Monday and a business hours schedule that starts on Wednesday, the SLA evaluation begins when the business hours begin.

   The created schedule appears in the Current Schedules pane.

   NOTE
   If you enter multiple business hours, the product verifies that the hours do not overlap.
11. Click Save, OK.
   The service-level agreement is defined.

View SLAs on the Dashboard

The dashboard contains information that you can use to monitor current SLA status. Monitor SLAs from the dashboard as follows:

- **Current SLA tab**
  Displays the current state of the SLA for each service. Each SLA has one of the following values in this column:
  – Compliant

Indicates that the SLA is compliant for the current SLA period.

– Violated

Indicates that the SLA threshold is violated for the current SLA period.

– Inactive

Indicates that the SLA is inactive, due to the current time being outside of the SLA period or business hours.

If there is no icon in this column, the service does not have a defined SLA.

- **SLA tab**
  Displays the following SLA information in the Details of Selected Service pane for the selected service in the Services pane:
  – **SLA Current Status**
  Displays a pie chart of the SLA status during the current SLA period. This view also displays the following information:
    - The SLA type, threshold, and description
    - The amount of time in a violated state, if applicable
    - The time that the chart was last updated

    Click the pie chart to view further details about the current SLA period in the Service SLA Summary report.
  – **SLA History**
  Displays a bar chart or line chart view of the SLA downtime over the last ten SLA periods. When you scroll over a bar or line, details appear about the nature of the downtime, which can be unplanned (outages and violations), planned maintenance, or unknown.

    NOTE
    The bar chart can show a single SLA period, but the line chart requires at least two SLA data points to draw the line.

    Click a bar or line to view further details about the represented SLA period in the Service SLA Summary report. The SLA History charts do not display the 'Click to execute the report' tooltip shown in the SLA Current Status view when you scroll over the bar or line. The SLA History part instead shows further SLA period details, but you can still generate a report by clicking the bar or line.

  Both views present SLA states as the following color-coded values:
  – **Up**: Green
  – **Unplanned**: Red

    NOTE
    Unplanned maintenance is the total of outage and violation time.

  – **Maintenance**: Brown
  – **Unknown**: Gray

    NOTE
    The SLA charts display the date in yyyy/mm/dd format. For example, 2010/05/08 is May 8th, 2010. The reports generated from the SLA charts use mm/dd/yyyy format (for example, 05/08/2010.

View Service SLA Summary Report

The Service SLA Summary report compiles the SLA data into a series of tables and charts for comprehensive SLA monitoring.

To run the Service SLA Summary report, perform one of the following actions:

- Click the charts in the SLA tab of the Dashboard. This option automatically runs the report for the specific SLA and the SLA time period represented in the chart.
- Click the Reports link and select the Service SLA Summary report. When you run the report from the reporting interface, you must enter settings that define the SLAs to include and the report time period.

  NOTE
  For more information about this report and reporting, see the Run Reports from the Dashboard section.

View SLAs Status on the Operations Console Data

Monitor the SLA status from the following places in the Operations Console:

- **Services tab in Contents pane**
  Displays a list of all services when you select the top-level Services item in the Services tab. The SLA Status column on this tab displays the current status of each service's SLA.
- **Information tab in Component Detail pane**
  Displays the following SLA information for the selected service in the Service Level Agreements table:
  - Name
  - Description
  - Current state and last known state
  - Last status update
  - First violation time

Edit or Terminate an SLA

You can change the name, description, or enabled state of an existing SLA.

You must terminate the existing SLA and create a new one if you need to change any other properties, such as threshold or SLA period. Once SLA evaluation starts, changes to the SLA period, threshold, or business hours may cause an incorrect determination of SLA status.

Follow these steps:

1. In the Service Modeler, click the SLA tab.
2. Perform one of the following actions:
   - Edit the SLA name or description, or clear the Enabled check box to temporarily stop SLA evaluation, and click Save, OK.
   - Click Terminate SLA and click Yes to confirm.

Disassociate Business Hours in a Service-Level Agreement

Before deleting Business Hours schedules, you must disassociate the related SLA.

Follow these steps:

1. In the Service Modeler, click the SLA tab.
2. Select the Other option in the Business Hours pane.
3. Move any schedules from the Current Schedules pane to the Available Schedules pane and click Save, OK.
   The SLA is saved and the hours are removed.

SLA Example

Consider a retail website that must be available at all times for customers to place orders. The business enforces concrete performance and availability requirements on the website. Not only must the website be available for a certain amount of time, it must perform above a certain threshold so that customers are not frustrated by a slow response time.

The administrator must ensure that the website response time does not dip below a defined threshold for more than 8 hours in a month.

Assume the following for this example:

- The administrator modeled a service for the website that includes the components that enable the website to perform as expected. The service includes the web servers, order processing applications, product databases, network hosts and routers, and so on. The service also accurately represents the relationships among the CIs and the significance of each component.
- The benchmark to which the website must adhere is response time. The domain manager that manages the service CIs can calculate the response time and translate the metric to a severity that you can associate with service quality in CA SOI.

To create an SLA that monitors the website response time against defined benchmarks, the administrator enters the SLA properties for the website service similar to the following:

- Base the threshold on Quality in the SLA tab that must equal or exceed a moderately degraded status for 28800 Seconds of the SLA time period.
  This threshold assumes that a moderately degraded service quality indicates that the website response time has crossed the response time threshold. It also assumes that any status more severe than moderately degraded also indicates a response time threshold breach.
- Create an SLA period that recurs monthly and starts on the first day of every month.
- Select 24x7 for Business Hours, because the web site must be available for customer use at all times.

After the SLA period begins, the administrator can monitor SLA status to ensure adherence to the response time benchmark. For example, you could schedule a Service SLA Summary report to run every day at a specific time to summarize the downtime, outages, and SLA status for that day.

How to Create and Manage Customers

As an administrator, you define customers and associate them with service models to see the impact of service degradation on the customer.

A *customer* in CA SOI is any consumer of a managed service. A customer can represent a specific division of the company such as a product division, region, or city. Customer impact provides IT personnel with an accurate understanding of what fault conditions really mean to a particular customer. For example, the administrator defines a customer to include all services within South America. If a fault occurs on a CI, such as a router, the IT personnel can easily determine how South America is impacted.

Alerts can indicate both service impact and customer impact. To determine the customer impact, an administrator creates a customer and assigns services to that customer.

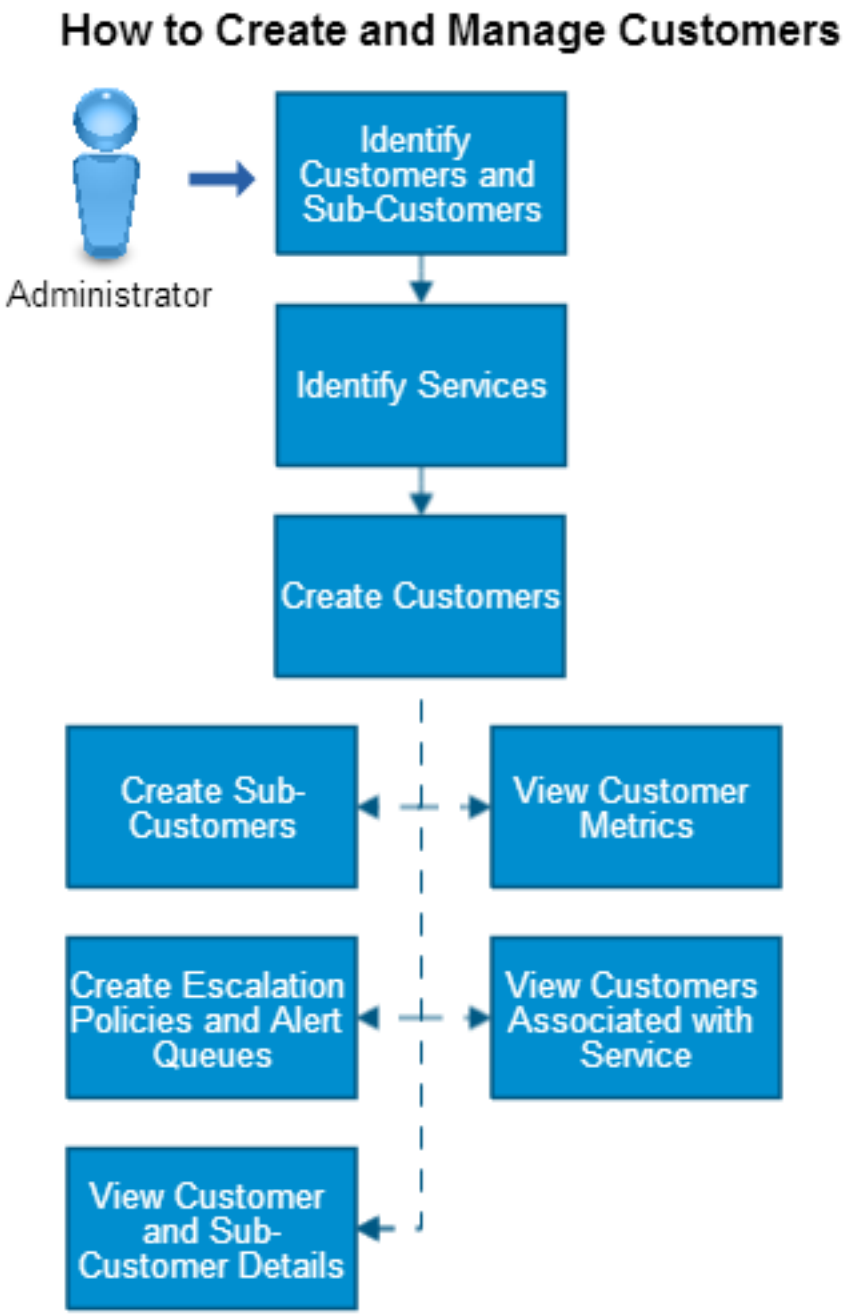
Sub-customers represent smaller divisions of a larger entity customer. For example, you can create a customer that represents a geographical division and sub-customers for smaller regions. You could continue to create additional sub-customers for cities, office buildings, office floors, and so on. You could also create a customer that is your cloud network and the sub-customers are the actual components utilized by those paying for your service. The number of sub-customers and nesting levels depends on how you want to monitor the customer impact.

Administrators can place a priority on a customer, which weighs that customer as higher or lower than other customers.

Administrators can use customer conditions as criteria to define escalation policy and alert queue rules. For example, you can create an escalation policy that sends an email to a specific IT person when the customer impact is Severe. An administrator can also use runtime tokens that display the customer name and severity in escalation actions. For more information about using runtime tokens, see Expandable Runtime Tokens.

Use this scenario to guide you through the process:

Figure 36: how to create customers



1. Identify?customers and?sub-customers
2. Identify?services
3. Create?customers
4. (Optional) Create?sub-customers
5. (Optional) Perform any of the following tasks:

- Configure?customer priority and labels.
- Create?escalation policy and alert queues using customer-related conditions.
- View?customer and the sub-customer information.
- View?customer metrics.
- View?customers that are associated with a particular service.

An example shows you a real-world scenario for creating the customers, sub-customers, priorities, assigning services, and creating escalation policies and alert queues utilizing customers.

Identify Customers, Sub-Customers, and Priorities

First, you identify your customers. Is your customer a region, a company department, an actual buyer of your cloud services, or another entity?

Sub-customers are customers within a larger customer entity. For example, say that your company has divisions in North and South America and each division has the following departments: Accounting, Finance, and Human Resources.

You can identify the following customers (divisions) and sub-customers (departments). You can then monitor the customer impact of any managed services on any division as a whole or on an individual department:

North America

- Accounting
- Finance
- Human Resources

South America

- Accounting
- Finance
- Human Resources

For each customer and sub-customer, you can optionally apply a customer priority.

Although the alert severity determines the customer impact, you can use both customer priority and customer impact to determine escalation policy and alert queues. For escalation policies, if an alert impacts multiple customers, the escalation policies on the higher priority customer are applied first. For alert queues, you can use the customer priority and the customer impact when building rules. For example, you can define an alert queue where alerts of any severity are included if the customer priority is a certain level. Similarly, you can define an alert queue where alerts are included if the customer impact is above a certain threshold, such as Moderate.

Consider an administrator that creates customers for its Infrastructure as a Service (IaaS) services. Each customer represents a virtual machine or network in a cloud that are sold to clients. The administrator determines that a particular virtual machine is the highest importance due to the service level agreement. The administrator creates a customer that is comprised of that virtual machine's services and sets the priority to 10 (the highest priority). The administrator can then configure the escalation policy to trigger first on the highest priority customer. Operators then resolve the highest priority virtual machine issues quickly and they maintain the service level agreement.

Identify Services

Once you have identified your customers and sub-customers, you can easily determine services that are associated with each customer. When you create your customers and sub-customers in CA SOI, you create an association with the services. Therefore, you must have a service model available from which you can identify the services.

Create Customers

Create customers using a wizard.

Follow these steps:

1. Open the Operations Console and click the Customers tab.
2. Click the Create a New Customer icon.
3. Enter a Customer Name, Customer Identity, and optionally a Description. These values appear on the Information tab in the Contents pane.

   NOTE
   The customer identity (customer ID) uniquely identifies a customer. You cannot have duplicate customer IDs, but you can have duplicate customer names.
4. (Optional) Select a customer priority from the Priority drop-down list. You can select a value from 1 through 10; 1 represents the lowest priority and 10 represents the highest. If an alert impacts multiple customers, the escalation policies on the higher priority customer are applied first. You can also use the customer priority and customer impact to determine alert queues.

   NOTE
   Click the configure priorities

   icon next to the Priority drop-down list to configure the customer priority. You can assign a meaningful label to each priority level; for example, Gold to represent the highest priority. If you do not want to use a specific priority level, you can disable it.
5. Click Next.
   You use this screen to assign the services to your new customer.

   Consider the following items:

   - When you assign a service to a customer, CA SOI automatically assigns the subservices also.
   - An asterisk (*) indicates sub-services that are automatically assigned with the selected parent service.
   - Assigning a parent service automatically includes all its sub-services. Similarly, removing a parent service automatically removes its sub-services, unless a sub-service is a child to another parent in the Assigned Services list. The sub-services are prefixed with * and you cannot remove the sub-services unless their parent(s) are also removed. Select the Show top level parent services only check box to hide all sub-services and to show only the highest level parent services in the Available Services list.
6. Use the arrows to add or remove services from the Available Services and Assigned Services lists. You can enter a string to filter either list.
7. Click Next.
   This screen lets you assign the user groups that have access to the new customer. For more information about how service and role-based security affect customers, see How to Configure Role-Based Security.
   Consider the following items:
   – The user group must also have access privileges to the services you assigned to the new customer.
   – You can also manage user group access to customers in the Users tab.
8. Use the arrows to add or remove user groups from the Available Groups and Allowed Groups lists. You can enter a string to filter either list.

   NOTE
   User groups marked with an asterisk have their access set to all customers either by default or by an administrator.
9. Click Next.
10. Verify the new customer information and click Finish.
    The new customer displays in the Customers tree.

Create Sub-Customers

Sub-customers represent smaller divisions of a larger entity customer. Create sub-customers under existing customers.

Follow these steps:

1. Open the Operations Console and click the Customers tab.
2. Select the customer or sub-customer for which you want to create a sub-customer.
3. Click the Create a Sub-Customer icon.
4. Enter a Customer Name, Customer Identity, Customer Priority, and optionally a Description, which appears on the Information tab in the Contents pane.

   NOTE
   The Customer identity (customer ID) uniquely identifies a sub-customer. You cannot have duplicate customer IDs.
5. Select a priority from the Priority drop-down list. Customer priority helps to determine the impact of an alert to a customer (customer impact). If an alert impacts multiple customers, the escalation policies on the higher priority customer are applied first.

   NOTE
   Use the configure priorities

   icon next to the Priority drop-down list to configure the customer priority. You can assign a meaningful label to each priority level; for example, Gold to represent the highest priority. If you do not want to use a specific priority level, you can disable it.
6. Click Next.
7. Use the arrows to add or remove services from the Available Services and Assigned Services lists. You can enter a string to filter either list.

   Consider the following items:

   - When you assign a service to a customer, CA SOI automatically assigns the subservices also.
   - An asterisk (*) indicates sub-services that are automatically assigned with the selected parent service.
   - Assigning a parent service automatically includes all its sub-services. Similarly, removing a parent service automatically removes its sub-services, unless a sub-service is a child to another parent in the Assigned Services list. The sub-services are prefixed with * and you cannot remove the sub-services unless their parent(s) are also removed. Select the Show top level parent services only check box to hide all sub-services and to show only the highest level parent services in the Available Services list.
8. Click Next.
   This screen lets you assign the user groups that have access to the new sub-customer.
   Consider the following items:
   – If a parent customer is given a permission for a user group, the child customer also gets the permission. For more information about configuring role-based security, see the How to Configure Role-Based Security section.
   – The user group must also have access privileges to the services you assigned to the new sub-customer.
   – You can also manage user group access to customers in the Users tab.
9. Use the arrows to add or remove user groups from the Available Groups and Allowed Groups lists. You can enter a string to filter either list.

   NOTE
   User groups marked with an asterisk have their access set to all customers either by default or by an administrator.
10. Click Next.
11. Verify the new sub-customer information and click Finish.
    The new sub-customer displays in the Customers tree.

Configure Customer Priorities and Labels

You can optionally apply a priority that can determine escalation policy or alert queues. If an alert impacts multiple customers, the escalation policies on the higher priority customer are applied first. You can configure the customer priority while creating or editing your customer.

You can also assign a meaningful label to each priority level to represent your unique environment. For example, you can use labels such as Gold, Silver, and Bronze to represent the particular customer tiers. You could label customers with the highest priority (say, a value of 10) as "Gold", medium priority (say, a value of 5) as "Silver" and so on.

Additionally, if you do not need certain priority levels, you can disable them. For example, per your organization policy, you set only four priority levels. In this case, you can disable the remaining six priority levels to avoid any confusion when creating alert queue rules or defining escalation policy.

Follow these steps:

1. Open the Define Customer screen in the customer wizard either by creating a customer or editing a customer.
2. Click the configure priorities

   icon next to the Priority drop-down list.
3. Enter appropriate names in the priority level fields (for example, Normal in Priority Level 1 and Moderate in Priority Level 2).
   The values that you enter in these fields are represented as options for the corresponding priority levels in the Priority drop-down list.
4. Select the required Disable option for the priority level that you do not want to use.
   The disabled priority levels do not display as options in the Priority drop-down list.
5. Click OK.
   The customer priority is configured.

Create Escalation Policies and Alert Queues

You can use the following customer attributes as criteria for escalation policy or alert queues:

- Customer ID
- Customer impact
- Customer name
- Customer priority
- Highest customer impact
- Highest customer priority
- Number of impacted customers

For example, you can create an alert queue with all alerts associated with customers assigned a Customer Priority higher than 8 and a Customer Impact that is Severe.

Similarly, you can create an escalation policy where a specific technician receives an email when the Customer Impact for his or her region (where the region is the customer) is Moderate or higher.

You can also use customer-related runtime tokens in escalation actions.

View Customer and Sub-Customer Details

You can view information about customers and sub-customers, including list of sub-customers, associated services, and associated service alerts.

Follow these steps:

1. Open the Operations Console and click the Customers tab in the Navigation pane.
   A list of all customers displays. The columns to the right of the customer name display the number of alerts of each severity for the customer and the total number of alerts for the customer. These alerts are the alerts that are impacting the services that are assigned to the customer.
   If a customer is associated with a service and an alert is generated that impacts that service, the alert also impacts the associated customer. An alert can impact multiple customers for the following reasons:
   – Because multiple customers are associated with a single service.
   – Because an alert impacts multiple services and each service has an associated customer.
2. View the customer tree icon color for any customer to see the overall customer health. Of all the services that are assigned to the customer, the service with the worst health represents the customer health. The customer tree icon color, therefore, shows the color that is based on that service's health.
3. Select a customer (or sub-customer) in the Customers tree and do any of the following tasks:
   – Click the Alerts tab to view all the alerts (from the services that are assigned to the selected customer) for the customer. This tab includes information about alert severity, category, summary, count of impacted customers, and so on.

     NOTE
     You can also display the number of impacted customers by adding the # Impacted Customers column.

     When you remove or add a service to a customer, the associated alert list is changed accordingly.

     NOTE
     The alert list for the parent customer shows the aggregate of all alerts from all of its child customers.
   – Click the Services tab in the Contents pane to view a list of all services that are associated with the customer. This tab includes information about the service name, health, risk, granularity, and so on.
   – Click the List tab in the Contents pane to view a list of sub-customers. This tab includes information about the all the sub-customers available under the selected parent customer. The tab displays the customer name, ID, priority, impact (or customer health), quality, risk, and description.

     NOTE
     Of all the services that are assigned to the customer, the service with the worst health represents the customer health. Similarly, the service with the worst quality represents the customer quality, and the service with the maximum risk represents the customer risk.
   – Click the Information tab in the Contents pane to view general information about customers. This tab includes general information about the customer: name, ID, and priority. This tab also includes current metric information: health, quality impact, risk, and priority. The Security pane shows the user groups that have access to the customer.
   – Click the Customer Impact tab to view the customer impact level. This tab is available to user groups with access privileges only.
4. Review the information. For the Services and Alerts tabs, you can use the additional tabs in the Component Detail pane to get detailed information.

View Customer Metrics

Customer metrics are similar to service metrics; however, customer metrics show the health, quality, and risk that are associated with a customer rather than a service. CA SOI provides the following customer metrics:

- **customer health**
  Identifies the worst state that is currently held by either customer quality or customer risk.
- **customer quality**

Identifies the worst customer quality for the services that are assigned to the customer. Quality indicates the level of excellence that consumers of an IT service experience. The highest propagated impact of an associated quality alert determines the customer quality value.

- **customer risk**
  Identifies the maximum risk on the customer. This metric indicates the likelihood of delivering the quality of service that is required to support the overall business objectives. The highest propagated impact of an associated risk alert determines the customer risk value.

On the Services tab:

1. Open the Operations Console and click the Services tab.
2. Select a service.
3. Click the Customers tab in the Contents pane.
   The Customers tab displays the customer name, identity, description, and customer metrics.

On the Customers tab:

1. Open the Operations Console and click the Customers tab.
2. Select a customer.
3. Click the Information tab in the Contents pane.
   The customer metrics display in the Information tab.

View Customers Associated with a Service

You can view all the customers that are associated with a specific service. This information helps you analyze the impact that a particular service has on different customers. When an administrator associates or removes a service from a customer, CA SOI updates the information in the Customers tab accordingly.

Follow these steps:

1. Open the Operations Console and click the Services tab.
2. Select a service for which you want to see the associated customers.
3. Click the Customers tab in the Contents pane.
   All customers that are associated with the selected service display in the pane. The pane also provides detailed information about the related customers; for example, name, ID, description, priority, and so on.

Example: Creating and Working with Customers

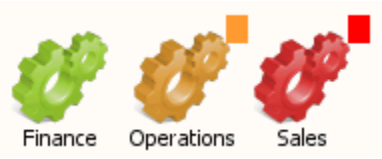
In this example, Forward Inc. is an MSP that provides finance, operations, and sales IaaS solutions for customers. Forward Inc. has already modeled several services to monitor their network. To ensure the highest level of service for their clients, Forward Inc. now wants to see how alerts on their modeled services impact specific clients. The clients are represented as customers in CA SOI. Forward Inc. Operators are assigned to monitor different services and customers and are provided service and customer access accordingly. The Admin Operator has access to all services and customers.

Forward Inc. provides three service levels: Bronze, Silver, and Gold. Clients who pay for Bronze service are provided the minimum service level and Gold provides the highest service level.

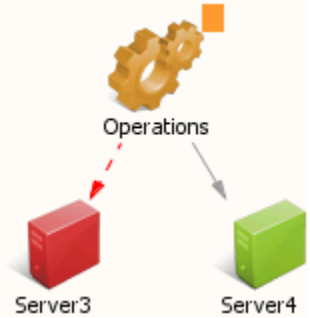
Identify Services

Services Available: Finance, Operations, Sales

The administrator models the three services that are based on the network for each IaaS offering as shown in the following graphic:



The administrator models the Finance and Operations services with two computer system CIs. Similarly, the administrator models the Sales service with three computer system CIs. The modeled services are shown in the following graphics:



Your company's service models are obviously more complex than the models provided in this example. You employ the techniquessuch as propagation policy, CI significance, and complex relationships. However, for our demonstrative purposes, we are using simple service models and not delving into the particulars of each service model. Our focus is on the service alerts and how they impact customers of those services.

Identify Customers and Priorities

Customers: Company-A, Company-B

Forward Inc. has two clients that use the IaaS solution services. The administrator plans to create two customers in CA SOI, Company-A and Company-B, to represent these two clients. Company-A has paid a premium for a higher service level, so the administrator will assign a higher customer priority to Company-A.

Company-A uses the Finance and Operations IaaS solutions and Company-B uses the Operations and Sales IaaS solutions, so the administrator will assign the services as follows when creating the customers:

Company-A: Finance and Operations services

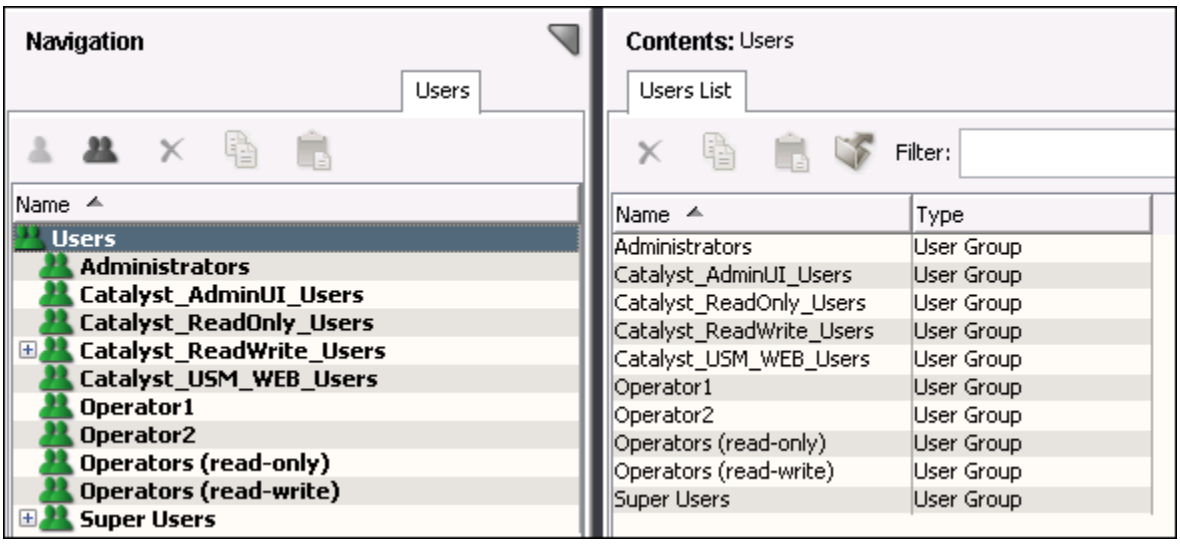
Company-B: Operations and Sales services

Company-A pays for the Gold service level and Company-B pays for the Bronze service level.

An administrator could elect to create sub-customers also. For example, Company-A and Company-B could have sub-customers that represent specific divisions of each client's company. If different divisions used different solution products (or a combination of solution products), an operator could see how alerts impact a specific division. However, for this example, we will use customers only.

Create User Groups

The administrator creates two "Operator" user groups, Operator1 and Operator2 in the Users tab. The users assigned to these user groups are operators that monitor customers. The administrator will assign the customers to Operator1 and Operator2 when using the wizard to create the customers.



The administrator would also assign one or more users to the Operator1 and Operator2 user groups. However, for simplicity in this example, we refer to the users as the Operator1 user and Operator2 user. Remember that these user groups include all users assigned to them.

The administrator also uses the Users tab to ensure that each user group has access to the services they monitor as described in the previous section. When an administrator grants user group access to a Operator user, the Operator user must close the console.

Create Customers, Assign Services, and Assign User Groups

The administrator creates the customers, Company-A and Company-B, and assigns the operator user groups as follows:

Operator1: Company-A customer

Operator2: Company-A customer, Company-B customer

Operator1 will monitor the service for the Company-A customer and Operator2 will monitor the services for both the Company-A and Company-B customers.

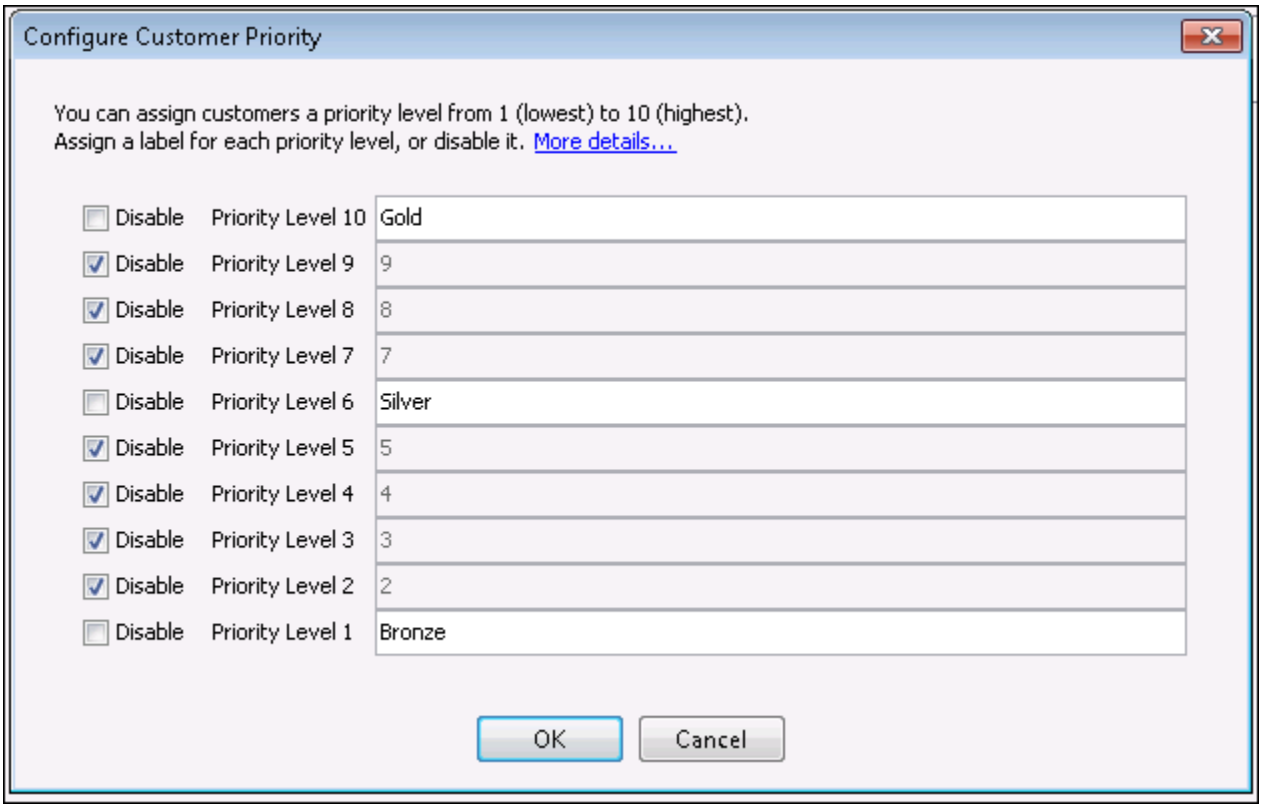
> **NOTE**
> Operator1 has permissions to the services (which are Finance and Operation services) of Company-A. Operator2 has permissions to the services (which are Finance, Operation, and Sales services) of Company-A and Company-B. Permissions to the services are given from the Services or Users tab. For more information, see the How to Configure Role-Based Security section.

The administrator launches the Create New Customer wizard and completes the Define Customer screen with the Company-A information as shown in the following graphic:

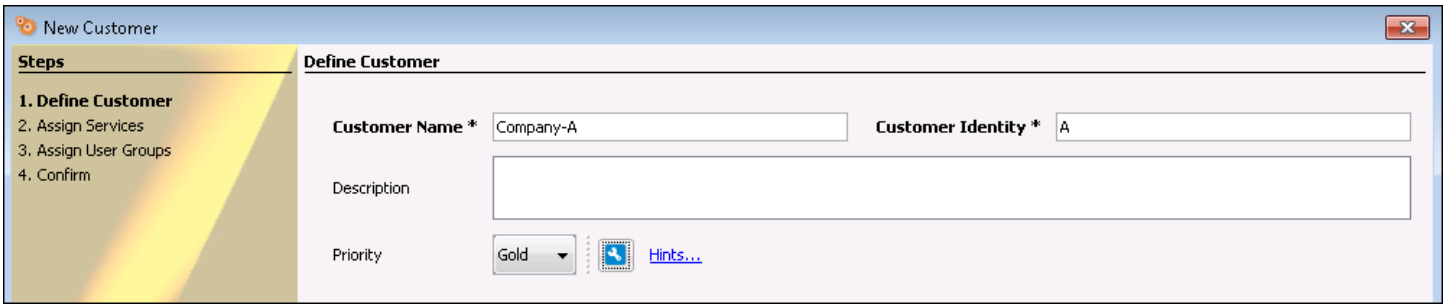
The customer Priority should reflect the Gold, Silver, and Bronze service levels. Therefore, the administrator will modify the Priority Levels to create more meaningful drop-down list labels. The administrator launches the Configure Customer Priority dialog and sets the following Priority Level labels:

- Priority Level 10 label to Gold
- Priority Level 6 label to Silver
- Priority Level 1 to Bronze

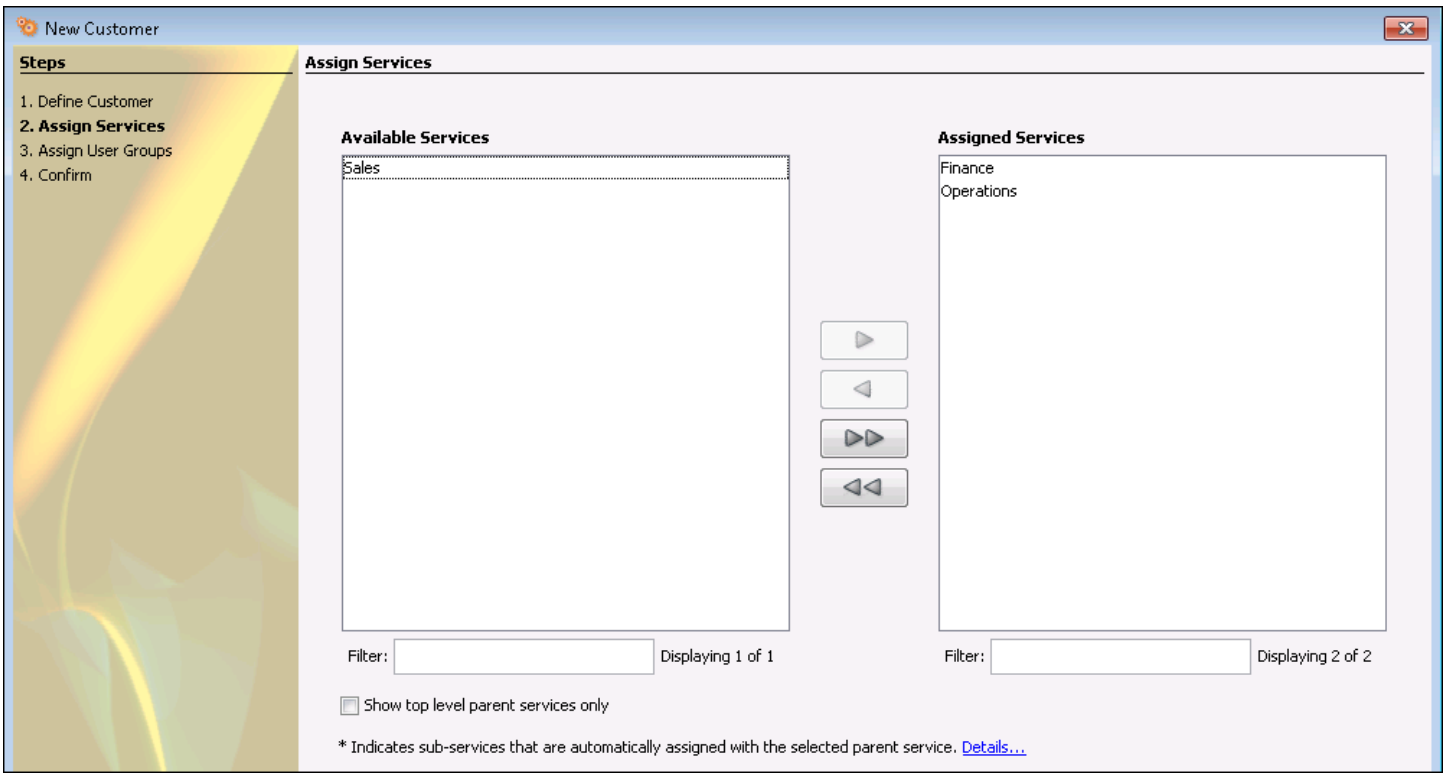
The administrator disables the unused priority levels. The completed dialog is shown in the following graphic:



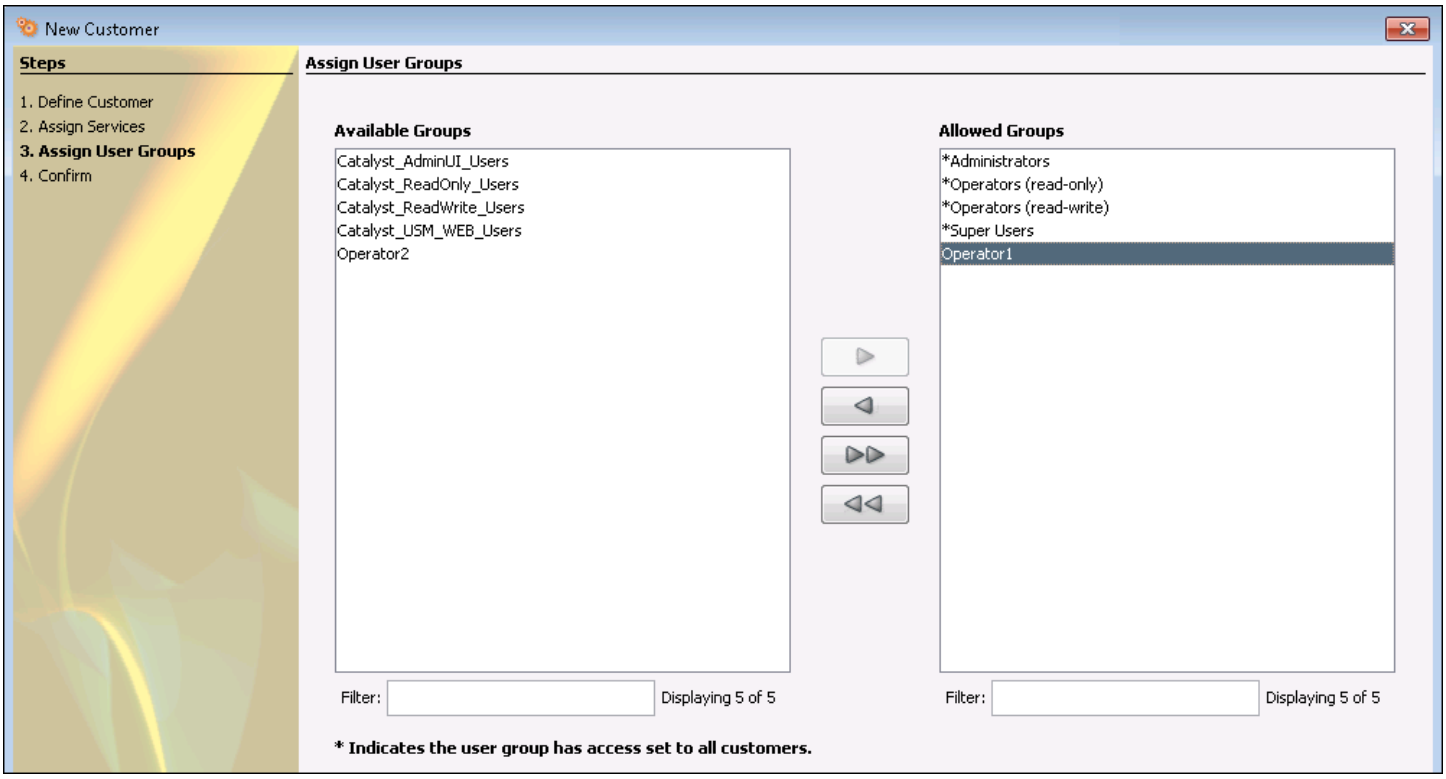
The administrator now sees the Company-A customer with the priority label Gold as shown in the following graphic:



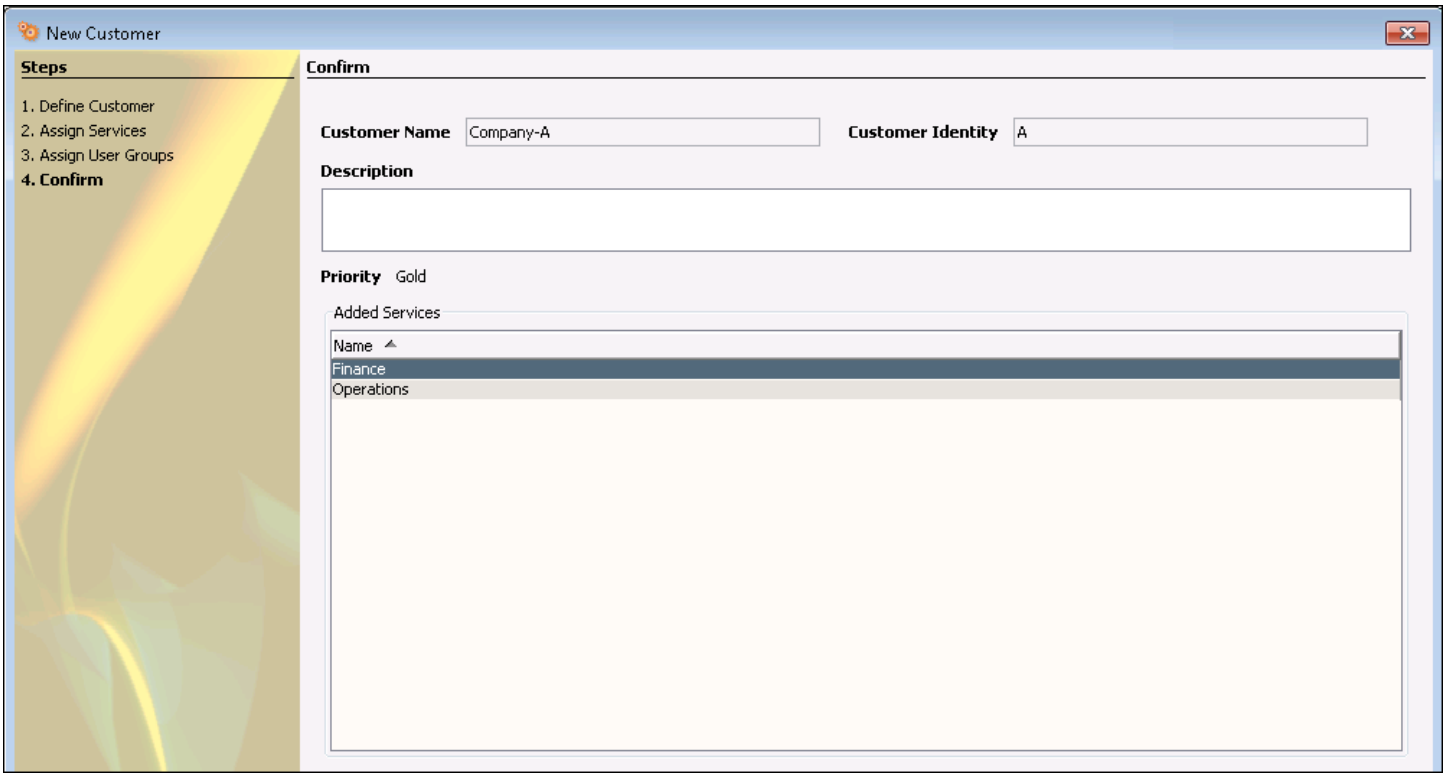
The administrator then assigns the Finance and Operations services to Company-A as shown in the following graphic:



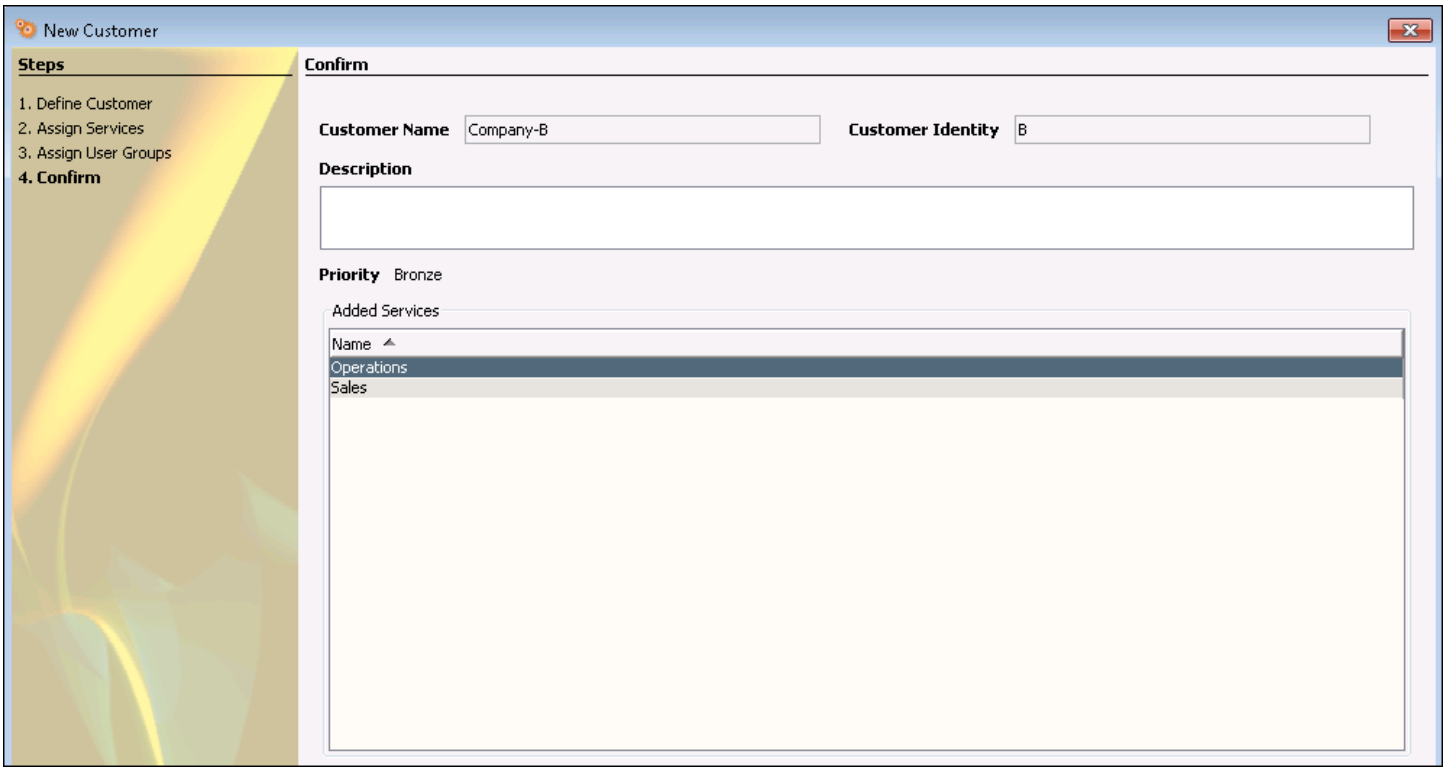
The administrator assigns the Operator1 user group to the Company-A customer in the Assign User Groups screen as shown in the following graphic:



The administrator reviews the Company-A customer information in the Confirm screen and completes the customer creation.

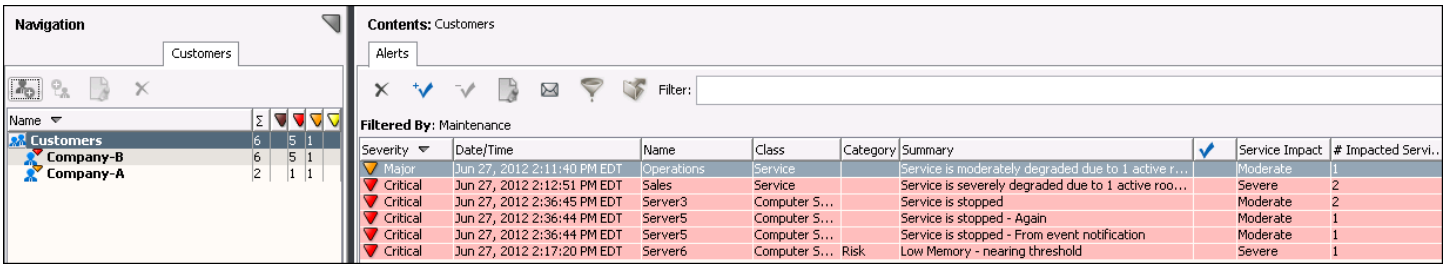


The administrator uses the same process to create the Company-B customer as shown in the Confirm screen. The administrator assigns the Operations and Sales services and sets the Priority to Bronze. The Confirm screen is shown in the following graphic:

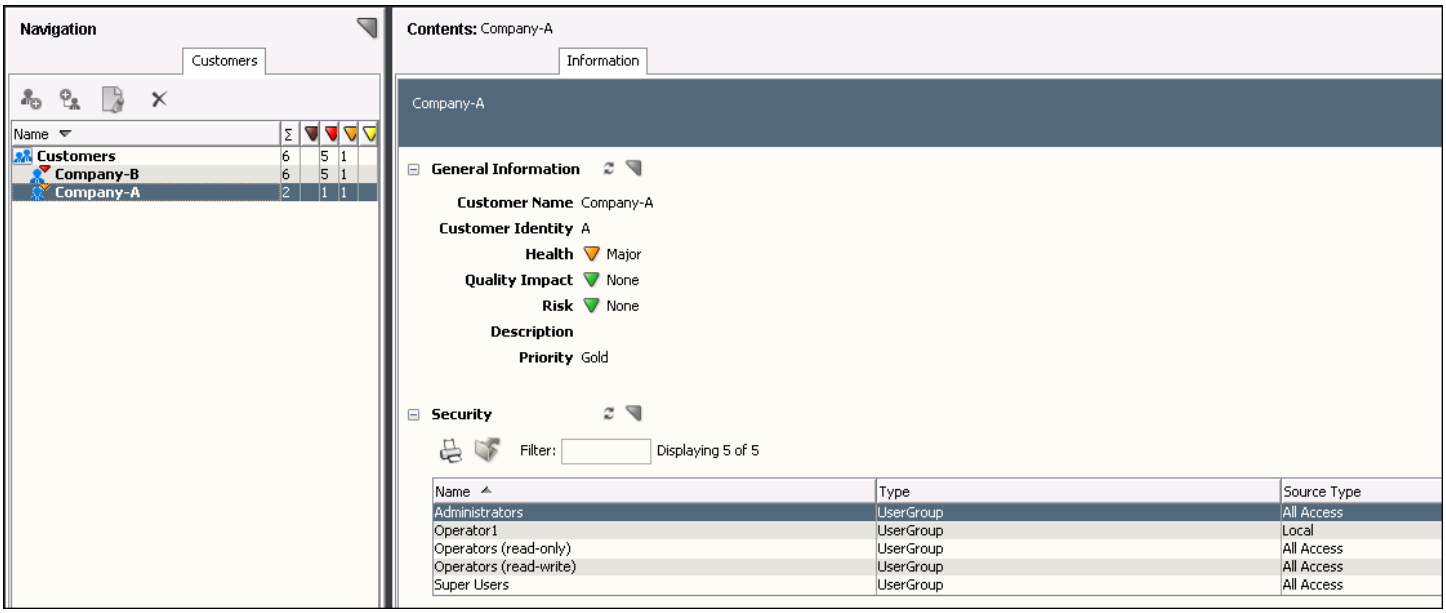


View Alerts on Customers

The administrator clicks the Customers tab and selects the parent Customers node in the customer tree, then clicks the Alerts tab in the Contents pane. The Alerts tab displays all alerts on the Company-A and Company-B customers as shown in the following graphic:

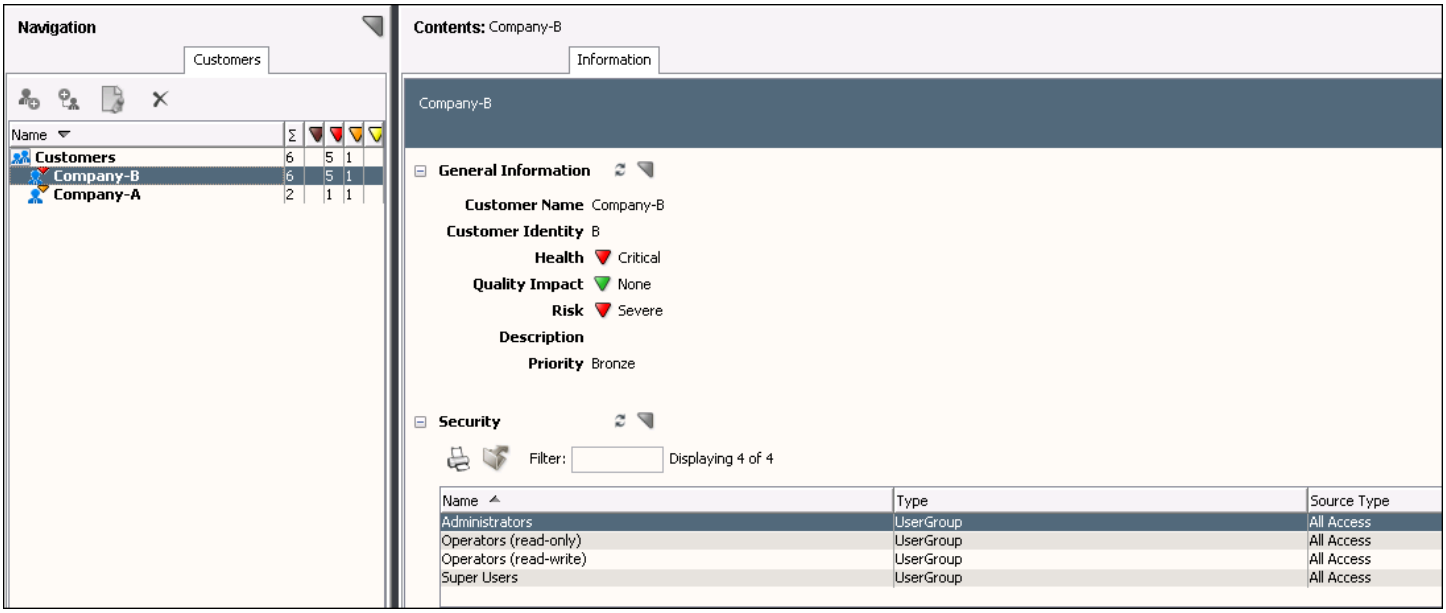


The administrator then selects the Company-A customer in the Navigation pane and clicks the Information tab to view the health status. The health status is Major as shown in the following graphic:



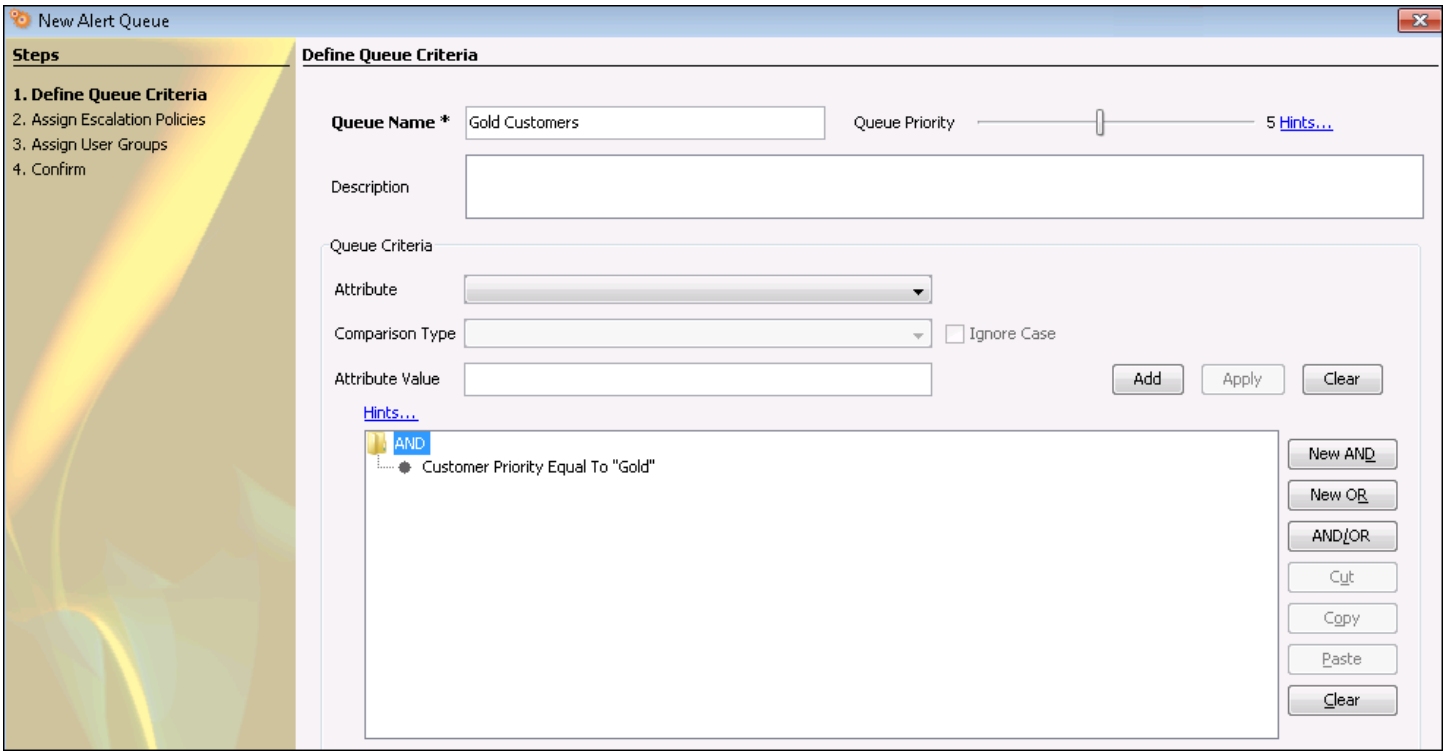
Remember that the customers available to a user depend on the user group the user is assigned to and that user group's customer assignment. Because the administrator has access to all customers and services, the administrator sees both Company-A and Company-B customers on the Operations Console. If an Operator1 user logs in to the Operations Console, the user sees only Company-A. The administrator assigned both Company-A and Company-B to the Operator2 user group, so an Operator2 user sees both the Company-A and Company-B customers.

The administrator then views the Information tab for the Company-B customer, which shows a health status of Critical as shown in the following graphic:

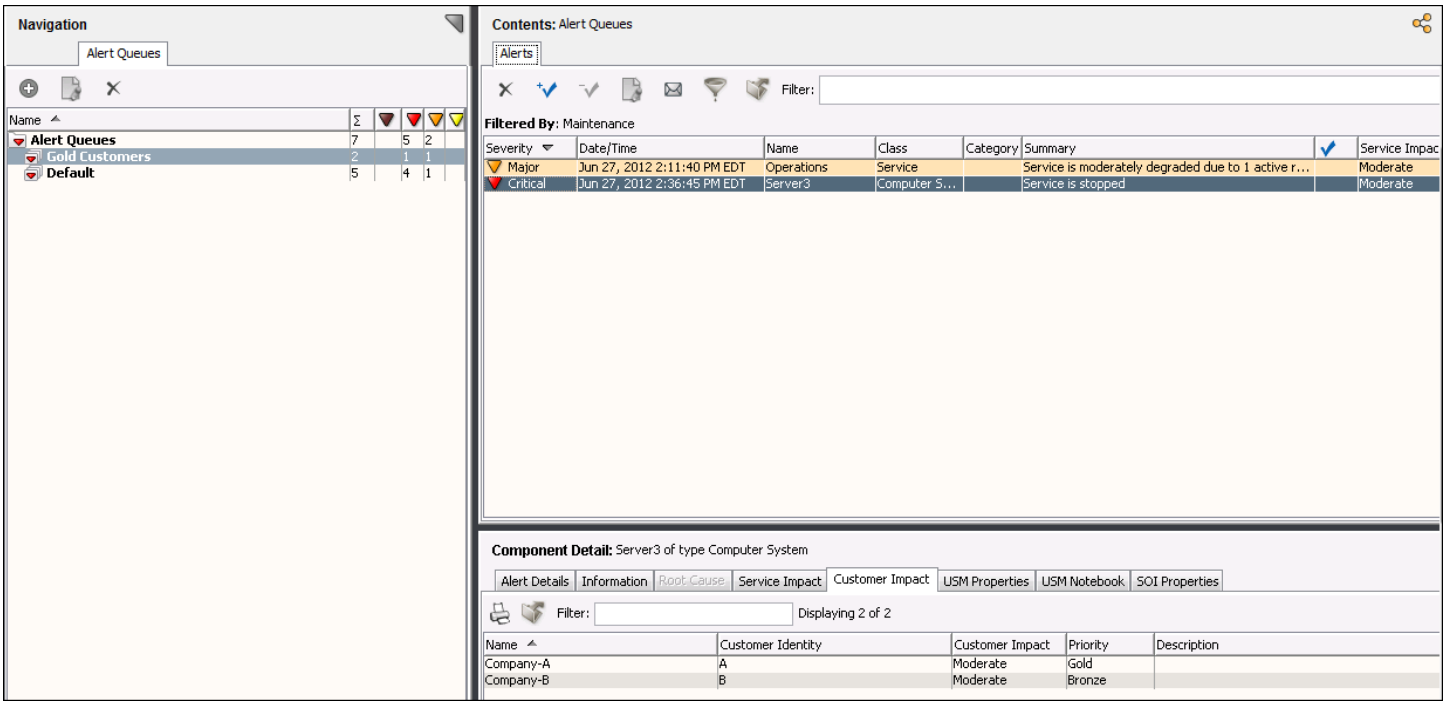


Create Alert Queues Using Customer Attributes

The administrator wants to create an alert queue named "Gold Customers" that shows the alerts on all Gold service level (priority) customers. The administrator launches the New Alert Queue wizard and defines the queue criteria where the Customer Priority attribute is equal to the Gold level. The completed Define Queue Criteria screen looks like the following graphic:

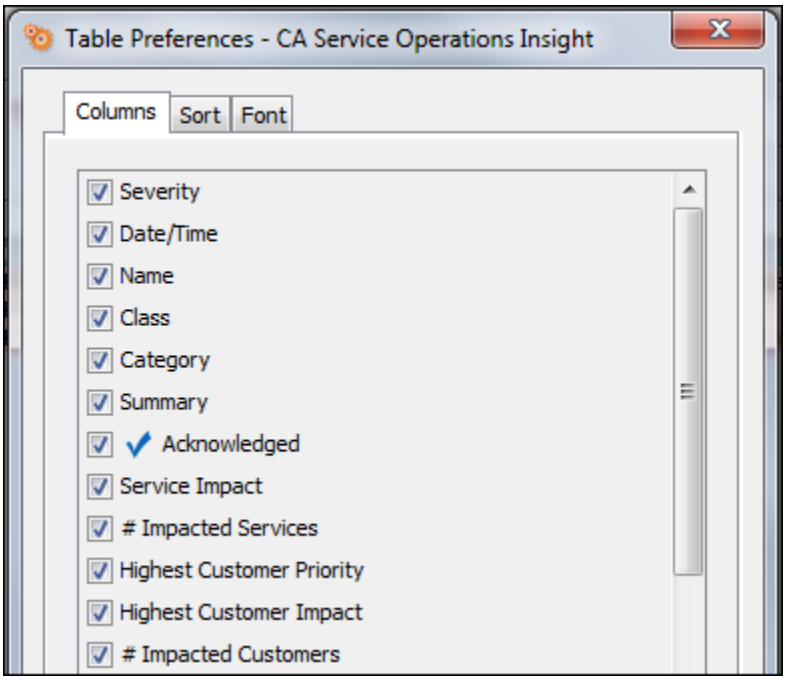


The administrator clicks the Alert Queues tab on the Operations Console, clicks the Alerts tab in the Contents pane, then clicks the Customer Impact tab in the Component Detail pane. The Operations Console appears like the following graphic:

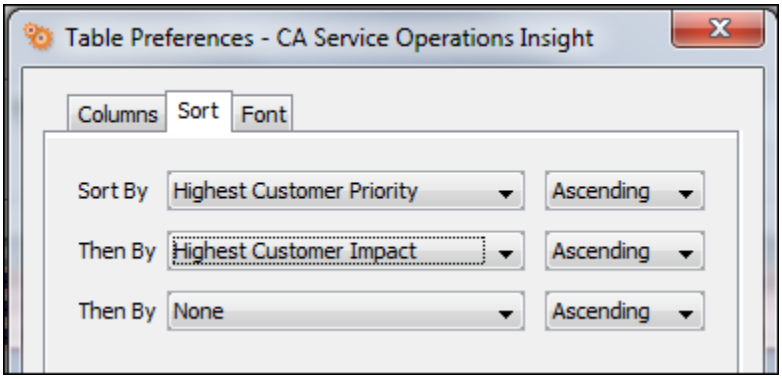


In this example, only the Company-A customer has a Gold customer priority, so only its alerts appear in the new Gold Customers alert queue. If the administrator adds a customer, say Company-C, with a Gold customer priority, its alerts dynamically appear in the queue.

The administrator adds additional columns to the Alerts tab by right-clicking a column heading and selecting Highest Customer Priority and Highest Customer Impact. The completed dialog is shown in the following graphic:



The administrator can then sort the Alerts tab by these columns also. For example, the following graphic shows that the administrator is sorting the Alerts tab first by Highest Customer Priority and then by Highest Customer Impact.

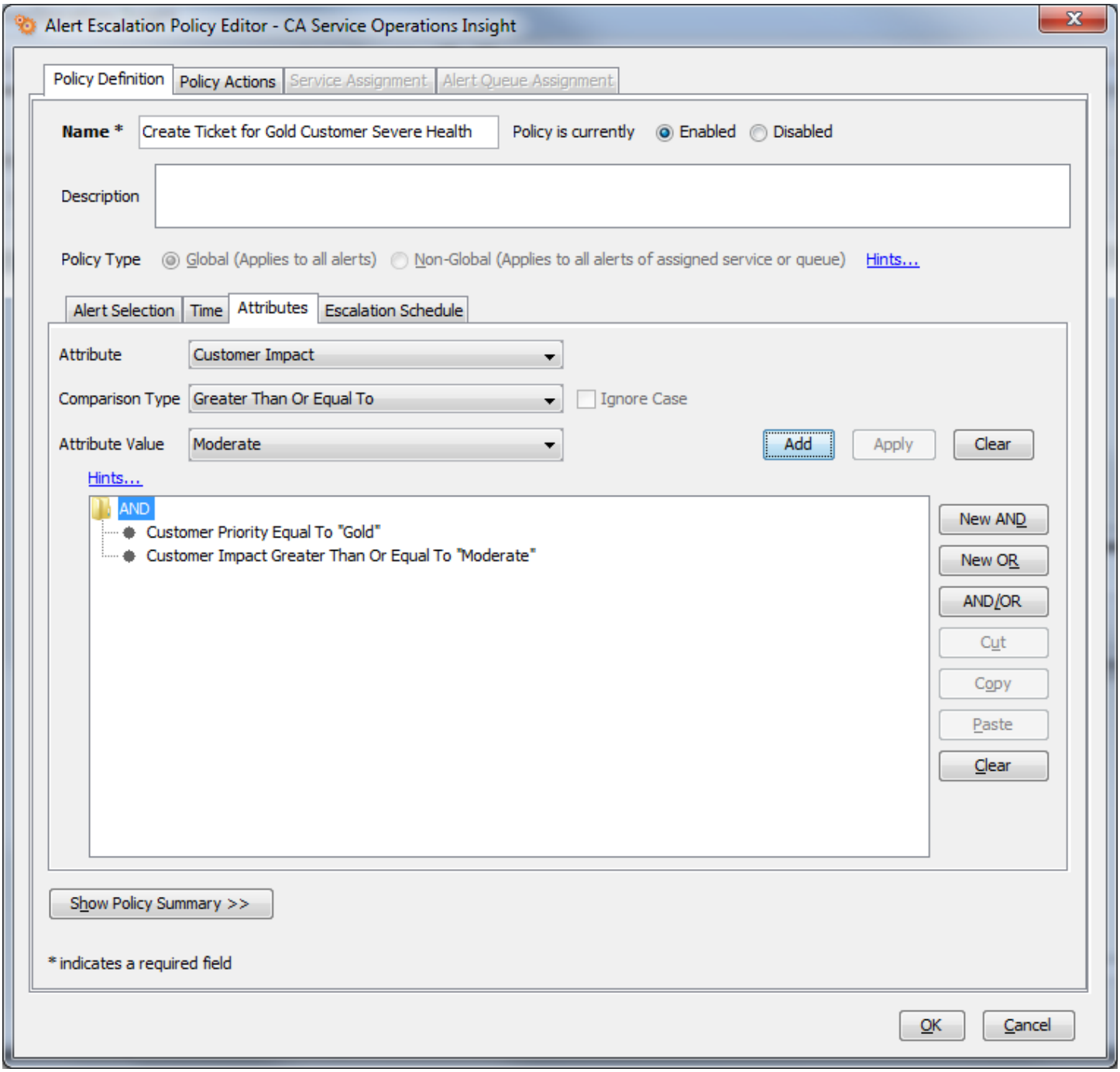


This sort would show the highest customer impact on Gold service level customers.

Create Escalation Policy Using Customer Attributes

The administrator also elects to use the customer attributes to create escalation policy. The administrator launches the Alert Escalation Policy Editor and uses the following attributes: Customer Priority Equal to "Gold" AND Customer Impact Greater Than or Equal To "Moderate".

The completed Attributes tab is shown in the following graphic:



The administrator then creates a Policy Action that automatically creates a help desk ticket when the policy conditions are met.

WSSSAServiceCmdV2 Command Usage

The WSSSAServiceCmdV2 command allows you to import and export services from one SOI system to another SOI system. Before importing or exporting a service, generate an encrypted password by executing WSSamEncryptCmd command.

NOTE

You can import and export a single service or all the services. We recommend you to import a service in a fresh SOI system.

Follow these steps:

1. Navigate to **<SOI_Home>\SOI\tomcat\bin** folder.
2. Encrypt the password by executing the following command:

```
WSSamEncryptCmd
```

3. To import the service, use the following command:

```
WSSSAServiceCmdV2 -h<wsHostName:wsPort> -u <wsUsername> -p <encrypted wsPassword> -
a<Import> -s <Service Instance ID|*> -f <fileName>
```

Example 1:
Importing a particular service:

```
WSSSAServiceCmdV2 -hlocalhost:7090 -usamuser -
p"EFbJeXR3zLsi9aPfOQ9FzRVOPPEUwCGxCWUFTNF4kxKD" -aImport -s"Service:servicename,WSMan
 New Service" -fc:/myServiceImport.xml
```

Where,
- **-h** specifies the hostname and port number
- **-u** specifies the username
- **-p** specifies the encrypted password
- **-a** specifies the export or import option
- **-s** specifies the service Instance ID
- **-f** specifies the service filename

Example 2:
Importing all the services:

```
WSSSAServiceCmdV2 -hserver1:7090 -usamuser -
p"EFbJeXR3zLsi9aPfOQ9FzRVOPPEUwCGxCWUFTNF4kxKD" -aImport -s* -fc:/
allServicesImport.xml
```

4. To export the service, use the following command:

```
WSSSAServiceCmdV2 -h<wsHostName:wsPort> -u <wsUsername> -p <encrypted wsPassword> -
a<Export> -s <Service Instance ID|*> -f <fileName>
```

Example 1:
Exporting a particular service:

```
WSSSAServiceCmdV2 -hlocalhost:7090 -usamuser -
p"EFbJeXR3zLsi9aPfOQ9FzRVOPPEUwCGxCWUFTNF4kxKD" -aImport -s"Service:servicename,WSMan
 New Service" -fc:/myServiceImport.xml
```

Example 2:
Exporting all the services:

```
WSSSAServiceCmdV2 -hserver1:7090 -usamuser -
p"EFbJeXR3zLsi9aPfOQ9FzRVOPPEUwCGxCWUFTNF4kxKD" -aExport -s* -fc:/
allServicesExport.xml
```

NOTE

The exported service file is located in the SOI server system.

Performance Results for WSSSAServiceCmdV2

This section provides the performance results of WSSSAServiceCmdV2:

The environment details are as follows:

Function	Memory (GB)	CPU
Export	4 GB	
Import	4 GB	

No of Services	Entities in each service	Export time taken	Import time taken
1	10	8 secs	14 secs
1	25	13 secs	36 secs
1	50	13 secs	29 secs
10	10	20 secs	50 secs
10	25	25 secs	1 min 44 secs
10	50	35 secs	2 min 55secs`
25	10	29 secs	1 min 34 secs
25	25	55 secs	3 min 35 secs
25	50	1 min 12 secs	6 secs
50	10	40 secs	3 min 6 secs
50	25	1 min 44 secs	6 min 45 secs
50	50	4 min 14 secs	12 min 28 secs

Alert Management

This section provides information about alert management administration procedures.

See the following topics for details:

Intended Audience

The information in this section is intended for any user who is responsible for managing any or all of the product alertactivity. This user base can include operators for a specific IT function, domain administrators, product administrators, help desk operators, and so on. This section refers to all these users collectively as an operator.

See also:

- Alert Management for Operators

Introduction to Alert Management

Contents

The topics in this section provide information about alert properties and and overview of service security.

Alert conditions that impact CIs in integrated domain managers appear in CA SOI as infrastructure alerts. CA SOI provides a powerful and unified alert console that provides operations personnel a view of the following items:

- All managed alerts with their associated services
- All unmanaged alerts with their associated alert queues
- All alerts that identify a degraded service quality or risk from infrastructure alert conditions.
  These service alerts originate from CA SOI based on analysis of the service model, impact policy, and active infrastructure alerts.

CA SOI alerts include details like the alert severity that the domain manager assigns, the number of services the alert impacts, and the impact of the alert condition on those services. In CA SOI, you can acknowledge, assign, annotate, escalate, and clear alerts.

You can work with alerts as follows:

- From a comprehensive alert management perspective using alert queues. The Alert Queues tab lets you define queues to create logical alert categories and manage all collected infrastructure alerts, not only those alerts that are associated with managed services.
- From a service-oriented perspective on the Services tab. The Services tab displays managed services and the alerts that are associated with those services.

You can use alert queues or a combination of both management methods as your infrastructure matures and you move toward a service-oriented management paradigm.

Operations personnel are responsible for the day-to-day tasks involved in monitoring the health of services and resources. This section explains the properties of an alert and contains basic alert management procedures.

Alert Properties and Extended Alert Information

CA SOI alerts contain the following properties. Some properties originate from the domain manager. Other properties (for example, service impact and number of impacted services) originate from CA SOI.

- **# Impacted Customers**
  Indicates the number of customers that the alert impacts. This number is based on the number of customers that are assigned to the service that the alert impacts.
- **# Impacted Services**
  Indicates the number of services the alert impacts based on the number of services its associated CI is included in.
- **Acknowledged**
  Indicates whether an operator has acknowledged the alert.
- **Assigned**
  Indicates the name of the operator that is assigned to the alert.
- **Category**
  Indicates whether this alert condition affects the quality or risk of the services it impacts.
- **Class**
  Indicates the class (USM type) of the CI the alert is associated with.
- **Date / Time**
  Indicates the date and time when this alert was generated.
- **Family**

Indicates the CI class family that the alert is associated with.

- **Highest Customer Impact**
  Indicates the highest impact value that the alert causes for an associated customer.
- **Highest Customer Priority**
  Indicates the highest customer priority of a customer that is associated with a customer associated with a related service.
- **Is Clearable**
  Defines whether the alert can be cleared in the domain manager. If you enabled the 'Respect Underlying MDR Clear Alert Setting', this property also determines whether the alert can be cleared in CA SOI.
- **Is Exempt**
  Indicates whether the alert is excluded from impact analysis calculations.
- **Maintenance**
  Indicates whether the CI associated with the alert is currently in maintenance mode.
- **Name**
  Indicates the associated CI that the alert condition impacted.
- **Service Impact**
  Indicates the impact, which is calculated by multiplying alert severity and the significance of the CI to the service. When multiple services are impacted, the most affected service is displayed.
- **Service Impact Value**
  Indicates the impact value of the service alert. This value is always a factor of 10. The Service Impact Value displayed in the Alerts table can be different from the service impact value for the corresponding service in the Topology tab. The Topology tab displays how the child objects impacted the service.
- **Severity**
  Indicates the alert severity that the originating domain manager assigned.
- **Source**
  Indicates the domain manager where the alert originated. The format is *MdrProduct_domainserver@connectorserver*. For example, CA:00005_spectrohost.ca.com@spectrohost.ca.com refers to a CA Spectrum connector installed on spectrohost.ca.com monitoring a CA Spectrum instance that is installed on the same system.
- **Source Alert ID**
  Indicates the ID number of the alert in the source domain manager. Only infrastructure alerts have a Source Alert ID, because service alerts are generated in CA SOI, not from a source domain manager.
- **Summary**
  Describes the alert condition.
- **Ticket ID**
  Indicates the ID of the associated help desk ticket.
- **Unmanaged**
  Indicates whether the alert is associated with any services. An unmanaged alert does not have a service association.
- **User Attribute (1-10)**
  Indicates any configured customized values. These attributes are blank by default, but you can send values to the attributes through Event Management. You can also customize the attribute names.

  TIP

  If you require more user attributes than are provided, as a best practice, you can assign multiple different values to a user attribute and use Matches Regex logic in escalation policy to filter out values and take action based on a certain type of value in the attribute.

You can also view the correlatable USM properties for the alert's associated CI:

- ModificationTime
- PrimaryIPV4Address
- PrimaryIPV4AddressWithDomain
- PrimaryIPV6Address
- PrimaryIPV6AddressWithDomain
- PrimaryMacAddress
- PhysSerialNumber
- BioSystemID
- Vendor
- AssetNumber
- PrimaryDnsName
- SysName

> **NOTE**
> For more information about USM properties, see USM schema documentation.

In addition to these properties, alerts have associated extended information such as annotations, update history, and escalation history in the Alert Details tab. This information provides a full audit trail of the manual and automated actions that are taken to help diagnose and remedy an alert condition.

- **Annotations**
  Indicates the interim steps that were taken to resolve the situation that caused an alert. These comments highlight the incident management process in real time, and they can provide information for the problem management process.
- **Update History**
  Indicates how the alert has evolved since alert creation. Updates can include changes in severity and properties (such as the acknowledged flag).
- **Escalation Action History**
  Indicates the automated actions that notify, diagnose, or remedy the problem and the results of those actions. For example, if an email notification is sent, confirmation that it was sent successfully is included. If a remote device was pinged, the results are included. Escalation history therefore provides a detailed audit trail of the automated actions taken in response to an alert condition.
- **Alert Queues**
  Show the alert queues to which the alert belongs.
- **User Defined Attributes**
  Displays the names and values of the user-defined attributes.
- **Most Recent Audit Trail**
  Provides a list of recent object actions.

Alerts and Security

The services for which you can view alerts are based on the user groups you are in and its service access privileges. Your administrator sets your user group and access privileges. Your administrator also sets your ability to manage alerts (acknowledge, annotate, assign, clear, send alert email, set alert ticket).

Alert Lifecycle

Contents

All alert data that is collected from connectors initially become events, but the lifecycle of an alert can vary. The following process summarizes the typical lifecycle of a message that is retrieved from a connector data source:

1. Connectors convert all types of messages (informational events, error messages, high-level alarms, and so on) from their domain manager to use USM alert properties.

2. Connectors store each USM alert entity as an event in the Event Store on the connector system.

   NOTE
   A record of each raw event with pre-normalized properties is also retained in the Event Store.
3. Event Management evaluates each event against defined policies. If the event matches policy criteria before the event becomes an alert, one of the following actions could happen:
   – The event could be discarded as part of a filter policy and prevented from becoming an alert.
   – The event could be enriched with additional information as part of an enrichment policy.
   – The event could be manually normalized to USM alert properties as part of a normalization policy.
4. Events with a severity greater than Normal that pass through Event Management processing without being discarded become alerts in the Operations Console that are associated with the affected CI.

   NOTE
   Events with a severity of Informational or Normal are automatically prevented from becoming alerts.
5. Alerts that are associated with a service are evaluated for the service impact. If the alert directly affects the service health, it becomes a root cause alert.
6. Alerts are evaluated against alert queue and escalation policies. If a match occurs, one of the following actions could happen:
   – The alert becomes a part of any alert queue with matching criteria.
   – If the alert matches escalation policy criteria, the associated escalation action occurs.
7. Alerts update based on user actions such as assignment, annotations, acknowledgment, and manual escalation.
8. An alert is cleared when one of the following actions occurs:
   – An operator manually clears the alert in the Operations Console.
   – A corresponding Normal alert occurs on the CI.
9. The alert disappears from the main Operations Console views and remains stored as a cleared alert for historical analysis.

Alert Lifecycle Examples

The following examples show how an alert's lifecycle can vary based on several factors:

Example 1: CA Spectrum network outage alarm

- The CA Spectrum connector receives an alarm from CA Spectrum indicating that a router is offline.
- The alarm is normalized and stored as an event.
- The event record progresses through Event Management processing without matching any policy. It becomes an infrastructure alert and displays associated with its router CI.
- The alert causes the associated Network service health to change to severely degraded. A service alert is created for the service degradation, with the infrastructure alert as the root cause.
- The alert triggers an escalation policy that sends an email to the technician responsible for the affected Network service.
- The technician fixes the router and clears the alert.

Example 2: Event log authentication failure

- The Event connector receives an event from the Windows Event Log indicating that an authentication failure occurred.
- The event is normalized and stored in the Event Store.
- The event matches an Event Management filter policy that discards all events from the Windows Event Log with a Minor severity. The event is discarded and never appears as an alert on the Operations Console.

Alert and Event Visualization

The following interfaces let you view and interact with events and alerts:

- Operations Console
  Displays all managed and unmanaged alerts in the context of their associated services, customers, and queues. You can perform all operations on alerts from the Operations Console and view comprehensive alert details.The Operations Console also includes the Event Policies dialog, which lets you search for and view events and create event policies.
- Mobile Dashboard
  Lets you view managed and unmanaged alerts in the context of their associated services and queues. You can view basic alert properties and can escalate alerts using existing actions.
- USM Web View
  Lets you perform detailed searches for alerts and subscribe to RSS feeds that provide notification when alerts are created or updated.
- Dashboard
  Displays a summary of services and their alerts.
- Alert Management Reports
  Display report data based on alert response time, ticketed alerts, and top alert sources.

Alert Management Administration

This section describes how to configure alert management settings.

Configure Alert Escalation Integrations

As an administrator, you manage the integration settings for email, help desk, CA Process Automation, mobile dashboard, and USM Web view on the Administration tab. Establish all necessary integrations before creating escalation actions.

Follow these steps:

1. Access the Dashboard, and click the Administration tab.
2. Expand CA Service Operations Insight Manager Configuration and the SA Manager server name, and click one of the following items:
   - Email Configuration
     Defines email server connection information for sending email through escalation actions.
   - Help Desk Configuration
     Defines help desk connection information for opening help desk tickets through escalation actions.
   - Process Automation Server Configuration
     Defines CA Process Automation connection information for running automated CA Process Automation processes through escalation actions.
   - Mobile Dashboard Server Configuration
     Defines Mobile Dashboard connection information for using a runtime token to embed its URL in escalation action output, such as an email or help desk ticket property.
   - USM Web View Configuration
     Defines USM Web View connection information for accessing the interface from the Dashboard and using a runtime token to embed its URL in escalation action output, such as an email or help desk ticket property.
3. Enter all necessary information and click Save.

Configure Alert Management Global Settings

As an administrator, you can manage the alert-related Global settings for maintenance mode, cleared alerts, escalation policies and actions.

Follow these steps:

1. Access the Dashboard, and click the Administration tab.

2. Expand CA Service Operations Insight Manager Configuration and the SA Manager server name, and click Global Settings.
3. Make the appropriate selections in the following drop-down lists and fields:
   - **Maintenance Mode Settings**
     - **Propagate Maintenance Impact**
       Specifies whether an alert impact is propagated to the parent objects when the associated CI is in maintenance mode.
       Select Yes if you want alerts for CIs in maintenance mode to propagate the impact to the parent objects.
       Select No if you do not want alerts for CIs in maintenance mode to propagate the impact to the parent CIs. In this case, the alerts are still generated and the CIs display the state.

       NOTE
       Changing a Global Settings flag while the SA Manager is running only impacts future alertsit does not affect existing alerts and associated services.

       Default: No
     - **Unknown Alert Setting**
       Controls the severity that CA SOI assigns to incoming alerts that have a severity of Unknown. A setting of Ignore prevents the alerts from appearing in the Operations Console.
       Default: Minor
   - **Cleared Alerts Setting**
     - **Reload Cleared Alerts Setting**
       Controls if alerts that CA SOI clears are reloaded from their source domain managers upon restart of CA SOI or the domain managers.
       Select yes so previously cleared alerts are reloaded into CA SOI and displayed in the Operations Console.
       Select No so previously cleared alerts that the domain managers resend are ignored and omitted from the Operations Console.
       Default: No
     - **Enable Pending Cleared Alerts**
       Controls whether to place an alert in a Pending Clear state before being fully cleared. For more information, see Enable Pending Clear State for Alerts.
     - **Respect Underlying MDR Clear Alert Setting**
       Controls whether CA SOI respects the clear alerts settings imported from the domain manager in the alert's IsClearable property. Set this property to Yes to prevent users from being able to clear alerts in CA SOI that they would not be allowed to clear in the domain manager. Set this property to No to all clearing of all alerts regardless of the domain manager's imported IsClearable setting.
   - **Escalation Policy and Action Settings**
     - **Perform Action Retries**
       Controls if CA SOI retries escalation actions when they fail.
       Select Yes to retry in the number of minutes entered in the Retry Frequency field. Also retries for the total number of days entered in the Retry Duration field.
       Select No to quit an escalation action after one failed attempt.
       Default: Yes
     - **Retry Frequency (minutes)**
       Defines the number of minutes between escalation action retries after a failed attempt.
       Default: 30
     - **Retry Duration (Days)**
       Defines the number of days CA SOI continues to retry escalation actions after failed attempts. At the end of this duration, CA SOI stops attempting failed escalation actions.
       Default: 2
4. Click Save.
5. Restart the CA SAM Application Server service.

Set Root Cause Analysis Mode

Contents

Root Cause Analysis Modes

An administrator sets the root cause analysis mode in the Global Settings page in CA SOI. The root cause analysis mode determines where root cause analysis is performed. Because CA SOI operators do not have access to the Administration tab on the Dashboard, the administrator should communicate the root cause analysis mode that is set. As a best practice, operators should add the MDR Root Cause and MDR Symptoms columns to the Contents pane, so the operators can easily see if the domain manager is determining the root cause and symptoms. For more information about customizing the Operations Console, see Operations Console Customization. You can set root cause analysis to be performed by:

- CA SOI
- The domain manager (such as CA Spectrum)
- A combination of both.

For the domain manager or the combined root cause analysis settings, CA SOI relies on a connector to provide the root cause analysis information. For connector support of this feature, see the product-specific *Connector Guide* that is provided with the connector package.

> **NOTE**
> The Domain Manager Derived and Combination modes require that the domain manager and its connector both support sending the root cause analysis information to CA SOI. For current root cause analysis support see the product-specific *Connector Guide* that is provided with each connector.

The following root cause analysis modes are available.

- **CA SOI Derived**
  A CA SOI derived root cause analysis is performed in CA SOI. CA SOI determines the root cause, symptoms, and the path-of-impact and ignores the domain managers' (such as CA Spectrum) root cause determinations. This method is beneficial when all CIs that are critical to the root cause analysis are modeled in and managed by CA SOI.
- **Domain Manager Derived**
  A domain manager derived root cause analysis uses one or more domain managers (such as CA Spectrum) that determine the root cause, symptoms, and the path-of-impact. CA SOI does not perform root cause analysis and uses the domain manager's root cause determination. This method is beneficial in the following example situations:
  - The domain manager determines that a CI is the root cause and CA SOI is not managing the CI.
  - Multiple domain managers contribute to a CI, but only one domain manager has determined the root cause CI. CA SOI is not managing the CI.
  - CA SOI does not have service models that manage CIs that the domain manager reports alerts and root cause analysis on.
- **Combination (CA SOI and Domain Manager Derived)**
  A combination of CA SOI and domain manager derived root cause analysis acts as a boolean OR when either CA SOI or a domain manager determines a CI is the root cause. If either CA SOI or the domain manager determines a root cause, CA SOI uses that root cause. The Operations Console may display more than one root cause if CA SOI and the domain manager determine different root causes. If neither a domain manager nor CA SOI determines a root cause, then there is no root cause. The symptoms are determined by either CA SOI or the domain manager. This method is beneficial when CA SOI does not have service models that manage CIs that the domain manager reports alerts on.

> **NOTE**
> Combination mode does not require that all domain managers support root cause analysis.

Examples Root Cause Analysis by Mode

The examples in this section show how the root cause and symptoms display in the Operations Console, depending on the Root Cause Analysis global setting.

The three examples that follow use a service that is named "Service_A." This service is modeled in CA SOI and contains a server that is named "server1" and a router that is named "router1." The examples use the Sample Connector for its data. The following graphic shows the service topology:

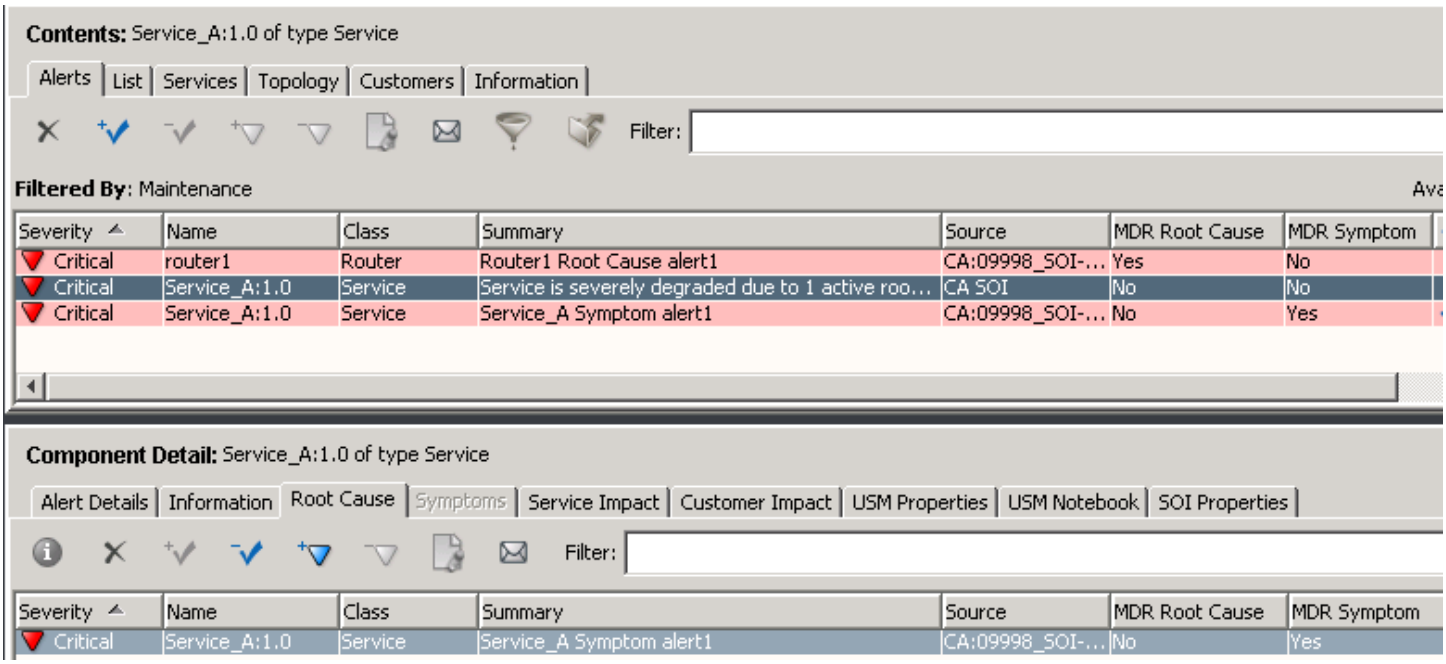
The domain manager is CA Spectrum, which is configured to determine root cause, and symptoms. In each example, you will see how changing the root cause analysis mode changes the root cause and symptoms on the Operations Console.

Example CA SOI Derived Root Cause Analysis

In this example, the administrator configures the Root Cause Global Setting to CA SOI for the root cause analysis. In this mode, CA SOI determines the root cause and ignores the root cause determined by the domain manager (CA Spectrum).

CA SOI uses the service impact to determine the root cause. In this case, the administrator has not modified the default significance for the CIs, so CA SOI determines that "Service_A" (with a significance of 10) is the root cause and not "router1" (with a significance of 9). As you will see in subsequent examples, the domain manager (CA Spectrum) determines a different root cause than CA SOI. This demonstrates not only the importance of using significance to determine service impact, but also determining which root cause analysis mode you set.

Because CA SOI determined that "Service_A" is the Root Cause, selecting the "Service_A" alert or CI shows "Service_A" (with CA SOI as the source) as the root cause.



However, if you select "Service_A" with CA Spectrum as the Source, the Root Cause tab is empty. This is because you are using CA SOI to determine root cause.

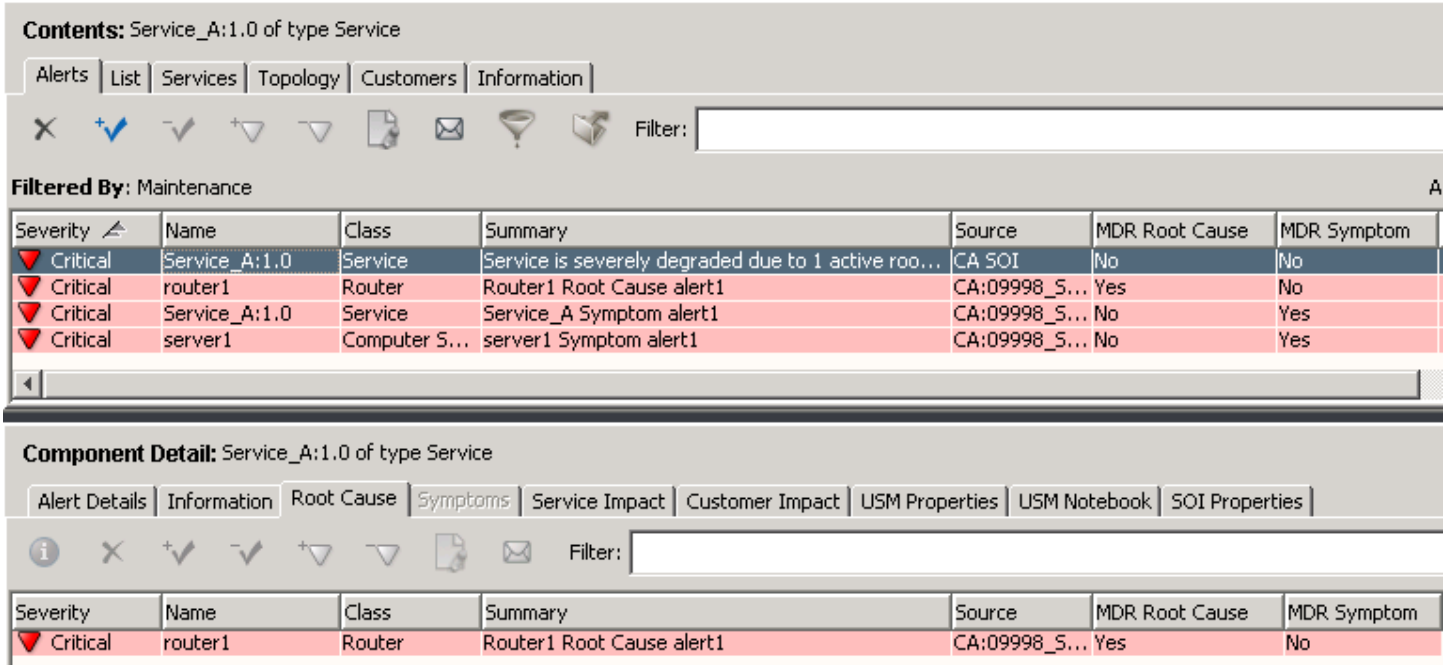
Therefore, in CA SOI mode, verify that you are viewing alerts with CA SOI as the Source in the column. CA Spectrum determines the symptoms in this example, not CA SOI. CA SOI calculated the root cause.

The Symptoms tab is disabled because CA SOI has not determined any symptoms, although CA Spectrum has. You can still see "Service_A" as a symptom, but that is according to CA Spectrum, which is not used in root cause analysis.

Example Domain Manager Derived Root Cause Analysis

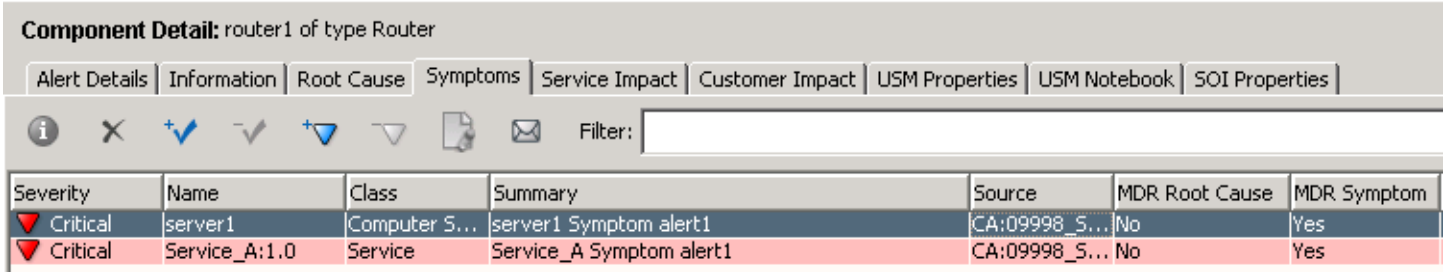
In this example, the administrator configures the Root Cause Analysis Setting to Domain Manager. In this case, the domain manager is CA Spectrum, so CA Spectrum determines the root cause analysis and symptoms.

The MDR Root Cause and MDR Symptom columns, which the user manually added, show the alerts that CA Spectrum determines are the root cause and symptoms, respectively.



CA Spectrum determined that "router1" is the root cause and "server1" is the symptom. Therefore, if you select Service_A (with either CA Spectrum or CA SOI as the source) or "server1", the Root Cause tab shows "router1" as the Root Cause. This is in contrast to the CA SOI derived root cause analysis example, where CA SOI determined that "Service_A" was the root cause.

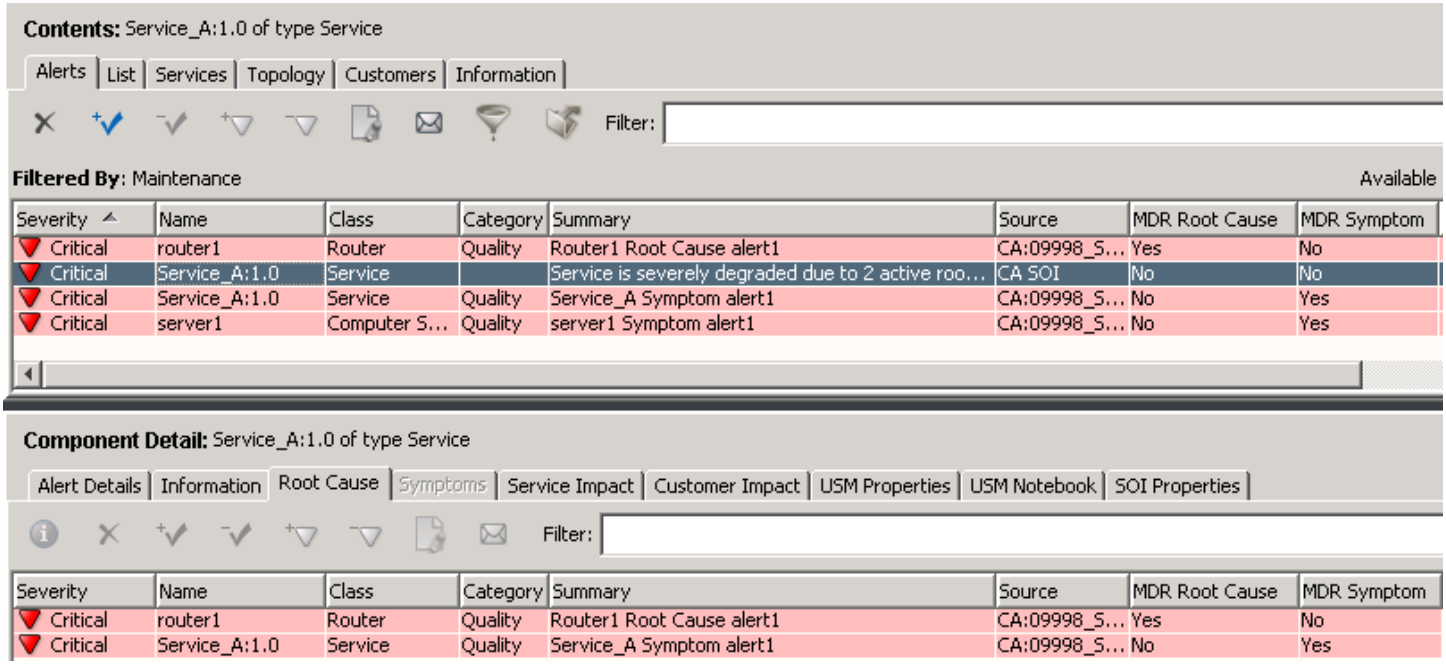
If you select "router1", which is the root cause, the Symptoms tab shows "server1" and "Service_A" as the symptoms as determined by the domain manager, CA Spectrum.



Example Combination (CA SOI and Domain Manager) Derived Root Cause Analysis

In this example, the administrator configures the Root Cause Analysis Setting to Combination mode. In Combination mode, either or both CA SOI and the domain manager can determine the root cause.

As you recall from the previous examples, CA SOI determined that "Service_A" is the root cause and the domain manager, CA Spectrum, determined that "router1" is the root cause.



The Operations Console indeed shows that there are actually two root causes: "Service_A" with CA Spectrum as the source and "router1" with CA Spectrum as the source. The MDR Root Cause and MDR Symptom columns also show what CA Spectrum determined.

Exempt Alerts from Impact Analysis

Contents

As administrator, you can remove an alert from participating in the impact analysis calculations.

If you do not want an alert condition to affect the overall status of a service, you can exempt an alert. You can exempt infrastructure alerts only, not service or policy alerts. When you exempt an alert, the name of the alert is dimmed to indicate exemption.

NOTE

- Service alerts imported from the CA SOI domain connector are treated similar to alerts imported from any domain manager, and therefore can be exempted.
- If an alert is in exempt status, the CI for that alert is changed to Green in the Topology view.

You can exempt alerts using any of the following methods:

- On the Operations Console.
- With Event Management.
- With customized connector policy.
- On a mobile device.

Exempt Alerts on the Operations Console

You can manually exempt alerts on the Operations Console. The exempted alerts appear dimmed in the Alerts tab.

Follow these steps:

1. Select a service in the Services Tab or an alert queue in the Alert Queues tab.

2. In the Contents pane Alerts tab, select an alert or press Ctrl/Shift + click to select multiple alerts.
3. Click the Exempt selected alerts

   icon.
   The exempted alerts are dimmed.
   Consider the following items:
   – You can also right-click an alert and exempt the alert with the context menu.
   – To unexempt the alerts, repeat the same steps and click the Unexempt selected alerts

     icon.
   – You can add the Exempt column to the Alerts tab. The column displays Yes or No for Exempt/Unexempt alerts. For more information about adding columns, see Operations Console Customization.
   – The Alert Details tab lets you view and change the Exempt status of the selected alert.

Exempt Alerts with Event Management

You can use the Event Management UI to exempt alerts from the impact analysis calculations. To do so, you define criteria in the Event Management UI that matches a certain class of alerts and then specify the exempt action. Specifying the exempt action sets the isExempt property to true, which implies that the alert is not considered for any impact calculations.

For example, consider a scenario where the CA SystemEDGE monitoring generates alerts (through CA Spectrum) on FileSystem D: due to low space. The FileSystem is not modeled within CA SOI as a separate CI, but rather is managed as part of a ComputerSystem CI. The customer understands that this FileSystem does not impact any services in the infrastructure, and wants to prevent the alert from participating in the CA SOI state management. The event management enhancement enables you to define criteria that matches these FileSystem alerts (for example, *matches(Summary, 'FileSystem D:')*), and specify an associated exempt action.

> **NOTE**
> For more information about how to work with event policies, see Working with Event Policies and Actions.

Follow these steps:

1. Open the Operations Console, and select Tools, Event Policies.
2. Specify the criteria that you want to use.
   For example, specify *matches(Summary, 'FileSystem D:')* in the Event Pattern 1 field of the Search Criteria section for the example scenario.
3. Click Create Policy.

   > **NOTE**
   > Only one user can create an event policy at one time. If any other logged in user has the Create Event Policy wizard open, an error message appears saying that another user has a lock on the functionality. The user must close the dialog before you can create event policies.
4. Enter a name in the Policy Name field, and select the Exempt Event option.

   > **NOTE**
   > The policy name cannot have more than 128 characters. Also, the name cannot contain any characters listed as prohibited on the tooltip that appears when you hover over the Policy Name text.
5. Click Next.
   The Select Data Sources page opens.
6. Perform one of the following steps:
   – Select the Save policy only option, and click Finish.

The policy is saved but not deployed, and it appears in the Saved Policies section of the Events tab. This policy does not dynamically evaluate and process events according to its defined search patterns and resultant actions until you deploy it. Skip the rest of this procedure if you want to save the policy without deploying.

– Select the Save and Deploy policy option.
  The list of available connectors becomes available.

7. Select the connectors on which to deploy the policy, move them to the Selected Data Sources pane, and click Finish. The policy is created, and it appears in the Deployed Policies section of the Events tab and under the appropriate data source in the Data Source section. After the policy is deployed to a connector, you can check the connector extensions (<SOI_HOME>\Core\CatalogPolicy\extensions) to verify that a policy extension, where the Exempt Event action was selected, exists. The naming convention for the policy extension file is *ConnectorPolicyName.EventPolicyName*.xml.

Exempt Alerts with Customized Connector Policy

To have a connector set the isExempt property, customize the connector policy (<SOI_HOME>\resources\Core\Catalogpolicy*name*_policy.xml). You can customize the policy by adding a normalization mapping rule to the connector Alert EventClass.

> **NOTE**
> For more information about the normalize policy operation, see Normalize Operation.

Follow these steps:

1. Navigate to the <SOI_HOME>\resources\Core\Catalogpolicy folder and locate the connector policy file; for example, spectrum_policy.xml.
2. Open the policy file in a text editor.
3. Add the normalization mapping rule. An example snippet is provided as follows:

```
<EventClass name='Alert'>
   <Normalize>
   ...
     <Field output='isExempt' type='map' input='someProperty'>
       <mapentry mapin='someRegexPattern' mapout='true'/>
       <mapentry mapin='.*' mapout='false'/>
     </Field>
   ...
   </Normalize>
</EventClass>
```

   This mapping rule causes the policy to compare the input (*someProperty*) with the specified pattern (*someRegexPattern*). If the value of someProperty matches the regular expression (Regex) specified in someRegexPattern, the property isExempt is set to *true*. You can define any number of map entries to allow multiple Regex patterns to be easily mapped. If no entry is matched, the policy assigns the default value *false*, which implies that the alert participates in the impact analysis.
4. Save the changes in the policy file.
5. Restart the CA SAM Integration Services service.

Exempt Alerts on a Mobile Device

You can exempt or unexempt alerts on the Mobile Dashboard.

Follow these steps:

1. Access the Mobile Dashboard.
2. Tap the Status Indicator to the right of a given service.
3. Tap an alert.
4. Tap Exempt Alert or Unexempt Alert and confirm the operation.

Hide Alerts in Maintenance Mode

Contents

As an administrator, you can configure a setting to determine if alerts on services in maintenance mode are hidden. CA SOI does not automatically place CIs for a service in maintenance mode because the CIs may belong to another service that is not in maintenance mode. Consequentially, operators could receive alerts for a service that is in maintenance mode. The operators then mistakenly open tickets against the alerts.

CA SOI hides the alerts based on the following rules:

- If a service has subservices that are not in maintenance mode and you select the parent service in the Services tab:
  - The Alerts tab does not show the subservice alerts when viewing in the context of the parent service in the Topology view.
  - The Alerts tab does not display alerts for any of subservices under this parents context because the parent is in maintenance mode.
- If the parent service is not in maintenance, but the subservice is in maintenance:
  - The Alerts tab of the selected parent service hides any alerts that are for CIs only belonging to the subservice.
  - The Services root node does not hide any alerts for CIs in maintenance mode if the CIs are not part of other subservices that are not in maintenance.

Consider the following items:

- This feature acts as a visual filter only. Alerts are not deleted and do not impact features such as propagation policy.
- Maintenance mode affects alerts displayed in the Alerts tab only in conjunction with the services selected in the Services tab. No other views are affected (for example, alert queues).

The topics that follow show how to change the setting and example situations to show how alerts display in different service maintenance mode situations.

With service-specific escalation policy, the escalation policy continues to work as before even if alerts do not show for a particular service. The Alert Escalation Policy Editor provides more options for maintenance mode. For more information about the Alert Escalation Policy Editor, see the How to Create Escalation Policy.

If you want to exclude specific CIs in a subservice from escalation policies, put the subservice into maintenance mode also and clear the Alerts for CIs in maintenance mode option in the Alert Escalation Policy Editor.

Configure Maintenance Mode Alert Setting

You can configure if CA SOI hides alerts for parent services in maintenance mode on the Global Settings page. For more information about configuring the Global Settings page, see Configure Alert Management Global Settings.

Follow these steps:

1. Enter the following URL in your web browser:
   ```
   http://<samanagerServer>:<samanagerPort>/sam/admin/hideMaintModeAlerts.jsp
   ```
2. For the Maintenance Mode Settings, select Yes or No:
   - **Yes**
     Select Yes if you want CA SOI to hide alerts on CIs with a parent service in Maintenance Mode.
   - **No**
     Select No if you want CA SOI to display all alerts on CIs regardless of Maintenance Mode.

   Default: No
3. Click Save.
4. Restart the Operations Console for the change to take effect.

Maintenance Mode Examples

The following examples show how CA SOI hides alerts with different services in maintenance mode. Assume the following conditions for the examples:

- All CIs have some alert severity on them.
- The setting for "Hide alerts when in Maintenance mode" is set to Yes.
- You have access to all services.

You have the following service structure in the Services tab of the Navigation pane:

```
Services
Service_1
     Service_3
        CI_A
        CI_B
        CI_C
        CI_D
     Service_4
        CI_A
        CI_B
        CI_E
        CI_F
Service_2
     CI_G
     CI_H
       Service_3
          CI_A
          CI_B
          CI_C
          CI_D
```

Note the following conditions in the service structure:

- Services is the root node on the Services tab service tree.
- Service_1 has two subservices: Service_3 and Service_4
- Service_2 has two CIs and one subservice: Service_3

In the next topics, you will see how putting different services into maintenance mode impacts the alerts that you can see.

Parent Service in Maintenance Mode

In this example, Service_1 is in Maintenance Mode. Therefore subservices Service_3 and Service_4 have alerts on their CIs hidden. However, Service_3 is also a subservice of Service_2. In the context of Service_2, alerts on CIs for Service_3 appear.

```
Services
Service_1--Maintenance Mode
     Service_3
        CI_A
        CI_B
        CI_C
        CI_D
     Service_4
        CI_A
        CI_B
```

```
        CI_E
        CI_F
Service_2
     CI_G
     CI_H
       Service_3
          CI_A
          CI_B
          CI_C
          CI_D
```

The following table shows which alerts on CIs you can see with Service_1 in Maintenance Mode and is based on the service you select in the Navigation pane.

Service selected in Services tab	Alerts display for these CIs	Comments
Services	CI_A, CI_B, CI_C, CI_D, CI_G, CI_H	All alerts on CIs under Service_1 are hidden. All CIs under Service_2 appear.
Service_1	All alerts on CIs are hidden	All alerts on the Service_1 CIs are hidden because Service_1 is in Maintenance Mode.
Service_3 (subservice to Service_1)	All alerts on CIs are hidden	All alerts on the Service_1 CIs are hidden because Service_1 is in Maintenance Mode.
Service_4	All alerts on CIs are hidden	All alerts on the Service_1 CIs are hidden because Service_1 is in Maintenance Mode.
Service_2	CI_A, CI_B, CI_C, CI_D, CI_G, CI_H	Service_2 is not in Maintenance Mode, so all alerts on the CIs appear.
Service_3 (subservice to Service_2)	CI_A, CI_B, CI_C, CI_D	Service_3 is not in Maintenance Mode, so all alerts on the CIs appear.

Subservice in Maintenance Mode

In this example, Service_4 is in Maintenance Mode. In this configuration, alerts on CIs under Service_4 are hidden, when you select Service_1 or Service_4. However, CI_A and CI_B are shared with Serivce_3, which is not in maintenance mode, so alerts on those CIs show in the context of Service_1 and Service_3.

```
Services
Service_1
     Service_3
        CI_A
        CI_B
        CI_C
        CI_D
    Service_4--Maintenance Mode
        CI_A
        CI_B
        CI_E
        CI_F
Service_2
     CI_G
     CI_H
       Service_3
```

```
        CI_A
        CI_B
        CI_C
        CI_D
```

The following table shows which alerts on CIs you can see with Service_4 in Maintenance Mode and is based on the service you select in the Navigation pane.

Service selected in Services tab	Alerts display for these CIs	Comments
Services	CI_A, CI_B, CI_C, CI_D, CI_G, CI_H	Alerts on CI_E and CI_F are hidden. Alerts on all other CIs appear.
Service_1	CI_A, CI_B, CI_C, CI_D	CI_A and CI_B are shared with Service_3 and Service_4, but the alerts show on these CIs because Service_3 is not in Maintenance Mode.
Service_3 (subservice to Service_1)	CI_A, CI_B, CI_C, CI_D	Alerts show on Service_3 because Service_3 is not in Maintenance Mode.
Service_4	All alerts are hidden	Service_4 is in Maintenance Mode so all alerts are hidden for the CIs.
Service_2	CI_A, CI_B, CI_C, CI_D, CI_G, CI_H	No services are in Maintenance Mode, so all alerts on the CIs appear.
Service_3 (subservice to Service_2)	CI_A, CI_B, CI_C, CI_D	No services are in Maintenance Mode, so all alerts on the CIs appear.

Shared Subservice in Maintenance Mode

In this example, Service_3 is in Maintenance Mode. Note that Service_3 is a subservice to parent services Service_1 and Service_2. Alerts on Service_3 CIs are hidden. However, CI_A and CI_B are shared with Service_3 and Service_4 and Service_4 is not in maintenance mode. In the context of Service_4, alerts on CI_A and CI_B appear.

```
Services
Service_1
   Service_3--Maintenance Mode
        CI_A
        CI_B
        CI_C
        CI_D
     Service_4
        CI_A
        CI_B
        CI_E
        CI_F
Service_2
     CI_G
     CI_H
    Service_3--Maintenance Mode
          CI_A
          CI_B
          CI_C
          CI_D
```

The following table shows which alerts on CIs you can see with Service_3 in Maintenance Mode and is based on the service you select in the Navigation pane.

Service selected in Services tab	Alerts display for these CIs	Comments
Services	CI_A, CI_B, CI_E, CI_F, CI_G, CI_H	Alerts on CI_A and CI_B appear because they are shared in Service_4.
Service_1	CI_A, CI_B, CI_E, CI_F	CI_A and CI_B are shared with Service_3 and Service_4 so they show even though Service_3 is in Maintenance Mode. CI_C and CI_D are not shared in Service_4, so alerts on those CIs are hidden.
Service_3 (subservice to Service_1)	All alerts on CIs are hidden	Service_3 is in Maintenance Mode, so all alerts on the CIs are hidden.
Service_4	CI_A, CI_B, CI_E, CI_F	Alerts on CI_A and CI_B appear because the CIs are shared among several services.
Service_2	CI_G, CI_H	The alerts on these CIs appear, because Service_2 is not in Maintenance Mode.
Service_3 (subservice to Service_2)	All alerts on CIs are hidden	Service_3 is in Maintenance Mode.

Rename Custom User Attributes

Alerts, CIs, and services have user-defined attributes that you can populate with custom data using any of the following methods:

- Event Management event policies with enrichment or create event actions (alert custom attributes only)
- Universal connector
- Custom connector policy

TIP

If you require more user attributes than are provided, as a best practice, you can assign multiple different values to a user attribute and use Matches Regex logic in escalation policy to filter out values and take action based on a certain type of value in the attribute.

All custom attributes are available in the User Defined Attributes section of the Information tab in the Component Detail pane. Alert custom attributes are also available for inclusion in alert table views and as criteria in escalation and alert queue policies.

If you use any of the custom attributes, you can rename them. The custom attributes then reflect the custom data to which they are assigned.

The renaming only applies to how the attributes appear in the main Operations Console windows. In configuration dialogs such as those for alert queues, alert filters, and event policies, the alert custom attributes appear using their original names, and any policies work that use those attributes. For example, you rename the alert User Attribute 1 to Location and you create an event policy that enriches the User Attribute 1 property with location information. The enrichment data correctly appears under the Location attribute. For manual customizations such as custom connector policy and the Universal connector, use the original attribute name for the information to appear correctly under the renamed attribute.

NOTE

If you add one of the User Attribute properties to the Alerts table in the Operations Console and then rename the property, it can disappear from the Alerts table when you reopen the Console. As a workaround, add the renamed property to the Alerts table again.

Follow these steps:

1. Open the following file in a text editor:
   <SOI_HOME>\SamUI\webapps\sam\WEB-INF\alarm\config\column-userDefined1-config.xml

   NOTE
   Ten files correspond to the ten user attributes, and the files are numbered sequentially as column-userDefined1-config.xml, column-userDefined2-config.xml, and so on.
2. Search for the <name> tag and update it as shown in the following sample code:

```
<?xmlversion="1.0" encoding="UTF-8"?>
<column id="column-userDefined1-config"
     xmlns ="http://www.aprisma.com"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="http://www.aprisma.com
               ../../common/schema/column-config.xsd">

  <name>NewNameHere</name>

  <content>
    <attribute>AlarmAttrID.USERATTRIBUTE1</attribute>
  </content>

</column>
```

3. Save the file.
4. Restart the Operations Console.
   The new name becomes visible.

Set CI User Attributes

You can set up to ten user attributes for configuration items (CIs) either from the Operations Console or in the CA SOI connector policies. You can then use these attributes as criteria for escalation, alert queue, and filter policies for managed and unmanaged alerts.

Operations Console

On the Operations Console, you view and set the CI user attributes in the Information tab of the Component Detail pane by expanding the new "User Defined Attributes" section.

The new attributes are of type String with a maximum length of 256 characters.

Connector Policy

In connector policy files, you set the CI attributes using attribute names "CIuserAttribute*X*", where *X* represents a number from 1-10. For example, "CIuserAttribute1" populates what is shown as "CI User Attribute (1)" in the Operations Console.

Example

In the DO NOT USE section of an event class in a policy file, add the following text to set "CIuserAttribute5" to "This is a Spectrum item":

```
<Field output="CIuserAttribute5"
    format="This is a Spectrum item" input="" />
```

Some connector policies write (publish) a specific attribute list. Other connector policies use a wildcard to publish all attributes that are set in the policy. If an explicit list is used, add the new attribute to the properties list in the <Write> section of the event class:

```
<Field type="publishcache"
```

```
    properties="ClassName,...,CIuserAttribute5" />
```

You can set the Universal connector user attributes in the input XML:

```
<property tag="CIuserAttribute1"
    value="secondary cluster node" />
```

You can use the new CI user attributes as runtime tokens in alert escalation actions. The right-click menu in the Escalation Action Editor now includes the following attributes with labels:

$[CI User Attribute *X*]

The labels for the alert user attributes are now changed to "$[Alert User Attribute *X*].

Enable Pending Clear State for Alerts

The Pending Clear state lets you operate on or investigate an alert after it clears. The pending cleared alerts remain in the CA SOI interfaces with the Pending Clear flag set to true; they do not impact services and are marked as exempt. You can modify a global setting (from the CA SOI Administration UI) to configure SOI into Pending Clear mode. When CA SOI is configured in Pending Cleared mode, **all** cleared alerts will be placed into a 'pending clear' state.

Post processing on cleared alerts may include the following actions:

- You are aware of the alert and want to exclude it from service impact processing
- You want to manage cleared alerts as if they are active and have them actioned, acknowledged, annotated, and manually cleared
- You want to define specific alert queues for cleared alerts for further investigation
- You want to create an escalation policy to clear the alert after it is acknowledged
- You want to double check that the condition described by the cleared alert is truly resolved before fully clearing the alert

When Pending Clear is enabled, a second clear operation (from CA SOI) clears the alert, removes it from the CA SOI interfaces, and moves it to the Cleared Alerts table. A second clear of the alert from a connector will be ignored for alerts in Pending Clear (refer to the table below). You can create an escalation policy to perform this second clear operation after a certain amount of time.

Follow these steps:

1. Access the CA SOI Dashboard and click the Administration tab.
2. Expand CA Service Operations Insight Manager Configuration and the SA Manager server folder, and select Global Settings.
3. Set Enable Pending Cleared Alerts to Yes.
4. Restart the CA SAM Application Server service.
5. Restart the CA SAM User Interface Server service.

To better visualize the Pending Clear state, you can add a new Pending Clear column to the Operations Console. You can also use the Pending Clear state as an alert queue attribute, an escalation policy attributes, for filters, and so on.

The following table illustrates how the alert life cycle workflow changes when you enable the Pending Clear state:

NOTE

: If you also enable the 'Respect Underlying MDR Clear Alert Setting' option, this option prevents any alert that is not clearable in the domain manager from entering the Pending Clear state. Only after the alert is cleared in the domain manager can the alert proceed through the workflow in the table from Pending Clear to fully cleared.

Alert State in CA SOI	Alert Action	Result	Notes
Open	Cleared in domain manager	State changes to Pending Clear	The alert is not synchronized to the domain manager and any associated ticket is not closed.
Open	Cleared in CA SOI	State changes to Pending Clear	The alert is not synchronized to the domain manager and any associated ticket is not closed. The clear operation in CA SOI can come from any CA SOI interface.
Open	Associated ticket is closed	State changes to Pending Clear	The alert is not synchronized to the domain manager.
Open	Updated in domain manager	Updates occur in CA SOI	
Pending Clear	Cleared in domain manager	State does not change in CA SOI	Pending clear alerts can only be cleared by actions in CA SOI.
Pending Clear	Cleared in CA SOI	Alert is cleared	The alert is synchronized to the domain manager and (if applicable) the ticket is closed in the help desk product. The clear operation in CA SOI can come from any CA SOI interface.
Pending Clear	Associated ticket is closed	Alert is closed	If alert synchronization is enabled, the alert closure is sent to the domain manager.
Pending Clear	Updated in domain manager (Severity does not change)	Updates occur in CA SOI	Alert is not re-opened.
Pending Clear	Severity increases	Alert is re-opened	If configured to re-open alerts when the severity of the alert increases, the alert appears as an open alert in CA SOI by having the Pending Clear and isExempt settings set to no.
Pending Clear	Severity changes	Alert is re-opened	If configured to re-open alerts when the severity of the alert changes (upward or downward), the alert appears as an open alert in CA SOI by having the Pending Clear and isExempt settings set to no.
Pending Clear	Update with alertUserAttribute set to UNPEND	Alert is re-opened	If configured to re-open alerts based on the string UNPEND appearing in a defined alertUserAttribute, the alert appears as an open alert in CA SOI by having the Pending Clear and isExempt settings set to no.

NOTE

For any scenarios involving alert synchronization, the alert synchronization feature would have to be configured. For any scenarios involving ticket closure, bi-directional ticket updates would have to be configured in order to have tickets synchronized between CA SOI and the help desk product.

Configure Pending Clear Alerts to Reopen after Updates

After an alert attains the Pending Clear state, updates to the alert may require the alert to be reopened and reevaluated. For example, a product that monitors CPU usage could send a minor alert that CA SOI sets to Pending Clear. However, if the CPU utilization grows and the alert severity increases to critical, you may want the alert to reopen.

You can configure this setting to customize when an Pending Clear alert reopens based on severity increase or the passing of an UNPEND value through a custom user attribute.

The severity-based trigger simply reopens the alert if the severity increases.

The user attribute value lets you programmatically configure any criteria for reopening a pending clear alert based on whatever triggers the UNPEND value to be assigned to a specific user attribute. You can update a connector's policy or create Event Management policy so that the UNPEND value is added to a user attribute at the appropriate time. For example, you could update the CA Spectrum connector policy to set User Attribute 1 to UNPEND when an alert attains a severity of Fatal, so that any time this severity occurs, the alert reopens if it is in the pending clear state. This example shows how you can exhibit more granular control over reopening pending clear alerts than simply reopening after any severity increase.

NOTE

Triggering a reopen based on user attribute values does not take connector restarts into account. For example, if the alert reopens due to an UNPEND value and an operator triages the alert and moves it back to pending clear, a connector restart will again pick up the UNPEND value and reopen the alert.

Follow these steps:

1. Open the SOI_HOME\tomcat\custom\UnpendPropsConfig.xml file on the SA Manager.
2. Set the PendingClearModeUpdatePolicy enabled property to true:
   ```
   <PendingClearModeUpdatePolicy enabled="true">
   ```
3. Do any of the following based on what you want to trigger a reopen of a pending clear alert:
   – Change the setting in the Severity change tag to true:
     ```
     <Severity change="Increases">true</Severity>
     ```
     This setting reopens an alert in the pending clear state if the severity increases from its original value from when it entered the pending clear state. You can also set the change attribute to All to reopen the alert when any severity change occurs.
   – Change the setting in the AlertUserAttribute number tag to true:
     ```
     <AlertUserAttribute number="1">true</AlertUserAttribute>
     ```
     This setting reopens an alert in the pending clear state if the specified user attribute attains a value of UNPEND. You can specify any user attribute to be evaluated for this value by changing the 'number' value in the syntax.
4. Save and close the file.
5. Restart the CA SAM Application Server service to apply the changes.

Configure the Assigned Property to Clear when a Pending Clear Alert Reopens

You can configure the Assigned field set in CA SOI to clear when an alert transitions from Pending Clear back into an opened state.

1. Locate and edit the SOI_HOME\tomcat\custom\UnpendPropsConfig.xml file on the SA Manager machine:
2. Add the following XML section inside the PendingClearModeUpdatePolicy XML tags:

```
<WhenUnPended>
<ClearAssigned>true</ClearAssigned>
</WhenUnPended>
```

3. Restart the CA SAM Application Server service.

Enable Delta Processing for CI User Attributes

You can configure CA SOI delta processing to factor in user attribute changes when discerning whether an object has changed since the last connector or CA SOI restart.

1. Open the relevant connector configuration file located at SOI_HOME\resources\Configurations (IFW-based connector) or CATALYST_HOME\CatalystConnector\registry\topology\physical\<nodename>\modules\configuration (CA Catalyst Container-based connector) on the connector server, and add the CI user attributes to the DeltaProperties attribute as follows:
   – For CA Catalyst Container-based connectors, add the following line into the <ModuleProperties> element:

```
<property name="DeltaProps"
 value="CIuserAttribute1,CIuserAttribute2,CIuserAttribute3,CIuserAttribute4,CIuserAttribute5,CIuserAttribute6,CIu
>
```

```
<ModuleProperties>
    <property name="UILabel" value="Unified Infrastructure Management(UIM) connector" />
    <!-- Connectors can enable/disable delta processing by setting DisableDelta property to false/true (default is false). -->
    <property name="DisableDelta" value="true" />
    <!-- This connector exposes a USM type, so we need to load the USM types -->
    <property name="LoadUSMTypes" value="false" />
    <property name="DisableUSMValidation" value="true"/>
    <property name="performDeltaProcessing" value="1" />
    <property name="DeltaProps" value=
    "CIuserAttribute1,CIuserAttribute2,CIuserAttribute3,CIuserAttribute4,CIuserAttribute5,CIuserAttribute6,CIuserAttribute7,CIuserAttribute8,CIuserAttribute9,CIuserAttribute10"/>
</ModuleProperties>
```

   – For IFW-based connectors, add the following line after ConnectorControls:

```
<DeltaProps
 properties="CIuserAttribute1,CIuserAttribute2,CIuserAttribute3,CIuserAttribute4,CIuserAttribute5,CIuserAttribute
>
```

```
<ConnectorControls dns_resolution="1"
                   getCIsAtStartUp="0"
                   getRelationshipsAtStartup="0"
                   getServicesAtStartUp="0"
                   isRemotable="0"
                   performDeltaProcessing="0"
                   useAlertFilter="0"
                   useEventStore="1"/>

<DeltaProps properties=
"CIuserAttribute1,CIuserAttribute2,CIuserAttribute3,CIuserAttribute4,CIuserAttribute5,CIuserAttribute6,CIuserAttribute7,CIuserAttribute8,CIuserAttribute9,CIuserAttribute10"/>
```

2. Restart the CA SAM Integration Services service (IFW-based connectors) or the CA Catalyst Container service (CA Catalyst Container-based connectors).
   The connector will now factor in CI user attributes changes during startup.
3. Repeat Steps 1-2 for other connectors if needed.

How to Create Escalation Policy

As an administrator, you define escalation policies and actions to automate the alert escalation process.

Alert escalation is the process of performing some action that facilitates the resolution of the alert condition. Alert escalation policy automates alert escalation according to user-defined criteria. When the policy criteria is met, a specified escalation action runs.

Escalation policy is based on any or all of the following criteria:

- **Alert type**
  Specifies to act on all alerts of a specific type that meet the defined criteria. For example, an escalation policy can apply to service alerts only, or a subset of infrastructure alerts, such as root cause alerts.

Each escalation policy must specify the alert types to include. You can create escalation policy with only alert types and no criteria to escalate all alerts that meet the type requirement (for example, to escalate all root cause alerts).

- **Time-based criteria**
  Specifies to act on alerts according to time-based thresholds, such as the alert age or time in an alert queue. For example, you can specify to act on any alert that has not been assigned within 10 minutes.
- **Attribute-based criteria**
  Specifies to act on alerts with attributes that meet specified criteria.

The following types of escalation policies are available:

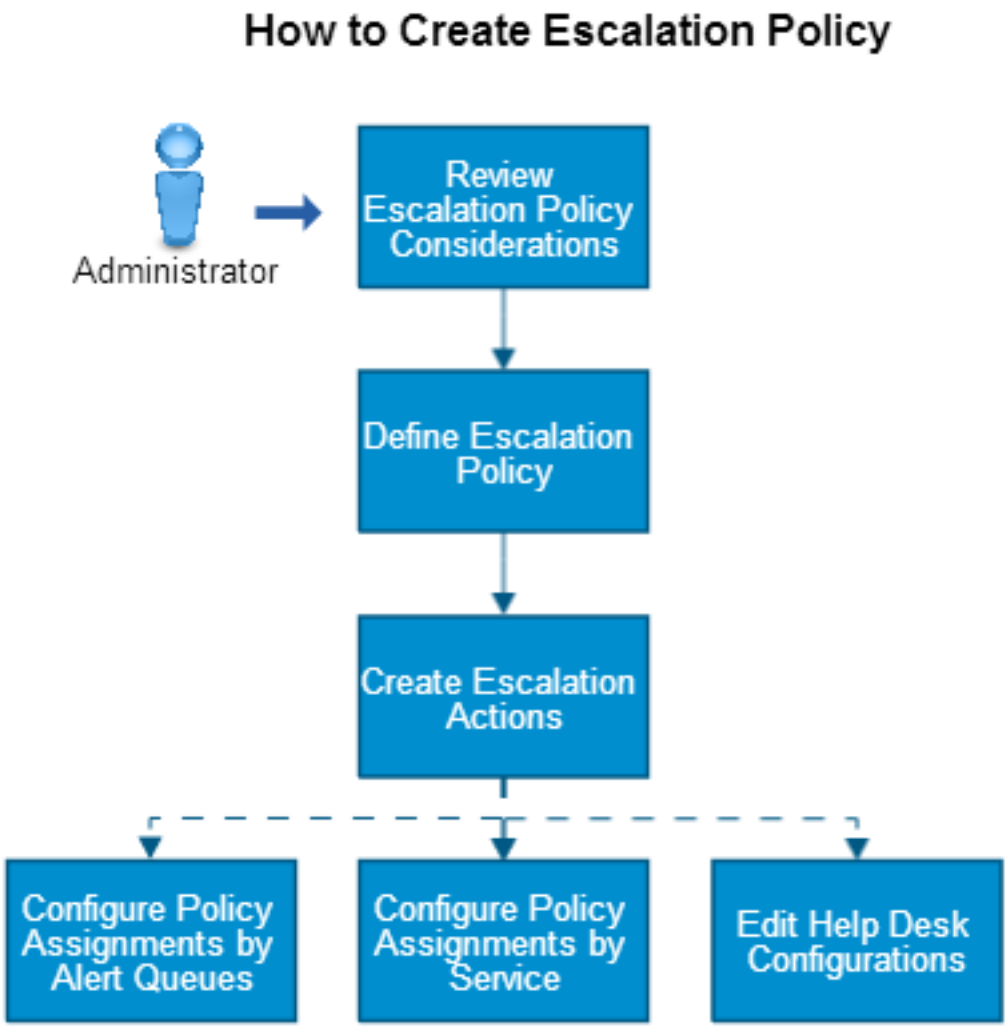
- **Non-Global (service or alert queue specific)**
  Escalates alerts in one or more specified services or alert queues that meet the policy criteria. When an alert is being considered for escalation, it is evaluated in the following policy order: non-global (service then alert queue) then global. For example, you can create service-specific escalation policy for a payroll service owner who wants a notification when CA SOI raises an alert against the payroll service. You can also create a policy that sends an email when critical alerts have not been cleared in an alert queue for a specified time period.
- **Global**
  Escalates all alerts that meet the policy criteria. For example, you create a global escalation policy for an IT manager who wants notification when CA SOI raises *any* service alert.

Escalation policy results in one of the following actions:

- Run a command
- Send a notification by email to a technician or operator
- Open a help desk ticket
- Open a help desk announcement
- Execute a CA Process Automation process
- Clear an alert

Use this scenario to guide you through the process:

Figure 37: How to Create Escalation Policy



1. Review the escalation policy considerations.
2. Define the escalation policy.
3. Create escalation actions.
4. (Optional) Configure policy assignments by alert queues.
5. (Optional) Configure policy assignments by service.
6. (Optional) Edit help desk configurations.

Escalation Policy Considerations

Consider the following situations before creating escalation policy:

- Before you implement escalation policy, verify the configuration for email and help desk applications. Configure email for both the SA Manager and UI Server.
- An alert escalates once for each policy, not each time the severity changes.
- You can define escalation policy with no criteria beyond a specific alert type. In this case, each alert of the specified type is escalated.
- If you create time-based and attribute-based criteria for a policy, the criteria always have a boolean AND relationship.

Define Escalation Policy

Contents

As an administrator, you create escalation policy in the Operations Console as follows:

- From the Escalation Policies and Actions dialog
- When creating or editing a service in the Service Modeler
- When creating or editing an alert queue

The procedures in this section describe how to create escalation policy from the Operations Console. You can also create global escalation policy from the Alert Escalation tab of the Service Modeler. For more information about using this tab to create policy and assign it to a service, see Configure Policy Assignment by Service.

Follow these steps:

1. Start the Operations Console and select Tools, Escalation Policies and Actions.
   The Escalation Policies and Actions dialog opens.
2. Click

   .

   The Alert Escalation Policy Editor dialog opens.
3. Perform the following actions in the upper pane of the Policy Definition tab:
   a. Enter a policy name and optional description in the Name and Description fields.
   b. Select Enabled or Disabled to control whether the policy starts evaluating alerts immediately after you create it. If you select Disabled, you manually enable the policy when you want it to take effect.
   c. Select Global or Non-Global in the Policy Type section:
      - **Global**
        Specifies that the policy applies to all alerts.
      - **Non-Global**
        Specifies that the policy applies to alerts of an assigned service or alert queue.
4. Configure the alert types to escalate as follows on the Alert Selection subtab:
   – **Service Alerts**
     Escalates all service alerts that meet the policy criteria.
   – **Infrastructure Alerts**
     Escalates infrastructure alerts that meet the policy criteria. Select either of the following options to escalate a subset of infrastructure alerts only:

     NOTE
     Only select one of these options if you want to limit infrastructure alert escalation to a specific subtype. Selecting Infrastructure Alerts with no subtype or both subtype selections escalates all infrastructure alerts, regardless of type.

     - **Root cause infrastructure alerts**
       Escalates infrastructure alerts that are assigned as a root cause to a service alert.
     - **Symptom infrastructure alerts**

Escalates infrastructure alerts that are not a root cause of a service alert, but has been determined to be a symptom based on either a root cause policy or the domain manager (if in Domain Manager mode).

These alert subtypes are calculated based on the escalation policy type. A service-specific policy evaluates alert subtypes against the assigned service. A global policy evaluates alert subtypes that are based on any associated service. For example, an alert belongs to a CI that is associated with five services. The alert is a root cause for one service, a global policy considers it a root cause alert. However, a service-specific policy only considers the alert a root cause alert if it is a root cause for the assigned service.

As a best practice, only use the subtype selections with time-based escalation policy. Alert states can change after initial alert generation, which could lead to false escalations if the policy is not time-based.

– **Alerts for CIs in maintenance mode**
  Escalates infrastructure and service alerts for CIs that are currently in maintenance. If you clear this check box, the policy is only applied after the alert's CI is out of maintenance mode.
– **Infrastructure alerts for Services in maintenance mode**
  (Non-Global policy only) Escalates any type of infrastructure alert if the parent service is in maintenance. If you clear this check box, the policy is only applied after the service is out of maintenance mode. If you select the check box and the policy is assigned to multiple services, and one of the services is not in maintenance mode the policy still triggers.

Alert type settings are configured.

5. Click the Time subtab.
   Select whether any or all selected criteria must be met, select the criteria to apply, and enter the time period on which to base the criteria for each selection.
   All selections are based on the number of minutes elapsed.
6. Click Attributes.
   Select an alert attribute on which to filter, a comparison type, and a comparison value for the attribute, and click Add.
   Consider the following information:
   For definitions of available attributes, see the topic that follows.
   – To use regular expressions, select Matches regex from the Comparison Type drop-down list. Click Test Regex to open the Regex Tester and test the regular expression against a string. Regular expressions are not available for all attributes.

   You can define criteria that are based on the following items:
   – Alert properties
   – The correlatable USM properties of the associated CI
   – The associated customer name, customer impact, custom identity, and customer priority

   The attribute expression appears in the lower pane.
7. (Optional) Add more attribute criteria and create advanced logic using the logic buttons on the right of the dialog.
   For more information about creating advanced attribute criteria, click the Hints link above the expression pane.
8. (Optional) Click Escalation Schedule to apply the policy during a specific time period.
   The Escalation Schedule subtab opens.
9. Leave the default 24x7 selection if you want the policy to be in effect always, or select one of the following options:
   – **Escalate only on service's business hours**
     (Non-Global policy only) Escalates alerts only during the business hours schedule that is defined for the assigned service. If the assigned service has no business hours schedule, CA SOI always escalates alerts.
   – **Escalate only on the following selected business hours**
     Escalates alerts only during a business hours schedule that you select.
10. (Optional) If you selected 'Escalate only on the following selected business hours' perform one of the following actions:
    – Select an available schedule, if any, and click

      .

      The schedule appears in the Current Schedules pane.
    – Click Create.

The Create Business Hours dialog opens.

11. (New business hours only) Enter the required information in the Create Business Hours dialog, and click OK:

    NOTE
    If you enter multiple business hours schedules, the product verifies that the hours do not overlap.

12. Click the Policy Actions tab.
    The Policy Actions tab opens with a list of available escalation actions.
13. (Optional) Select the Perform escalation action after a minute(s) delay option and enter the number of minutes. For more information about this feature, click the Hints link.

    NOTE
    If you restart the CA SAM Application Server service, the timer restarts also.

14. Perform one the following actions:
    – Move actions to apply to the policy from the Available Actions pane to the Actions to Perform pane.
    – Create an escalation policy action.

    You can add multiple actions to the policy.
15. (Non-Global policy only) Select the Service Assignment tab.
    Move the services to which you want to assign the policy to the Assigned Services for this Policy pane and click OK.

    NOTE
    Alerts on subservice CIs do not trigger a parent service's escalation policy. If you want the subservice CI alerts to trigger the policy for the parent, assign each subservice to the escalation policy.

    You can also assign service-specific policy in the Service Modeler. For more information, see Configure Policy Assignment by Service.
16. (Non-Global policy only) Select the Alert Queue Assignment tab.
    Move the alert queues to which you want to assign the policy to the Assigned Alert Queues for this Policy pane and click OK.
17. (Optional) Return to the Policy Definition tab and click Show Policy Summary.
    A text box appears at the bottom of the dialog that describes the escalation policy in a brief expression. Use this description to verify that you defined the policy correctly.
18. Click OK.
    The escalation policy is defined and appears in the Escalation Policies and Actions dialog.

Escalation Policy Attributes

You can define escalation policy using the following attributes:

USM Properties

Any attribute not listed in the categories that follow are USM types and properties. For USM definitions, see How to Access the USM Schema Documentation.

Alert Properties

- **Acknowledged**
  Indicates that the alert is acknowledged.
  Value: Yes or No
- **Assigned**
  Indicates the name of the operator that is assigned to the alert.
  Value: String
- **Business Priority**
  Indicates the service priority.
  Value: Unspecified, None, Medium, Low, High, or Critical
- **Category**

Indicates whether this alert condition affects the quality or risk of the services it impacts.
Value: String

- **Is Clearable**
  Defines whether the alert can be cleared in the domain manager. If you enabled the 'Respect Underlying MDR Clear Alert Setting', this property also determines whether the alert can be cleared in CA SOI.
- **Message**
  Indicates a message entered by the operator.
  Value: String
- **Non-Service Impacting Alert**
  Indicates that the alert does not impact a modeled service.
  Value: Yes or No
- **Root Cause**
  Indicates if the alert is the root cause.
  Value: Yes or No
- **Service Impact**
  Indicates the impact, which is calculated by multiplying alert severity and the significance of the CI to the service. When multiple services are impacted, the most affected service is displayed.
  Value: Down, Moderate, None Slight, or Severe
- **Service Name**
  Indicates a service name associated with an alert.
  Value: String
- **Severity**
  Indicates the alert severity that the originating domain manager assigned.
  Value: Critical, Down, Major, Minor, Normal, or Unknown
- **Source**
  Indicates the domain manager where the alert originated. The format is MdrProduct_domainserver@connectorserver. For example, CA:00005_spectrohost.ca.com@spectrohost.ca.com refers to a CA Spectrum connector installed on spectrohost.ca.com monitoring a CA Spectrum instance that is installed on the same system.
  Value: String
- **Symptom**
  Indicates that the root cause is a symptom.
  Value: Yes or No
- **Ticket ID**
  Indicates the associated help desk ticket number.
  Value: String
- **Unclassified**
  Indicates the root cause in unclassified.
  Value: Yes or No
- **Unmanaged**
  Indicates if the alarm is associated with a model CI or not.
  Value: Yes or No

CI User Attributes

An administrator sets the CI user attributes and the attributes are labeled 1-10. CI user attributes let you define custom CI user attributes that are not provided on the attribute list through the USM schema. For more information about setting CI user attributes, see Set CI User Attributes.

Customer Properties

- **# Impacted Customers**
  Indicates the number of customers that the alert impacts. This number is based on the number of customers that are assigned to the service that an alert impacts.

Value: Number

- **# Impacted Services**
  Indicates the number of services the alert impacts, which is based on the number of services its associated CI is included in.
  Value: Number
- **Customer ID**
  Indicates the customer identification number.
  Value: String
- **Customer Impact**
  Indicates the alert impact to the customer.
  Value: Down, Moderate, None, Severe, or Slight
- **Customer Name**
  Indicates the customer name.
  Value: String
- **Customer Priority**
  Indicates the customer priority set by the administrator.
  Value: 1-10 or the values configured by the administrator.
- **Highest Customer Impact**
  Indicates the highest impact that the alerts caused for an associated customer.
  Value: Down, Moderate, None, Severe, or Slight
- **Highest Customer Priority**
  Indicates the highest customer priority number.
  Values: 1-10 or the values configured by the administrator.

Create Escalation Actions

Contents

As an administrator, you configure escalation policy to perform any of the following automated actions:

- Notification to a technician or operator by email
- Open a help desk ticket
- Update a help desk ticket
- Run a command
- Open an announcement
- Run a CA Process Automation process
- Clear an alert

You can define actions that are tailored to your enterprise. You then assign the actions to policies. For example, you can define an action that sends notification messages to specific recipients. You can define actions during escalation policy definition or in a separate operation. After actions are defined, they are available to any escalation policy that you later define or edit. The escalation actions are also available to users on the Mobile Dashboard.

To manually apply an escalation action, you can right-click an alert and select Take Action.

WARNING
Before you create escalation policy actions, consider the following items:

- Verify the configuration for email, help desk applications, and CA Process Automation Server configuration.
- You can configure email notifications so that CA SOI notifies you when an escalation action fails. For more information about failure email notifications, see Configure Email and Failure Notifications.

Expandable Runtime Tokens

CA SOI substitutes values for expandable runtime tokens to provide specific values in escalation action policies (help desk tickets, announcements, and emails). All CA SOI alert properties and USM alert properties have expandable runtime tokens available. The following tokens are examples of the tokens that are available:

- alert acknowledgement status
- alert cleared date
- alert queue name
- alert queue priority
- customer name
- customer priority
- maintenance flag status
- service impact
- URL to a helpdesk ticket

You can use expandable runtime tokens in the following situations:

- When you assign values to a property that you are adding to a Create Ticket or Create Announcement action
- When you populate the Message, Subject, Recipients, and From fields in a Send Email action
- When you populate the Command field in an Run a Command action
- When you add values to parameters in an Execute Automated Process action

For example, you can use the value for Assignee as set in the Alert Details tab in the Component Detail pane. In a Send Email escalation action, select $[Assigned] in the Property Value column, or right-click the Message field in the Escalation Action Editor and select $[Assigned].

To display a list of available tokens, perform *one* of the following actions:

- Right-click a field in a Send Email or Execute Command action and select More.
- Expand the Property Value drop-down list in a Create Ticket or Create Announcement action.
- Click More in the Available property values list of the Property Editor dialog.

The Select Runtime Token dialog opens when you click More and list every available token and the USM types to which it belongs.

Although the purpose of most tokens is obvious by its name, the following tokens require an additional explanation:

- **$[RCAGroupCustomPolicyAlerts]:**Lists the alerts and CIs that triggers the custom policy. If you need the list of CIs that violates the custom policy when you trigger any action on a policy alarm, use the $[RCAGroupCustomPolicyAlerts] runtime token when you define an action. The user receives the list of offending CI's along with the alerts in the following order:
  Alert Severity | Alert Type | Date or Time stamp of Alert creation | Class of CI | IP Address | Hostname | Device ID| Alert Summary| Alert Detail.

  NOTE

  - IP Address displays Primary IP V4 address or Device IP V4 address. Hostname displays Primary DNS name or Device DNS name.
  - If value is not available for any property, the corresponding property will be blank.
  - Alert Severity, Alert type, Date or Time stamp of Alert creation, Alert Summary, and Alert Detail properties are not applicable for CIs. These fields will be blank
- **$[Cleared Date]**
  Specifies the date an Operator cleared an alert.
- **$[Login User]/[$Login Host]**
  An administrator can use the $[Login User] token in defining Take Action policies. This token identifies the user that is logged into the Operations Console.Use this token to identify an operator opening a ticket, sending an announcement,

or sending an email. This token is useful for environments that do not use automatic escalation policy and operators open tickets manually.
If you use this token in an automatic escalation policy, the token cannot identify a user name and the token displays as Not Set.
Similar to $[Login User], you can use the $[Login Host] token to identify the name of the host on which the Operations Console was opened.

- **$[Mobile UI URL]**
  Displays as a URL link in emails and tickets. The user clicks the URL to launch the Mobile Dashboard and display the CI associated with the alert. You must enable use of this token.
- **$[USM Web View URL]**
  The token lets a user click the URL to launch USM Web View and display the CI that is associated with the alert. You must enable use of this token.

Enable Use of $ Mobile UI URL Runtime Token

The $[Mobile UI URL Runtime Token] displays as a URL link in emails and tickets. The user clicks the URL to launch the Mobile Dashboard and display the CI associated with the alert.

Before you can use this runtime token, you configure the server and port number of the Mobile Dashboard server so the URL link launches correctly.

Follow these steps:

1. Access the Dashboard, and click the Administration tab.
2. Expand CA Service Operations Insight Manager Configuration and the SA Manager system, then click Mobile Dashboard Server Configuration.
3. Enter the host name and port of the server that hosts Mobile Dashboard (the UI Server by default), and click Save.

Enable Use of $ USM Web View URL Runtime Token

You can embed the $[USM Web View URL] runtime token in an action (such as an email or ticket). The token lets a user click the URL to launch USM Web View and display the CI that is associated with the alert.

Before you can use this runtime token, you configure the server and port number of the USM Web View server. The configuration makes the URL launch correctly.

Follow these steps:

1. Access the Dashboard, and click the Administration tab.
2. Expand CA Service Operations Insight Manager Configuration and the SA Manager system, then click Mobile Dashboard Server Configuration.
   Enter the host name and port of the CA Catalyst Server, which hosts USM Web View, and click Save.

Create a Send Email Action

You can create a policy action that automatically sends an email to specified recipients (usually a technician or an operator).

Follow these steps:

1. Perform one of the following actions:
   - Click New on the Actions tab of the Alert Escalation Policy Editor dialog while defining an escalation policy.
   - Select Tools, Escalation Policies and Actions, and click

on the Actions tab of the Escalation Policies and Actions dialog to create an action independent of an escalation policy.

2. Select Send Email from the Action Type drop-down list.
   The email fields appear.
3. Complete the following fields:
   – Enter an action name and optional description in the Action Name and Description fields.
   – Specify a recipient in the Recipients field using the format *recipient@anycompany.com*. Separate multiple recipients with commas.
   – Enter an email address or sender name to display in the From field. By default, the product uses the user name of the policy creator and the SA Manager server host name in the format *user@server*.
   – Enter an email subject in the Subject field.
   – Enter an email message in the Message field.

   Consider the following items:

   - You can right-click in most text fields to select expandable runtime tokens for alert details. These tokens are dynamically substituted when an action is performed on an alert.
   - If you want to use the $[USM Web View URL] token, you first enable its use.
   - If you want to use the $[Mobile UI URL] token, you first enable its use.
   - If you want to use the CI user attribute tokens, you first set the attributes. For more information about setting CI user attributes, see Set CI User Attributes.
4. (Optional) If you want the action available now, select the Enabled option.
5. Click OK.
   The action is defined, and it appears on the Actions tab. If you defined the action in an escalation policy, it is automatically added to the policy.

Create a Ticket Action

You can create a policy action that automatically opens a ticket in your integrated help desk product.

Consider the following situations:

- If more than one ticket action exists to create a ticket under the same conditions, CA SOI creates only one ticket. CA SOI then updates that ticket for subsequent actions.
- CA SOI can automatically close any ticket that is opened after an alert has cleared. To enable this option, see How to Work with Configured Help Desk Integrations.

Follow these steps:

1. Perform one of the following actions:
   - Click New on the Actions tab of the Alert Escalation Policy Editor dialog while defining an escalation policy.
   - Select Tools, Escalation Policies and Actions, and click

     on the Actions tab of the Escalation Policies and Actions dialog to create an action independent of an escalation policy.
2. Select Create Ticket from the Action Type drop-down list.
   The ticket action fields appear.
3. Complete the following fields:
   – Enter an action name in the Action Name field.
   – (Optional) Enter a ticket description in the Description field.
4. (Optional) Click Add Exception Criteria to add rule-based properties. For more information, see Add Exception Criteria.
5. Click the Properties tab.

6. Select a Property Name from the drop-down list.
   The Property Value field requires you to either select an item from a drop-down list, enter an item manually, or complete a text field. The following properties are provided:

   Consider the following items:

   - You can right-click in most text fields to select expandable runtime tokens for alert details. These tokens are dynamically substituted when an action is performed on an alert.
   - If you want to use the $[USM Web View URL] token, you first enable its use.
   - If you want to use the $[Mobile UI URL] token, you first enable its use.
   - If you want to use the CI user attribute tokens, you first set the attributes. For more information about setting CI user attributes, see Set CI User Attributes.

   – If you are integrating with BMC Remedy, HP Service Manager, or Universal Help Desk API, you map the help desk properties to corresponding properties. For more information, see Help Desk Integrations.
   – **Ticket type**
     Specifies the help desk ticket type, which is typically: Incident, Problem, or Request.
   – **Description**
     Specifies a general ticket description.
   – **Summary**
     Specifies a general ticket summary.
   – **Affected End User**
     Specifies the customer name. The Property Value object is in the following format:
     `lastname,firstname`
   – **Assignee**
     Specifies the name that the ticket is assigned to. The Property Value object is in the following format:
     `lastname,firstname`
   – **Template**
     Specifies the template name that must exist in the help desk.
   – **Configuration Item**
     Specifies an identifier for the associated CI that you can look up in the help desk product.
   – **Request area**
     Specifies the request area name that must exist in the help desk.
   – **Asset**
     Specifies the asset value that determines how the asset is queried in the help desk.
     Example: A value of $[DNS Name] performs a look up in the help desk using the DNS name of the CI. Other available values are $[Host Name] and $[IP Address].
   – **Group**
     Specifies the help desk group name.
   – **Severity**
     Specifies the ticket severity.
   – **Impact**
     Specifies the ticket impact.
   – **Priority**
     Specifies the ticket priority.
   – **Root Cause**
     Specifies the ticket root cause.

   You can also add custom help desk ticket properties from the Help Desk Configuration dialog.
7. (Optional if available) Select the 'Create Object if not present' check box to create the object in the help desk if it does not currently exist.
8. Click Add.

The new property is added to the Default Properties list.
Note: You can edit the value for an existing property or delete a property from the Default Properties list.

9. (Optional) Repeat Steps 6-8 to add as many properties as necessary to the ticket.
10. (Optional) Click the Summary tab to view the current properties and values. Click the link for any exception on this tab to view exception details.
11. (Optional) Select the Enabled option if you want the action to be available now.
12. Click OK when you finish configuring the action.
    The action is defined, and it appears on the Actions tab. If you defined the action in an escalation policy, it is automatically added to the policy.

View a Help Desk Ticket

After an alert is associated with a help desk ticket, you can link to the help desk application to review ticket details. Any valid help desk user can view a ticket.

Follow these steps:

1. Click the alert whose ticket you want to view.
   Details about the alert appear on the Alert Details tab in the Component Detail pane.
2. Click the ticket number in the Ticket ID field.
   If the field is empty, no ticket exists associated with the alert.
   A dialog for the user name and password opens.
3. Enter the requested information, and click OK.
   The help desk application starts and displays details about the ticket.

> **NOTE**
> You can configure the help desk to go directly to the ticket detail without prompting for a user name and password first. For more information, see Enable Automatic Links to Tickets.

Create a Ticket Update Action

You can create a policy action that automatically updates a ticket in your integrated help desk product.

> **NOTE**
> You cannot update the ticket type or the template.

Follow these steps:

1. Perform one of the following actions:
   - Click New on the Actions tab of the Alert Escalation Policy Editor dialog while defining an escalation policy.
   - Select Tools, Escalation Policies and Actions, and click

     on the Actions tab of the Escalation Policies and Actions dialog to create an action independent of an escalation policy.
2. Select Update Ticket from the Action Type drop-down list.
   The ticket action fields appear.
3. Complete the following fields:
   – Enter an action name in the Action Name field.
   – (Optional) Enter a ticket description in the Description field.
4. (Optional) Click Add Exception Criteria to add rule-based properties. For more information, see Add Exception Criteria.
5. Click the Properties tab.
6. Select a Property Name from the drop-down list.

7. Select a Property Value from the drop-down list.
8. (Optional if available) Select the 'Create Object if not present' check box to create the object in the help desk if it does not currently exist.
9. Click Add.
   The new property is added to the Default Properties list.
10. (Optional) Repeat Steps 6-9 to add as many properties as necessary to the ticket.
11. (Optional) Click the Summary tab to view the current properties and values. Click the link for any exception on this tab to view exception details.
12. (Optional) Select the Enabled option if you want the action to be available now.
13. Click OK when you finish configuring the action.
    The action is defined, and it appears on the Actions tab. If you defined the action in an escalation policy, it is automatically added to the policy.

Run a Command Action

You can create a policy action that automatically runs a command. The command must meet the following criteria:

- The command can run on the SA Manager server.
- The command returns a result that requires no user input other than that entered in the command. For example, a command that launches a user interface does not meet this criteria.

Follow these steps:

1. Perform one of the following actions:
   - Click New on the Actions tab of the Alert Escalation Policy Editor dialog while defining an escalation policy.
   - Select Tools, Escalation Policies and Actions, and click

     on the Actions tab of the Escalation Policies and Actions dialog to create an action independent of an escalation policy.
2. Select Execute Command from the Action Type drop-down list and complete the following fields:
   – Enter an action name in the Action Name field.
   – (Optional) Enter an action description in the Description field.
   – Enter the full path of a command to run in the Command field. The command runs on the SA Manager server.

   Consider the following items:

   - You can right-click in most text fields to select expandable runtime tokens for alert details. These tokens are dynamically substituted when an action is performed on an alert.
   - If you want to use the $[USM Web View URL] token, you first enable its use.
   - If you want to use the $[Mobile UI URL] token, you first enable its use.
   - If you want to use the CI user attribute tokens, you first set the attributes. For more information about setting CI user attributes, see Set CI User Attributes.

   – UI commands such as notepad.exe or cmd.exe are not supported.
3. (Optional) If you want the action available now, select the Enabled option.
4. Click OK.
   The action is defined, and it appears on the Actions tab. If you defined the action in an escalation policy, it is automatically added to the policy.

Create an Announcement Action

You can create a policy action that automatically sends a help desk announcement.

Follow these steps:

1. Perform one of the following actions:
   - Click New on the Actions tab of the Alert Escalation Policy Editor dialog while defining an escalation policy.
   - Select Tools, Escalation Policies and Actions, and click

     on the Actions tab of the Escalation Policies and Actions dialog to create an action independent of an escalation policy.
2. Select Create Announcement from the Action Type drop-down list and complete the following fields:
   – Enter an action name in the Action Name field.
   – (Optional) Enter an announcement description in the Description field.
3. (Optional) Click Add Exception Criteria to add rule-based properties. For more information, see Add Exception Criteria.
4. Click the Properties tab.
5. Select a Property Name from the drop-down list.
   The Property Value requires you to either select an item from the drop-down list or enter an item or text manually.

   Consider the following items:

   - You can right-click in most text fields to select expandable runtime tokens for alert details. These tokens are dynamically substituted when an action is performed on an alert.
   - If you want to use the $[USM Web View URL] token, you first enable its use.
   - If you want to use the $[Mobile UI URL] token, you first enable its use.
   - If you want to use the CI user attribute tokens, you first set the attributes. For more information about setting CI user attributes, see Set CI User Attributes.

   – You map the help desk properties to corresponding properties in BMC Remedy, HP Service Manager, or the Universal Help Desk API if you are integrating with either of these help desk products. For more information, see Help Desk Integrations. You also set the help desk configuration for in-context links to help tickets. For more information, see Configure Help Desk Integration.
   – **Announcement Type**
     Specifies the announcement type, which is typically: Routine, Advisory, Emergency, or a custom-defined attribute.
   – **Text**
     Specifies general text.
   – **Close Date/Time**
     Specifies the date and time the announcement ends in the following format:
     ```
     DD/MM/YYYY HH:MM:SS
     ```
     HH must be an integer from 0 to 23.
   – **Active**
     Specifies if the announcement is active.

   You can also add custom announcement properties in the Help Desk Configuration dialog.
6. (Optional if available) Select the 'Create Object if not present' check box to create the object in the help desk if it does not currently exist.
7. Click Add.
   The new property is added to the Default Properties list.
8. (Optional) Repeat Steps 6-8 to add as many properties as necessary to the announcement.
9. (Optional) Click the Summary tab to view the current properties and values.
10. (Optional) If you want the action available now, select the Enable option.
11. Click OK when you finish configuring the action.
    The action is defined, and it appears on the Actions tab. If you defined the action in an escalation policy, it is automatically added to the policy.

Create a CA Process Automation Form or Process Execution Action

You can create an escalation action that automatically runs a CA Process Automation form or process when associated escalation policy criteria is met.

> **NOTE**
> CA Process Automation processes can have an associated form that lets you enter all information required for the process to run. For CA Process Automation processes with associated forms, you can select the process and populate the required parameters in the escalation action. You must manually enter the process name and required parameters and values for processes that do not have an associated form. For more information about CA Process Automation forms and processes, see the CA Process Automation documentation.

Follow these steps:

1. Perform one of the following actions:
   - Click New on the Actions tab of the Alert Escalation Policy Editor dialog while defining an escalation policy.
   - Select Tools, Escalation Policies and Actions, and click

     on the Actions tab of the Escalation Policies and Actions dialog to create an action independent of an escalation policy.
2. Select Execute Automated Process from the Action Type drop-down list.
   The CA Process Automation fields appear.
3. (Optional) Select the Enabled option if you want the action to be available now.
4. Complete the following fields:
   – Enter an action name in the Action Name field.
   – (Optional) Enter an action description in the Description field.
5. If you are using CA Process Automation with SSL, .
6. Do one of the following:
   – If you want to use an available CA Process Automation process with a form, see Use a Process with an Associated Form.
   – If you want to enter a CA Process Automation process manually, see Use a Process without an Associated Form.

Configure SSL Connection with CA Process Automation

For CA SOI to communicate with a CA Process Automation server that has been configured to use SSL, you must import a certificate into the CA SOI trust store.

Follow these steps:

1. Copy the itpamcertificate.cer file from the following location on the CA Process Automation server to a directory on your SA Manager server:
   PA_HOME\ITPAM\server\c20\.c20repository
2. Make a backup copy of the SOI_HOME\tomcat\conf\ssa.jks file.
3. Run the following command from a command prompt on the SA Manager system to import the certificate into CA SOI:

   ```
   "JAVA_HOME\bin\keytool.exe" -v -importcert -storepass password -file DIR\itpamcertificate.cer -keystore
   "SOI_HOME\tomcat\conf\ssa.jks" -trustcacerts -noprompt
   ```

   – **password**
   Defines the password for the CA SOI administrator user.
   – **DIR**
   Defines the path to the directory to which you copied the itpamcertificate.cer file.
4. Restart the CA SAM Application Server service.

5. Configure CA Process Automation integration in the Administration tab on the Dashboard. Select the SSL check box and use the SSL port number.
6. Click Test.

Use a Process with an Associated Form

When a CA Process Automation process has an associated form, its name and parameters appear on the Escalation Action Editor dialog. You can select the process and enter parameter values to use it in an escalation action.

Follow these steps:

1. Select the Use an Available Form option for the Execute Automated Process action that you are creating.
   The available forms from the integrated CA Process Automation server appear in the Select a Form pane.
   Consider the following:
   – Only CA Process Automation processes with an associated CA Process Automation form appear in the Select a Form pane.
   – If an update has been made on the CA Process Automation server, click Refresh Available Forms to update the Available Forms table.
2. Select a form.
3. Select a parameter and click Edit to edit the form parameter name, data type, or value.
   Consider the following:
   – Verify that all required parameters have values; otherwise, the process fails.
   – Password data types are encrypted in the Summary pane.
   – When you select the String data type, you can right-click in the Parameter Value field to select expandable runtime tokens for alert details. These tokens are dynamically substituted when an action is performed on an alert.
4. Repeat Step 3 for each parameter.
5. Click Update Summary to add the parameters to the Summary pane.
6. Click OK.
   The action is defined, and it appears on the Actions tab. If you defined the action while creating an escalation policy, it is automatically added to the policy.

Use a Process without an Associated Form

When a CA Process Automation process does not have an associated form, it does not appear in the Escalation Action Editor dialog. You must manually enter the process name and parameters to use it for the action.

Follow these steps:

1. Select the Execute another Process option for the Execute Automated Process action that you are creating.
2. Enter the path and process name.

   NOTE
   The process must exist in CA Process Automation.
3. Click Add to define a new process parameter.
   Consider the following:
   – Verify that all required parameters have values; otherwise, the process fails.
   – Password data types are encrypted in the Summary pane.
   – When you select the String data type, you can right-click in the Parameter Value field to select expandable runtime tokens for alert details. These tokens are dynamically substituted when an action is performed on an alert.
4. Click Update Summary to add the process parameter to the Summary pane.
5. (Optional) You can do any of the following:

- Repeat Steps 3 - 4 to define additional process parameters.
- Select a parameter and click Edit to edit the process parameter name, data type, or value.
- Select a parameter and click Delete to delete the process parameter.

6. Click OK.
   The action is defined, and it appears on the Actions tab. If you defined the action while creating an escalation policy, it is automatically added to the policy.

Create a Clear Alert Action

You can create an action that automatically clears an alert when the associated policy criteria are met.

Follow these steps:

1. Perform one of the following actions:
   - Click New on the Actions tab of the Alert Escalation Policy Editor dialog while defining an escalation policy.
   - Select Tools, Escalation Policies and Actions, and click

     on the Actions tab of the Escalation Policies and Actions dialog to create an action independent of an escalation policy.
2. Select Clear Alert from the Action Type drop-down list and complete the following fields:
   - Enter an action name.
   - (Optional) Enter an action description in the Description field.
3. (Optional) If you want the action available now, select the Enabled option.
4. Click OK.
   The action is defined, and it appears on the Actions tab. If you defined the action in an escalation policy, it is automatically added to the policy.

Add Exception Criteria

You can define rule-based assignment of properties during the ticket or announcement creation. The rules include alert and CI attributes, and if the rules are met, then the specified properties are used during the ticket creation.

Follow these steps:

1. Click Add Exception Criteria.
   Current exceptions appear in separate tabs that are sorted by the exception name. The default name for a new exception is Exception.
2. Change the default Exception name.
3. Select an attribute and comparison type.

   NOTE
   To use regular expressions, select Matches regex from the Comparison Type drop-down list. Click Test Regex to open the Regex Tester and test the regular expression against a string. Regular expressions are not available for all attributes.
4. Enter (or select if a drop-down list appears) an attribute value.
5. Click Add.
   The attribute expression appears in the lower pane.
6. (Optional) Add more attribute criteria and create advanced logic using the logic buttons on the right-hand side of the dialog.

NOTE
For more information about creating advanced attribute criteria, click the Hints link above the expression pane. For more information about the attributes, see Exception Criteria Attributes. The Properties for Exception pane lets you specify the properties that are used if the exception criteria are met.

7. You can perform the following actions:
   – Click Copy

     to add the default set properties defined in the Ticket Properties tab to the list.
   – Click Add

     to add a property.
   – Select an existing property and click Edit

   – Select an existing property and click Delete

   – Click Print

     to print the Properties for Exception list.
   – Click Export

     to export the Properties for Exception list to a data file for a spreadsheet.

After you create the exception, its tab is accessible from the main Escalation Action Editor dialog for the action and from a link on its name in the Summary tab.

Exception Criteria Attributes

You can define exception criteria using the following attributes:

USM Properties

Any attribute not listed in the categories that follow are USM types and properties. For USM definitions, see the *USM Schema Documentation.*

Alert Properties

- **Acknowledged**
  Indicates that the alert is acknowledged.
  Value: Yes or No
- **Assigned**
  Indicates the name of the operator that is assigned to the alert.
  Value: String
- **Business Priority**
  Indicates the service priority.
  Value: Unspecified, None, Medium, Low, High, or Critical
- **Category**
  Indicates whether this alert condition affects the quality or risk of the services it impacts.

Value: String

- **Message**
  Indicates a message that the administrator entered.
  Value: String
- **Non-Service Impacting Alert**
  Indicates that the alert does not impact a modeled service.
  Value: Yes or No
- **Root Cause**
  Indicates if the alert is the root cause.
  Value: Yes or No
- **Service Impact**
  Indicates the impact, which is calculated by multiplying alert severity and the significance of the CI to the service. When multiple services are impacted, the most affected service is displayed.
  Value: Down, Moderate, None Slight, or Severe
- **Service Name**
  Indicates a service name that is associated with an alert.
  Value: String
- **Severity**
  Indicates the alert severity that the originating domain manager assigned.
  Value: Critical, Down, Major, Minor, Normal, or Unknown
- **Source**
  Indicates the domain manager where the alert originated. The format is MdrProduct_domainserver@connectorserver. For example, CA:00005_spectrohost.ca.com@spectrohost.ca.com refers to a CA Spectrum connector installed on spectrohost.ca.com monitoring a CA Spectrum instance that is installed on the same system.
  Value: String
- **Symptom**
  Indicates that the root cause is a symptom.
  Value: Yes or No
- **Ticket ID**
  Indicates the associated help desk ticket number.
  Value: String
- **Unclassified**
  Indicates the root cause in unclassified.
  Value: Yes or No

CI User Attributes

An administrator sets the CI user attributes and the attributes are labeled 1-5. CI user attributes let you define custom CI user attributes that are not provided on the attribute list through the USM schema. For more information about setting CI user attributes, see Set CI User Attributes.

Customer Properties

- **# Impacted Customers**
  Indicates the number of customers that the alert impacts. This number is based how many customers that are assigned to the service that an alert impacts.
  Value: Number
- **# Impacted Services**
  Indicates the number of services the alert impacts. The number is based on how many services its associated CI is included in.
  Value: Number
- **Customer ID**
  Indicates the customer identification number.

Value: String

- **Customer Impact**
  Indicates the alert impact to the customer.
  Value: Down, Moderate, None, Severe, or Slight
- **Customer Name**
  Indicates the customer name.
  Value: String
- **Customer Priority**
  Indicates the customer priority that the administrator sets.
  Value: 1-10
- **Highest Customer Impact**
  Indicates the highest impact that the alerts caused for an associated customer.
  Value: Down, Moderate, None, Severe, or Slight
- **Highest Customer Priority**
  Indicates the highest customer priority number.
  Values: 1-10

Define Escalation Action Retry Behavior

The SA Manager automatically retries failed escalation actions according to a retry frequency and duration that you can configure. The retry mechanism recurs until either the action is successful or the duration threshold is reached.

If any of the following conditions are true, a retry does not occur for a failed action:

- The action is deleted or disabled
- The retry duration threshold is reached
- The alert that is associated with the action is deleted
- You migrated the action from a previous release

Follow these steps:

1. Click the Administration tab.
2. Click the plus sign (+) next to CA SOI Insight Manager Configuration.
3. Click the plus sign (+) next to the server you want to configure.
4. Click Global Settings.
5. Scroll down to the Escalation Policy and Action Settings section.
6. Complete the following fields:
   - **Perform Action Retries**
     Controls if CA SOI retries the escalation actions that are associated with an alert.
     Select Yes to retry again in the number of minutes entered in the Retry Frequency field and for the total number of days entered in the Retry Duration field.
     Select No to quit an escalation action after one failed attempt.
     Default: Yes
   - **Retry Frequency (minutes)**
     Defines the number of minutes CA SOI retries escalation actions after a failed attempt.
     Default: 30
   - **Retry Duration (Days)**
     Defines the number of days CA SOI continues to retry escalation actions after failed attempts. At the end of this duration, CA SOI stops attempting failed escalation actions.
     Default: 2
7. Click Save.
8. Restart the CA SAM Application Manager service.
9. Monitor the status of initiated actions from either of the following locations in the Operations Console:

- Escalation Action History table in the Alert Detail tab of the Component Detail pane (by alert)
- Action History tab of the Escalation Policies and Actions dialog (by escalation policy)

The following columns contain information about action status:

- **Time Executed**
  Defines the time that the action was first run.
- **Succeeded**
  Defines whether the action has completed successfully.
- **Time Retried**
  Defines the time of the last action retry on failure. When the duration threshold is reached, the Time Retried field displays the last time that the action was attempted.

Configure Policy Assignments by Alert Queue

When you create or edit an alert queue, you can configure the related global and queue-specific escalation policies.

Follow these steps:

1. Right-click an alert queue on the Operations Console Alert Queues tab in the Navigation pane, and select Edit Queue.
   The Edit Alert Queue wizard opens on the Define Queue Criteria page.
2. Click Next.
   The Assign Escalation Policies page opens.
3. Select one of the following options in the Global Escalation Policies pane:
   - **Apply the following global policies for this Queue**
     Applies any existing global escalation policies to the alert queue.
   - **Do not apply the following global policies for this Queue**
     Ignores the alert queue when evaluating any global escalation policies.

   The selection applies to all global escalation policies.
   You can also create, edit, or delete global policies in the Global Escalation Policies pane.
4. Select one or more escalation policies in the Available Policies pane and move them to the Current Policies pane.
   You can add more than one queue-specific policy to an alert queue.
   You can also create alert queue-specific escalation policies in the Queue Specific Escalation Policies pane by clicking New.
5. Click Save.
   The policy starts evaluating the alert queues after you save the alert queue.

Configure Policy Assignment by Service

When you create or edit a service, you can configure the global and service-specific escalation policies.

Follow these steps:

1. Perform one of the following actions:
   - Right-click a service and select Edit Service.
   - Select Tools, Create New Service.
2. Go to the Alert Escalation tab and select one of the following options in the Global Escalation Policies pane:
   - **Apply the following global policies for this service**
     Applies any existing global escalation policies to the service.
   - **Do not apply the following global policies for this service**
     Ignores the service when evaluating any global escalation policies.

   The selection applies to all global escalation policies.
   You can also create, edit, or delete global policies in the Global Escalation Policies pane.
3. Select one or more escalation policies in the Available Policies pane and move them to the Current Policies pane.

You can add more than one service specific policy to a service.

> **NOTE**
> Alerts on subservice CIs do not trigger a parent service's escalation policy. If you want subservice CI alerts to trigger the policy for the parent, you must also assign each subservice to the escalation policy.

You can also create service specific escalation policies in the Service Specific Escalation Policies pane by clicking New.

4. Click Save.
   The policy starts evaluating the service's alerts after you save the service.

How to Create and Manage Alert Queues

As an administrator, you can define alert queues then assign the escalation policies and user groups to the queues.

Alert queues are user-defined alert groups. CA SOI auto-assigns alerts to a particular alert queue based on user-defined policy, which can include alert content and associated CIs.

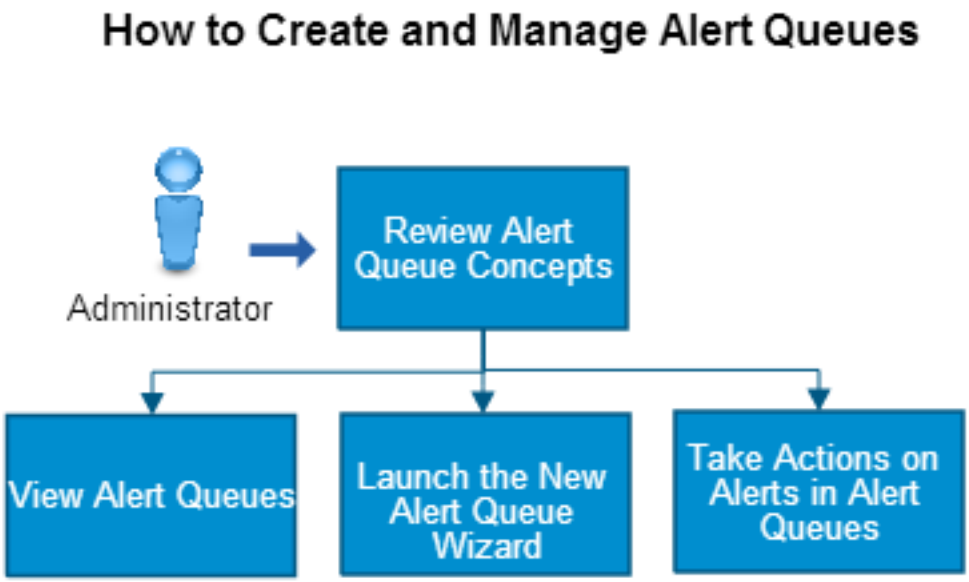
Alert queues let you group alerts as they come in based on specific criteria to monitor the status of your infrastructure more efficiently. You can add global and non-global escalation policies to alert queues to take a specified action automatically on alerts that come into a queue.

For example, consider a company with engineers responsible for different aspects of the infrastructure, such as networks, systems, and databases. Without defined queues, alerts from all integrated domain managers appear in one consolidated view on the Alert Queues tab. The administrator can define queues by domain (such as Network Alerts or Database Alerts). With organized queues, engineers can quickly find and resolve their alerts. Additional queues could be defined based on other alert categories, such as severity, assignment status, and description to enable an optimized unified alert management system.

Services provide a similar organizational function as alert queues at a higher level with the additional benefit of resource topology and impact analysis. Defining alert queues is less intensive than modeling services, and they can simplify alert management as you make the transition to a service-oriented management paradigm. Alert queues can also remain useful in an environment with services defined to provide a supplemental management perspective outside of services. For example, you can define a queue for alerts that have not been acknowledged or a queue for alerts from the same source domain manager.

Use this scenario to guide you through the process:

Figure 38: How to Create and Manage Alert Queues



1. Review the alert queue concepts to learn about the default alert queue and how alert queue security works.
2. View your alert queues.
3. Launch the New Alert Queue wizard to create alert queues.
4. Take actions on the alerts in your alert queues.

Review Alert Queue Concepts

The following concepts provide information about the default alert queue and how the alert queue security works.

Alert Queues and Security

Your CA SOI administrator assigns you to user groups. The administrator sets user group access to specific CA SOI features, services, alert queues, customers, and so on. Consider the following items when viewing alert queues:

- You can view all nonservice-impacting alerts and they can show in your alert queues. Because non service-impacting alerts are not associated with a service, they cannot be restricted with access privileges.
- You can view alert queues only to which you have access privileges.
- You can view the alerts in alert queues only to which you have access privileges.
- You can view all the alerts of a service regardless of the access privileges.
- You can edit alert queues for which you have access privileges. Only administrators can edit the Default queue.
- You can only delete alert queues that you created.

Default Queue

The Default queue is the only predefined alert queue. The Default queue contains all managed and unmanaged alerts to which you have access, but the alerts do not appear in another queue. You cannot rename or delete the Default queue or edit the queue criteria. However, administrators can edit the Default queue escalation policy and group assignment. The Default queue is available to administrators only by default.

Launch the New Alert Queue Wizard

You create alert queues by launching the wizard.

Follow these steps:

1. Start the Operations Console and click the Alert Queues tab in the Navigation pane.
2. Click Create a new Alert Queue

   or right-click a queue and select New Queue.
   The topics that follow guide you through alert queue creation using the wizard.

Define Queue Criteria

The Define Queue Criteria page lets you name your new alert queue and define the criteria that CA SOI uses to filter the alerts and automatically add them to the queue. The filter criteria are based on alert attributes, and you can build criteria with Boolean expressions to create advanced filter expressions.

Follow these steps:

1. Enter a queue name and an optional description.
2. (Optional) Set the queue priority, which determines the order in which escalation policies are applied if multiple queues contain the same alert. One is the lowest priority, and ten is the highest; the default is five.
3. Define the alert queue criteria in the Queue Criteria panel:

   NOTE
   The topic that follows provides attribute information.

   a. Select an alert attribute on which to filter, a comparison type, and a comparison value for the attribute, and click Add.

      NOTE
      To use regular expressions, select Matches regex from the Comparison Type drop-down list. Click Test Regex to open the Regex Tester and test the regular expression against a string. Regular expressions are not available for all attributes.

      You can define criteria based on the alert properties and the correlatable USM properties of the associated CI. The attribute expression appears in the lower pane.

   b. (Optional) Add more attribute criteria and create advanced logic using the logic buttons on the right of the dialog. For more information about creating advanced attribute criteria, click the Hints link above the expression pane.

   c. Click Next.
      The Assign Escalation Policies page opens.

Alert Queue Attributes

You can build the alert queue criteria using the following attributes:

USM Properties

Any attribute not listed in the categories that follow are USM types and properties. For USM definitions, see How to Access the USM Schema Documentation.

Alert Properties

- **Acknowledged**
  Indicates that the alert is acknowledged.
  Value: Yes or No
- **Assigned**

Indicates the name of the operator that is assigned to the alert.
Value: String

- **Business Priority**
  Indicates the service priority.
  Value: Unspecified, None, Medium, Low, High, or Critical
- **Category**
  Indicates whether this alert condition affects the quality or risk of the services it impacts.
  Value: String
- **Is Clearable**
  Defines whether the alert can be cleared in the domain manager. If you enabled the 'Respect Underlying MDR Clear Alert Setting', this property also determines whether the alert can be cleared in CA SOI.
- **Message**
  Indicates a message that an operator entered.
  Value: String
- **Non-Service Impacting Alert**
  Indicates that the alert does not affect a modeled service.
  Value: Yes or No
- **Root Cause**
  Indicates if the alert is the root cause.
  Value: Yes or No
- **Service Impact**
  Indicates the impact, which is calculated by multiplying alert severity and the significance of the CI to the service. When multiple services are impacted, the most affected service is displayed.
  Value: Down, Moderate, None Slight, or Severe
- **Service Name**
  Indicates a service name that is associated with an alert.
  Value: String
- **Severity**
  Indicates the alert severity that the originating domain manager assigned.
  Value: Critical, Down, Major, Minor, Normal, or Unknown
- **Source**
  Indicates the domain manager where the alert originated. The format is MdrProduct_domainserver@connectorserver. For example, CA:00005_spectrohost.ca.com@spectrohost.ca.com refers to a CA Spectrum connector installed on spectrohost.ca.com monitoring a CA Spectrum instance that is installed on the same system.
  Value: String
- **Symptom**
  Indicates that the root cause is a symptom.
  Value: Yes or No
- **Ticket ID**
  Indicates that the associated help desk ticket number.
  Value: String
- **Unclassified**
  Indicates the root cause in unclassified.
  Value: Yes or No
- **Unmanaged**
  Indicates if the alarm is associated with a model CI or not.
  Value: Yes or No

Customer Properties

- **# Impacted Customers**

Indicates the number of customers that the alert impacts. This number is based on the number of customers that are assigned to the service that the alert impacts.
Value: Number

- **# Impacted Services**
  Indicates the number of services that the alert impacts. The number is based on how many services the alert's associated CI is included in.
  Value: Number
- **Customer ID**
  Indicates the customer identification number.
  Value: String
- **Customer Impact**
  Indicates the alert impact to the customer.
  Value: Down, Moderate, None, Severe, or Slight
- **Customer Name**
  Indicates the customer name.
  Value: String
- **Customer Priority**
  Indicates the customer priority the administrator sets.
  Value: 1-10 or the values that the administrator configured.
- **Highest Customer Impact**
  Indicates the highest impact that the alerts caused for an associated customer.
  Value: Down, Moderate, None, Severe, or Slight
- **Highest Customer Priority**
  Indicates the highest customer priority number.
  Values: 1-10 or the values that the administrator configured.

Assign Escalation Policies

The Assign Escalation Policies page lets you apply or exclude global and non-global escalation policies to your alert queue.

> **NOTE**
> If you do not want to apply or exclude global policies, click Next to continue to the Assign User Groups page.

Follow these steps:

1. Select whether you want to apply or exclude all global policies.
2. Select the policies you want to include or exclude from the policy list.

   > **NOTE**
   > Ctrl + click to select multiple policies or click Create a new global policy
   >
   >
   >
   > to create a new policy.

   The global policies that you include will evaluate alerts in the queue and trigger the escalation policy when they meet the policy criteria.
3. Click the arrow buttons in the Queue Specific Escalation Policies pane to add queue specific non-global escalation policies.
   The escalation policies in the Current Policies pane will evaluate alerts in the queue and trigger the escalation policy when they meet the policy criteria
4. Click Next.
   The Assign User Groups page opens.

Assign User Groups

The Assign User Groups page lets you allow or deny user groups access to the new alert queue.

Follow these steps:

1. Use the arrow keys to move the Available Groups that currently do not have access to your alert queue to the Allowed Group list to grant that use group access. Perform the opposite to deny a user group access to your alert queue. User groups marked with an asterisk (*) have automatic access to all user groups.
2. Click Next.
   The Confirm page opens.

Confirm

The Confirm page lets you review the new alert queue properties you selected.

Click Finish to close the Confirm page and create the new alert queue, which appears in the Alert Queues tab.

> **NOTE**
> The new alert queue may not appear for several seconds.

Regular Expressions

Contents

Regular expressions are supported in the following areas:

- Alert filters
- Alert queue criteria
- Escalation policy definitions and action exception criteria
- Service Discovery dynamic services.

Regular Expressions Considerations

Consider the following situations when using Regular Expressions in CA SOI:

Regular Expressions in CA SOI act as a *find,* not as a *match.*

A *find* searches for the pattern across the strings, including the substrings.

A *match* searches for the pattern in the strings only.

Example:

With the following strings: "cart" and "artistic":

- A *find* for the "art" pattern finds "art" in the substrings of both the "cart" and "artistic" strings.
- A *match* for the "art" pattern does not match in "cart" or "artistic" strings because a *match* does not search the substrings.

To perform a *match* in CA SOI, enclose the pattern with "^" and "$", such as "^art$". You can use the Regex Tester to verify your expressions before implementing them.

Use the Regular Expression (Regex) Tester

You can use the Regular Expression (Regex) Tester to validate a regular expression before using the expression in CA SOI.

Follow these steps:

1. In a dialog that supports regex, click Test Regex.
   The Regex Tester dialog appears.
   The Operation field describes the conditions:
   – Case sensitivity
   – If the pattern is to match or not match
   If you entered an expression in the Attribute Value field, the expression appears in the Regex Pattern for editing.
2. Enter (or edit) the regular expression in the Regex Pattern field.
   The Valid? field indicates if the expression you entered is a valid regular expression. The field displays Yes or No.
3. Enter a test string in the Test Text field.
   The Found?/Not Found? field indicates if the regular expression finds or does not find the Test Text string, based on the Operation conditions. The field displays True or False.
4. Perform one of the following actions:
   – Click Use Pattern to close the RegEx Tester dialog and transfer the Regex Pattern to the Attribute Value field.
   – Click Cancel to close the RegEx Tester dialog and leave the Attribute Value field unchanged.

Event Management

This section introduces Event Management architecture and describes how events are stored.

See the following topics for details:

Intended Audience

This guide is intended for any user who is responsible for managing any or all of the productevent activity. This user base can include operators for a specific IT function, domain administrators, product administrators, help desk operators, and so on. This guide refers to all these users collectively as an operator.

Introduction to Event Management

Contents

An event is a message that indicates an occurrence or change in your enterprise. Events can indicate a negative occurrence or object state change. Event management is crucial to understanding the dynamic state of an enterprise across the network, security, system, application, service, and other domains. As the number of resources grows exponentially, so do the challenges of understanding and administering diverse management events from those resources.

CA SOI provides a scalable event management collection and distribution solution (Event Management) with remote visualization and administration capabilities. Entities that connectors report with the USM type of alert are collected into the Event Store on each connector system and are available for Event Management operations. After Event Management processing is complete, the processed events become alerts that you can escalate, add to alert queues, and include in service impact analysis. Typically, with no deployed event policies, all events become alerts. Event Management lets you control the event stream so that a consolidated, high quality, and actionable set of alert conditions appear in the Operations Console as alerts.

Implementing Event Management in a complex distributed environment helps organizations create a unified event management system for their enterprise, providing a holistic view of the entire IT infrastructure. The solution lets organizations collect, transform, correlate, filter, enrich, and manage events from various sources (such as network devices, applications, and enterprise servers) across the enterprise. You can track the events and can create policies to have a manageable set of actionable conditions and convert those conditions for escalation and inclusion in alert queues.

Event Types

Event Policies

An *event policy* is a combination of event search patterns and an action to perform when the patterns match. You can perform the following event policy actions:

- Save them for an on demand view of events that match the search patterns.
- Deploy them to evaluate incoming events that are dynamically based on the search criteria and perform the specified action in response to matches.

Event policies let you manage when and how events become alerts in CA SOI. CA SOI provides the following action types:

- Correlation to associate related events that are based on any criteria
- Filtering to eliminate extraneous events and subsequently lower alert volume
- Creating new events as a result of event policies to consolidate multiple conditions into one actionable condition
- Enriching events to add vital information from outside sources
- Normalization to map raw events to USM alert properties

Event Management Features

The Event Policy dialog in the Operations Console provides access to Event Management functionality. To access the Event Policy dialog, select Tools, Event Policies. You can do the following on the Event Policy dialog:

- Perform federated event searches on all historical events or on specific connector event sources to analyze events and detect event patterns.
- Save event searches as policies that you can access at any time to view the most recent search results.
- Create and deploy policies that detect patterns in real-time (based on historical event searches) and perform actions on events.

The robust event search system lets you track all events and deploy policies. You can then create a manageable set of actionable conditions and convert those conditions to alerts for escalation and inclusion in alert queues, if necessary. This extra layer of event processing is important to equalize the quality of event and alert data coming from the diverse set of connector sources. For example, a generic trap from the SNMP connector requires extra processing before it can be as useful as a fault alarm from the CA Spectrum connector.

Event Management provides functionality that enables the following common event actions:

- Correlation to associate related events based on any criteria
- Filtering to eliminate extraneous events and subsequently lower alert volume
- Creating new events as a result of event policies to consolidate multiple conditions into one actionable condition
- Enriching events to add vital information from outside sources
- Normalization to customize the mapping of raw events from specific connectors to the USM alert schema

Event Management Architecture

Contents

Event Management uses a distributed and manager-client model to help ensure scalability through filtering events as close to the data sources as possible. The query and retrieval of events is also done in a federated approach and the results are combined before being made available for searches.

The distributed, scalable architecture of Event Management simplifies event handling by providing the following functional capabilities:

- Event processing capabilities at two primary levelssimple and complex. Simple processing includes normalizing from an event source (raw) schema to the common (USM) schema. Complex processing includes filtering, consolidating, and refining message content according to the discovered patterns.
- Support for remote searches of event data to achieve the following objectives:
  - Enable a federated model that does not require large volume event data to move around the subsystem.
  - Support visualization requirements through integration with the Operations Console for event searches and policies.
- Ability to perform federated event searches on all events or on specific connector event sources to analyze events and detect event trends.
- Local cache and persistence within each node or tier.
- Ability to save event searches as policies that you can access at any time to view the most recent search results.
- Ability to create and deploy event policies that perform actions on events that match the search results.

How Events are Processed in Event Management

The following steps explain the flow of events in Event Management:

1. All message data collected from source domain managers (alerts, events, notifications, and so on) are converted to the USM alert schema format.
2. All events retrieved from connectors with the USM type of alert are collected into the Event Store component of Event Management on each connector system as events and are available for Event Management operations.
3. After Event Management processing completes, the processed events become alerts and are forwarded to CA SOI. You can escalate these alerts, add them to alert queues, and include them in service impact analysis.
4. (Optional) The Mid-tier connector, if used, acts as a single point of contact to collect normalized cross-domain connector alerts and perform actions on them before forwarding to CA SOI. In this case, all events flow through the Mid-tier connector before reaching the Operations Console as alerts.

Use the Operations Console to have federated access to the events stored in the Event Store. You can manage event rules and policies and configure the stored events.

Event Management Components

Event Management consists of the following components:

- Event Store
- Mid-tier Connector
- Event Service (Query/Deploy Service)
- Event Management UI Service

Event Store

The Event Store component includes XML files that store raw and normalized events that are collected from connectors. Federated access to the events stored in the Event Store XML files is available through the Operations Console. You can run searches to access the appropriate information.

The Event Store is installed with IFW, and Event Store XML files are stored on the same server where the corresponding connector is available. Therefore, if you have different connectors installed on different servers, each connector server includes a separate EventStore folder to store events from the connectors on that server. You can find the XML files under the <SOI_HOME>\resources\Core\EventStore folder (<EI_Home>\Core\EventStore in the case of the Event connector).

The EventStore folder also includes two folders: temp and archive. The *temp* folder contains temporary event files that cache incoming events, and the *archive* folder contains the archived Event Store XML files.

The following example shows how the information is organized in an Event Store XML file for the normalized and raw version of an event:

```
< events>
   <event
 container='siloName=CA:09998_ABC77.ca.com@ABC77.ca.com;requestID=cdcc108f-4f29-4070-8704-34d214752646;publishAction=IM
    <raw>
    <connectorID>f8124018-c599-4bbe-bac0-459c9a9b0c7c</connectorID>
    <PrimaryIPV4Address>172.31.255.255</PrimaryIPV4Address>
    <SAMID>ALL</SAMID>
    <siloName>CA:09998_ABC77.ca.com@ABC77.ca.com</siloName>
    <publisher>JMS</publisher>
    <publishAction>IMPORT</publishAction>
    <MdrElementID>mirror_Server_005</MdrElementID>
    <ClassName>ComputerSystem</ClassName>
    <entitytype>Item</entitytype>
    <siloHost>ABC77.ca.com</siloHost>
    <dns_resolution>1</dns_resolution>
    <MdrProduct>CA:09998</MdrProduct>
    <Description>mirror_Service:mirror_DBCluster02.xyz.com</Description>
    <LastModActivity>Create</LastModActivity>
    <LabelSection_ComputerSystem_1>mirror_DBCLUSTER02</LabelSection_ComputerSystem_1>
    <MdrProdInstance>ABC77.ca.com</MdrProdInstance>
    <temp_atleastoneset>172.31.255.255</temp_atleastoneset>
    <ComputerName>mirror_DBCluster02.xyz.com</ComputerName>
    <ConnectorConfigMdrProduct>CA:09998</ConnectorConfigMdrProduct>
    <requestID>cdcc108f-4f29-4070-8704-34d214752646</requestID>
    <siloID>f6c3a593-26dd-4862-b7ed-bce932e57e7c</siloID>
    <Label>mirror_DBCLUSTER02</Label>
    <connectorName>ABC77.ca.com</connectorName>
    <PrimaryDnsName>mirror_DBCluster02.xyz.com</PrimaryDnsName>
    <SysName>mirror_DBCLUSTER02</SysName>
    <eventtype>USM-Entity</eventtype>
    <ConnectorConfigMdrProdInstance>ABC77.ca.com</ConnectorConfigMdrProdInstance>
    </raw>
    < normal>
    <LastModActivity>Create</LastModActivity>
    <MdrProduct>CA:09998</MdrProduct>
    <MdrProdInstance>ABC77.ca.com</MdrProdInstance>
    <MdrElementID>mirror_Server_005</MdrElementID>
    <ClassName>ComputerSystem</ClassName>
    </normal>
     </event>
   ....
   ....
</events>
```


Event Store XML File Naming Convention

The Event Store creates a separate XML file hourly for events from each connector. The naming convention of an Event Store XML file is *MdrProduct-MdrProductInstance*!*year-month-day*!*hour*!*counter*.xml. For example, if the name of an Event Store XML file is CA-*09998-ssa-cat-xyz04.axy.com!2010-12-06!16!1.xml*, the file name then represents the following:

- **CA-09998**
  Represents the MdrProduct value of the connector that retrieved the event.
- **ssa-cat-xyz04.axy.com**

Represents the MdrProductInstance value of the connector that retrieved the event.

- **2010-12-06**
  Represents the date when the file was created.
- **16**
  Represents the hour when the file was created.
- **1**
  Represents the initial counter value.

The temporary event files roll over to the main Event Store XML file at a regular interval (approximately every 7 seconds). This interval is configurable.

The default size of the main Event Store XML file is 10 MB. You can configure this value based on your requirements. When the file size reaches the specified limit or a new hour starts, the events are rolled over to the new Event Store XML file.

Archiving Event Store XML Files

Event Management automatically archives and moves the Event Store XML files to the *archive* folder available under the main EventStore folder. Archiving old XML files provides the following benefits:

- Reduces the disk space usage by zipping archived files
- Prevents flooding the hard drive through automatic archival when low disk space is detected
- Maintains only the latest and required XML files in the main folder
- Improves the performance of the queries (because you have less data to query)
- Allows data to be purged or stored offline without requiring a backend services stoppage

How the Archiving Process Works

The archiving process includes the following steps:

1. Old XML files that are stored in the EventStore folder are added to a zip file.
2. The zip files move to the archive folder.
3. The old files are deleted from the EventStore folder.
4. The archived zip files are purged from the system after a specific amount of time.

Archiving removes the old data from the EventStore folder that is based on the archive retention interval. The archive retention interval represents the number of days the Event Store data is retained until it is archived. You can change the archive retention interval that is based on your organization requirements to decide how you want to archive your files.

For example, consider a scenario where the organization policy mandates that you must keep the files in the EventStore folder for seven days. You can archive an XML file only after it completes the prescribed period of seven days. In this case, you can set an archive retention policy that validates the XML files. The policy also archives the XML files only when the seven-day period expires.

The default archive and purge settings are as follows:

- Event Store XML files are moved to zip files in the archive folder after one day.
- Archived zip files in the archive folder are purged from the system after 30 days in the archive folder.
- If the disk volume on the system moves below 20 percent, the Event Store automatically archives all files to preserve the disk space.

> **NOTE**
> By default, archiving starts at midnight, and you cannot change this time. Additionally, you cannot archive any file on the day it is created. The policy requires a minimum of a one day's (24 hours) difference to be able to archive the file.

Archived File Restoration

An archived file is restored when you search (using appropriate scoping parameters in the Event Policies dialog) for events that are archived in the *archive* folder. The archived file is unzipped and restored in the EventStore folder.

> **NOTE**
> The EQUERY_UNZIP_ARCHIVE parameter determines whether the search would unzip the archived files. You can find this parameter in the <SOI_HOME>/tomcat/lib/eventManagerClientConfig.xml file. For more information about how to configure this parameter, see Configure Event Search Settings.

Configure the Event Store Parameters

You can configure the Event Store parameters per connector that are based on your requirements. For example, you can configure the archive retention interval that is based on your corporate archiving policy. You can also control how to archive the Event Store XML files and configure when archived files are purged.

Follow these steps:

1. Open the <SOI_HOME>\jsw\conf folder.
2. Locate and open the SAM-IntegrationServices.conf file in a text editor.
3. Add any of the following properties to the file, and save the file:

> **NOTE**
> Any properties that you do not add continue to use the default values defined in this section.

- **ESTORE_ARCHIVE_RETENTION**
  Specifies the number of days you want to keep the Event Store XML files in the EventStore folder before they are zipped and moved to the *archive* folder. If the disk volume on the system moves below 20 percent, the Event Store ignores this value and archives all current files.
  If your system uses enough disk space to activate this automatic archival, ensure that you enable searching of archived Event Store files. By default, Event Management does not search archived files.
  Default: 1
- **ESTORE_FREE_DISK_SPACE**
  Defines the threshold of free disk space percentage below which the Event Store automatically archives all files. For example, if you set this value to 10, the Event Store archives all files (regardless of the archive retention settings) when the free disk space on the system is below 10%.

  > **WARNING**
  > If an automatic archive occurs and it does not cause the available disk space to move above the defined threshold, another automatic archive occurs the next time the Event Store polls for available disk space. The second automatic archive overwrites the initial one, purging the events from the first archive.

  Default: 20
- **ESTORE_FREE_DISK_LIMIT**
  Defines the threshold of free disk space in GB below which the Event Store automatically archives all files. Combined with the ESTORE_FREE_DISK_SPACE property, archiving is done only if both the percentage of free disk space and the total amount of free disk space are below the respective property values. For example, on a large disk, 20% is still enough capacity for Event Store to continue writing the event data. On the other hand, on a small disk a 20% restriction allows writing of event data even after reaching the ESTORE_FREE_DISK_LIMIT value.
  Default: 10
- **ESTORE_PURGE_INTERVAL**
  Specifies the number of days you want to keep the Event Store archived zip files in the archive folder before purging them from the system. Set this property to -1 to retain all archived files indefinitely.
  Default: 30
- **ESTORE_ROLLOVER_FILESIZE**
  Specifies the Event Store XML file size in MB. When the file size increases beyond the specified limit, the file is rolled over to another Event Store XML file.

NOTE
We recommended that you do not increase the file size beyond 25 MB. You can encounter some issues while querying such large files.

Default: 10

– **ESTORE_MERGE_TIME_INTERVAL**
Specifies the time interval in seconds after which the size of the Event Store XML file is verified. The file is verified to see whether it has reached its specified limit and the file is ready to be rolled over to another file.
Default: 7

The Event Store parameters are configured.

4. Repeat Steps 1-3 for all connector configuration files that require the updated Event Store settings.
5. Restart the CA SAM Integration Services service.
The changes take effect.

Mid-tier Connector

The Mid-tier connector collects normalized connector alerts from all connectors, performs operations such as enrichment, filtering, correlation, and so on across domains, and publishes the resultant alerts to CA SOI. Therefore, it acts as a single point of contact and provides cross-domain information to CA SOI.

Using the Mid-tier connector offers the following benefits:

- Helps optimize the overall performance of the solution by handling large volume of cross-domain events.
- Provides cross-domain correlation, which policies on single connectors cannot perform. When you deploy a policy on a set of connectors, correlation occurs only within each connector source. The Mid-tier connector, due to its position in the event flow, can correlate across all data sources.
- Enriches events coming from different domains and adds additional information to the events, which helps administrators manage alerts more efficiently.
- Enables out of the box enrichments such as CA Spectrum and CA CMDB.
- Lets you perform enrichments on CA Catalyst connector data that are not supported directly on CA Catalyst connectors, such as JDBC enrichments.
- Performs actions on matching events from all data sources by default when you deploy policy to it.
- Reduces the number of alerts that are forwarded to CA SOI, decreasing the time to interpret and resolve critical alerts. For example, if you have received five different CPU-related events with the same severity from five different domain managers, you can decide which domain alert to suppress and which one to move forward and manage, based on the domain critically impacting your business services. This way, you can triage various cross-domain alerts depending on their business impact. You can also consolidate closely related alerts that share common property values into a single alert and reduce the administrative overhead associated with resolving duplicate alerts.

Event Service

The event service component provides the querying and deploying services. It performs several functions, such as the following:

- Performs event queries and retrieves events from the Event Store for those queries
- Collects connector information
- Deploys or gets a policy

The event service gets installed on the connector computer as a Windows service (CA SAM Event Management) when the IFW (CA SAM Integration Services) is installed.

Because you can distribute connectors on multiple computers or install them on a single computer (or both), the event service is also installed on each of those connector computers and manages events and information from those

connectors. A request from the Operations Console in the form of a query communicates with each of those event services (local or remote), federates queried data from multiple event services, and displays the federated data to users.

You do not need to run separate queries to get information from different Event Stores; a single query interacts with all the event services and helps ensure that you receive only the federated information from various Event Stores.

Event Management UI Service

The MQ server that is installed with CA SOI facilitates the overall communication between the event service and the Event Policies dialog in the Operations Console. In this interaction, the Event Management UI service acts as a client on behalf of the Event Policies dialog in the Operations Console. The event service acts as a server that completes the data requests that it receives from the client.

The Event Management UI service runs as part of the SA Manager. The XML file *eventManagerClientConfig.xml* is the configuration file for the Event Management UI service, which you can configure based on your requirements.

Configure the eventManagerClientConfig.xml File

You can configure the eventManagerClientConfig.xml configuration file to decide how you want to control the Event Management data flow to the Event Policies dialog.

Follow these steps:

1. Open the SOI_HOME\tomcat\lib\eventManagerClientConfig.xml file in a text editor.
2. Configure the following parameters, and save and close the file:
   - **timeoutValues**
     Specifies the amount of time in seconds that the event service waits for a response to a query request. Each type of request can have its own value. You can set the timeout values for the following actions:
     - DeployPolicy
     - DeployScript
     - GetConnectorInfo
     - GetDeployedPolicy
     - GetDeployedScript
     - GetEvents
   - **synchInterval**
     Specifies the polling interval in seconds for determining the available event services. The list of data sources available in the Event Policy dialog (Tools, Event Policies) in the Operations Console reflects the current state of this polled information. Any changes to the status of data sources may take up to the interval time (specified for this parameter) to update in the Operations Console.
     Default: 45
3. Restart the CA SAM Application Server service.

Event Management Examples

The examples in this document illustrate how to work with and make efficient use of Event Management functionality. The examples are spread throughout the document in the sections that describe the related functionality. The following types of examples are available:

- **Event searches**
  Event search examples show example search patterns for all types of event searches, such as time-based correlation, raw event searches, and so on. The following event search examples are available:

  - Time-based correlation
  - Occurrence frequency
  - Advanced search techniques
  - Raw events
  - Moving from simple to complex
- **Event policies**
  Event policy examples show example event policy creation for all types of policies, such as enrichment, normalization, and so on. The following event policy examples are available:
  - Filter action
  - Create event action
  - Database enrichment
  - Java method enrichment
  - Script enrichment
  - Map enrichment
  - Normalization
- **End-to-end scenarios**
  End-to-end scenarios combine event searches with policy actions to provide real-world use cases. Scenarios also include ways that you can leverage other product functionality to take advantage of Event Management data. The following end-to-end scenarios are available:
  - Filter duplicate events from integrated domain managers
  - Create a new event to indicate a crashing service
  - Combine a create event action with an enrichment using reevaluation
  - Normalize monitoring traps

Event Searches

This section describes how to run event searches that you can use to find events and leverage in event policies

Event Properties and Event Information

As an administrator, you search for and interact with events using the properties for the USM alert type. The valid properties appear when you right-click an Event Pattern field on the Event Search tab of the Event Policy dialog. Valid property values also appear for enumerated properties.

All USM alert properties are supported in searches, but many are optional properties that are not present in every event.

Use the lists of properties that follow to understand the information depicted by each property. Use the right-click menu in the Event Search tab to add properties to a search and add valid values for properties with enumerated values.

The properties present in every event that you can use in normalized event searches are as follows:

> **NOTE**
> For more information about USM alert properties, see How to Access the USM Schema Documentation.

- **AlertedMdrProduct**
  Defines the domain manager that originated the event. The tooltip for each data source on the left pane displays this value as Connector Type.
  The right-click menu in the event search tab displays these numeric values as domain manager names for increased usability. Always use the right-click menu to assign an AlertedMdrProduct value to avoid having to manually enter the numeric value.
  Example: CA:09998 (Sample Connector)
- **AlertedMdrProdInstance**

Defines the domain manager system that originated the event. This property is typically the host name of the system where the domain manager is installed. The tooltip for each data source on the left pane displays this value as Instance Name.

- **AlertedMdrElementID**
  Defines the unique identifier of the CI that originated the event.
- **AlertType**
  Defines the type of condition that the event reports. The most common valid values are Quality, Risk, Compliance, and Cost.
- **Severity**
  Defines the event severity.

  > **NOTE**
  > Even though events with severity values of normal and informational are returned in event searches and can participate in event policies, events with these severities cannot appear as alerts in the Operations Console.
- **Summary**
  Defines a summary description of the event.

You can include the following properties in event searches using the provided scoping controls for source and time. Therefore, they are typically not required in the actual event search pattern:

- **MdrProduct**
  Defines the domain manager that originated the event. The tooltip for each data source on the left pane displays this value as Connector Type.
  The right-click menu in the event search tab displays these numeric values as domain manager names for increased usability. Always use the right-click menu to assign an MdrProduct value to avoid having to manually enter the numeric value.
  Example: CA:09998 (Sample Connector)
- **MdrProdInstance**
  Defines the domain manager system that originated the event. This property is typically the host name of the system where the domain manager is installed. The tooltip for each data source on the left pane displays this value as Instance Name.
- **MdrElementID**
  Defines a unique identifier for the event.
- **OccurrenceTimestamp**
  Defines when the condition that caused the event occurred. This property uses the xs:dateTime format: YYYY-MM-DDTHH:MM:SS.SSS-Z.
- **ReportTimestamp**
  Defines when the event was created. This property uses the xs:dateTime format: YYYY-MM-DDTHH:MM:SS.SSS-Z.

The optional event properties that you can include in event searches are as follows. Not all events have these properties assigned, which would eliminate them from any search using these properties:

> **NOTE**
> When you select a property added to the usm-core2 update to the USM schema, it appears in the search pattern with a 'usm-core2:' prefix

- **AlertCategory**
  Defines a high-level category, such as Application, SystemAndStorage, and so on.
- **Assignee**
  Defines the Person CI to which the event is assigned in the following format: MdrProduct,MdrProdInstance,MdrElementID.

  > **NOTE**
  > Assigning an alert from the Operations Console does not affect this event value.
- **AssigneeUserName**

Defines the user name or login ID of the person assigned to the alert, if known.

- **Comments**
  Defines comments associated with the alert.
- **ElapsedTime**
  Defines the duration over which a number of identical events occurred. This property uses the xs:duration format.
- **ExtendedMessage**
  Provides a complete alert message when the message is longer than the 1024 character length permitted by the Message property.
- **ExtensionNameValuePairs**
  Defines a comma-separated string of name-value pairs, where the name and value are separate by an equal (=) sign.
- **ImpactedEntities**
  Defines a semi-colon-separated list of CIs experiencing issues related to this event. This property can only have a value when the AlertType is Risk-RootCause, and is therefore the root cause impacting other CIs. Each impacted CI is listed using the following format: MdrProduct,MdrProdInstance,MdrElementID.
- **IsAcknowledgeable**
  Defines whether the event can be acknowledged.
- **IsAcknowledged**
  Defines whether the event is acknowledged.
  Note: Acknowledging an alert from the Operations Console does not affect this event value.
- **IsClearable**
  Defines whether the event can be cleared when an equivalent normal severity event is received.
- **IsCleared**
  Defines whether the event is currently cleared.
  Note: Clearing an alert from the Operations Console does not affect this event value.
- **Mapped Types**
  Defines a comma-separated list of types that identify the types in the domain manager whose instances are mapped when creating the USM instance.
- **Message**
  Defines a detailed description of the event.
- **MetricName**
  Defines an identifying name for a metric.
- **MetricDescription**
  Defines a description of a metric.
- **MetricType**
  Defines the metric type.
- **MetricUnitDefinition**
  Defines a unit of measure defined by the SI and IEC Technical Committee standards.
- **MetricDataType**
  Defines the data type of the metric.
- **MetricValue**
  Defines a value for a metric that crossed a threshold, or otherwise was the reason for the alert.
- **OriginApplication**
  Defines the name of the application where the alert originated.
- **OriginDnsName**
  Defines the fully qualified DNS name of the device where the alert originated.
- **OriginIPV4Address**
  Defines the IPv4 address of the device where the alert originated.
- **OriginIPV6Address**
  Defines the IPv6 address of the device where the alert originated.
- **RelatedAlerts**

Defines a semi-colon-separated list of related events, which are events resulting from the same root cause. Each related event is listed using the following format: MdrProduct,MdrProdInstance,MdrElementID.

- **RelatedIncident**
  Defines the Incident CI created for this event in the following format: MdrProduct,MdrProdInstance,MdrElementID.
- **RelatedIncidentURL**
  Defines the URL of the Incident CI created for this event.
- **RepeatCount**
  Defines the number of identical events occurring within a specific time defined by the ElapsedTime property.
- **RetireTimestamp**
  Defines when the event is no longer relevant. For example, a maintenance time may only be in effect for one hour. This property uses the xs:dateTime format: YYYY-MM-DDTHH:MM:SS.SSS-Z.
- **SeverityTrend**
  Defines the current trend toward more or less severity.
- **Tags**
  Defines a comma-separated list of alert classifiers that are useful for visualization or query.
- **TenantID**
  Defines a tenant identifier.
- **UrlParams**
  Defines a URL to open the domain manager from which the event originated.

Normalized and Raw Event Types

As an administrator, Event Management lets you interact with the following event types:

- **Normalized events**
  Normalized events are events that have been processed to use the alert properties defined in the USM schema. These events become CA SOI alerts unless you create a policy to manipulate or filter them.
- **Raw events**
  Raw events are records of normalized events that still use the properties of their event source. Normalization always occurs by default but is often too generic to be useful for raw event sources. Events from raw event sources such as SNMP traps or CA NSM Event Management require a user action to normalize them appropriately to USM alert properties. You can create normalization policy for raw events that map to USM properties to facilitate faster resolution when they become alerts.

The following connectors are examples that produce raw events with only generic normalization:

- SNMP connector
- Event connector
- IBM Tivoli NetCool connector

You can search for normalized and raw events. See the section Event Search Syntax Guidelines and Best Practices for information about the syntax rules.

Event Data Sources

Event data sources are connectors that are feeding events into the Event Store on each connector system. The available data sources are listed under Data Source in the Events tab of the Event Policy dialog. If a connector appears under Data Source, then you can run a search and deploy policy on its events. The data sources can be any of the following:

- **Connectors**
  Each CA Catalyst connector reporting to the SA Manager appears as an individual data source. Individual connector data sources display in the following format: *connectorname_domainserver@connectorserver*.
  - **connectorname**

  Defines the common name of the connector, such as Sample Connector.
  - **domainserver**
    Defines the host name of the system where the integrated domain manager is installed.
  - **connectorserver**
    Defines the host name of the system where the connector is installed.

  Connectors with multiple instances configured on the same system have a separate entry for each instance.
- **Event connector sources**
  The sources that are provided by the Event connector, such as the Windows Event Log, appear separately from one another in the following format: *adaptor_connectorserver@connectorserver*.
  - **adaptor**
    Defines the name of the Event connector source, such as MS-Syslog (Windows Event Log).
- **Mid-tier connector**
  The Mid-tier connector is automatically installed on the SA Manager and displays in the following format: Event Management MidTier Connector_*SAManagerserver@connectorserver*. All events are routed through the Mid-tier connec*tor* before appearing as alerts on the Operations Console. Search on the Mid-tier connector to search on events from all sources, and deploy event policies on the Mid-tier connector to perform actions on events from all sources.

Select a data source to view its details in the Details tab and automatically scope the Event Search tab to search on that source only. When you deploy a policy on a data source, the policy appears underneath that data source entry.

The data sources display depending on the status of the CA SAM Event Management service and the connector as follows:

- If the CA SAM Event Management service on the connector system is running, data sources on the system appear in green.
- If the CA SAM Event Management service on the connector system is not running, data sources on the system appear in red.
- If a connector is offline or removed, its data source disappears from the list.
- If a removed or offline connector still has searchable events in the Event Store, it remains on the list in green.

Run an Event Search

As an administrator, you create an event search, which is a detailed search for events that match simple or complex patterns. Event searches and subsequent event policies that are created by leveraging these searches are mechanisms that help you control how the product responds to important events or event patterns. You can run event searches:

- View matching events
- Save a policy and view the matching events at any time
- Create and deploy a policy that performs a specified action when matching events occur

You can federate event searches across all event data sources or scope them to one or more specific data sources. Time-based scoping can further reduce the target set of events. In addition to scoping, the search functionality lets you define the following to narrow your search:

- Complex patterns for up to three separate event types
- Operators to define the appropriate relationship between events
- A time interval to define the interval within which all matching events must occur
- Whether to search normalized or raw events
- A frequency to define how many times the event must occur within the time interval to match the pattern

Adhere to the following conventions to help ensure a successful search:

- The event properties that you use in searches must be valid USM alert properties as described in Event Properties and Event Information unless you are searching for raw events.
- The search patterns must follow the syntax rules that are described in Event Search Syntax Guidelines and Best Practices.
- This procedure contains instructions for searching normalized events. For more information about how to search for raw events, see Run a Raw Event Search.

Follow these steps:

1. From the Operations Console, select any item in the Navigation pane, and select Tools, Event Policies.
   The Event Policy dialog opens. The Events tab displays the available data sources and existing policies. The Event Search tab displays on the right pane for running searches.
2. Perform one of the following actions to scope your search:
   – If you are scoping to one data source, select it in the Data Source list on the Events tab.
     The source appears in the field next to the Source button in the Scope pane. To search all event sources, select the Mid-tier connector entry. The Mid-tier connector collects events from all connectors.
   – If you are scoping to more than one data source, click Source on the Event Search tab.
     The Select Data Source dialog opens with all available data sources.
3. (Multiple data sources only) Select any number of specific connector data sources to search on a subset of connectors.

   NOTE
   If you include all data sources or add the Mid-tier connector in a search that includes specific data sources, the search will return duplicate events, because the Mid-tier connector collects events from all connectors.

   The selected sources appear in the Source field.
4. (Optional) Click Time Range to narrow the time range of events to search.
   The Time Range for Event Search dialog opens.
5. (Optional) Select one of the following and click OK:
   – **Show items for a time range**
     Lets you define a specific time range to search down to the second. Use the Start and End fields to define the time range. Events are stored in files by the hour with the operating system last modified date determining whether a given hour's events fall within the time range. For example, if a steady stream of events has been flowing and stored for several hours, and it is now 6:35, a search from 5:30 to 6:30 would find all events stored from 5:00 to 6:00, and all events stored from 6:00 to the current time.
   – **Show items for the last N hours**
     Lets you search events that occurred within a specific number of hours from the current time. Use the arrows to specify the number of hours to search. Events are stored in files by the hour with the operating system last modified date determining whether a given hours events fall within the time range. For example, if a steady stream of events has been flowing and stored for several hours, and it is now 6:35, a search of the last 4 hours would find all events stored from 2:00 to the current time.

     NOTE

     If you restart the SA Manager or a connector, it places all events collected during the down time into a single file. Therefore, if this file falls into the time range that you specify, you may see events from that file that fall outside of the specified time range.
6. The time range appears in the Time Range field.
7. Enter a valid search pattern in the Event Pattern 1 field. Right-click the field for a list of valid properties, enumerated values, functions, and operators available for selection in a normalized event search.
   Each Pattern field represents criteria for one discrete type of event. Therefore, enter all necessary criteria for a single event type (using the necessary properties, functions, and operators) in one Pattern field.
   After you enter a search pattern in the Event Pattern 1 field, the second Event Pattern field becomes available.

NOTE

The names of the second and third Event Pattern fields vary depending on the criterion you select in the Additional Criterion pane. The names are sequential when you select 'ALL events occur within *N* seconds' and the same when you select 'ANY event occurs'.

8. (Optional) Enter a valid search pattern in the second Event Pattern field for a separate event type that you want to correlate with the first search pattern, and do the same in the third Event Pattern field if necessary.
9. Select Normalized Events in the Additional Criterion pane to search normalized events.

NOTE
For more information about raw event searches, see Run a Raw Event Search.

10. Select one of the following in the Additional Criterion section and click Search when finished:

NOTE
If you populated only the Event Pattern 1 field, the selected criterion does not influence the query results unless you select OCCURS *N* times within *N* seconds.

- **ANY event occurs**
  Returns any event that matches any of the patterns.
- **ALL events occur within *N* seconds**
  Returns a set of events that match all entered patterns that occur within the specified time interval of one another. For example, if you search for events of a certain severity in one pattern and events that meet a certain description in another pattern, the pair of events that match the patterns is returned if they occur within the specified time period. The results are grouped to indicate which events occurred together.
- **Sequence Enforced**
  Returns events that match all entered patterns that occur within the specified time interval and occur in the same order as the search patterns.
- **OCCURS *N* times within *N* seconds**
  (Single pattern only) Returns events that match the entered pattern and that occur the specified number of times or more within the specified time interval. For example, you can search for an event with a certain description that must appear four times within a minute to match the pattern. The results are grouped to indicate which events occurred together. If you select this option, all Event Pattern fields other than Event Pattern 1 are disabled.

NOTE

For example search patterns, see Event Search Examples: Time-Based Correlation, Event Search Examples: Occurrence Frequency, and Event Search Examples: Moving from Simple to Complex.

The results appear in the table above the Details tab. You can filter the results by entering a property value in the Filter field.
The results button to the upper left of the table indicates whether the search was successful, or if errors occurred. If the button is green, the search completed successfully and it can be deployed as a policy (see step 11). Yellow or red color can still mean the search returned a valid result but policy deployment is not possible. Click the button to view the returned error messages. If you receive an error message that you need help interpreting, see Error Messages.

NOTE
The time range search performs a search based on the r*eported time* of an event. The reported time is the time when the event is processed by a connector. The o*ccurrence time* is the time when an alert occurred. All processing is done on the raw alerts before they are added to the SA Store database tables for the best efficiency. The occurrence time (represented by the column Occurrence Time) and reported time (column Reported Time) are viewable in the query results window in the Event Policy UI. The occurrence time does not change. In many environments, connectors are restarted with active alarms and alerts are updated. As a result, events are generated for these alerts with more of a disparity between the occurrence time and reported time.

11. (Optional) Select a returned event.

The event properties and values appear in the Details tab. This tab shows many of the USM properties for the selected event and the following unique properties:

– **Group**
  Indicates the group number to which an event belongs. Grouping organizes events detected as part of a pattern so that you can see events in the context of the other events that triggered a pattern match. Grouping applies to time-based search types. Right-click the event results table and select Group to see how resultant events are grouped. Time-based searches have Group values starting with A and incrementing each time a group is detected. Non-time based searches all show the same Group value of A. A maximum of ten groups can appear.

  NOTE
  You can change the maximum number of returned groups. For more information, see Configure Event Search Settings.

  Frequency-based searches (OCCURS *N* times within *N* seconds) create groups based on the longest sequence within the timespan. For example, if you search for an event that must occur five times within 30 seconds, and the event occurs nine times within a 30-second window, the results organize all of these events into one group. Subsequent events matching the criteria in different 30-second windows would be organized into different groups. Searches using 'ALL events occur within *N* seconds' may cause events to appear multiple times if they are detected as being parts of multiple groups. For example, if you search for a combination of events occurring within 30 seconds, and each of the events occurs multiple times within a 30-second window, the results organize each unique pair into a separate group and display the events multiple times as a part of each group.

– **Pattern**
  Indicates the pattern that the selected event matched.

12. (Optional) Click Create Policy to save the search or create an event policy based on the search.

Run a Raw Event Search

Searching for raw events lets you view events with the internal properties from their source domain manager. All raw events also have a normalized version, but viewing the raw version of an event for a raw event source (such as a log file or event log) can help you better understand the event content. You can then use raw event search results to create an event policy with a normalization action to define rules for a more granular normalized event.

Follow these steps:

1. From the Operations Console, select any item in the Navigation pane, and select Tools, Event Policies.
   The Event Policy dialog opens. The Events tab displays the available data sources and existing policies. The Event Search tab displays on the right pane for running searches.
2. Select the source on which to search in the Data Source list on the Events tab.
   The selected source appears in the field next to the Source button in the Scope pane.
   Limit raw event searches to one data source, because this typically provides results that are best suited for normalization. The Select Data Source dialog prevents you from selecting multiple sources when Raw Events is selected.
3. Select Raw Events in the Additional Criterion pane.
   All other criteria in the pane are disabled. Raw event searches support a single pattern search with no occurrence criteria.
4. Enter a valid search pattern in the Event Pattern 1 field. Complete the search pattern using properties from the source domain manager instead of USM alert properties. The right-click menu options for these properties are disabled when you select Raw Events. Running a basic raw event search populates the right-click menu with the properties returned by the search. You can also still use the right-click menu options for operators, functions, and connections.
   Enter all search criteria in the Event Pattern 1 field. Raw event searches support multiple conditions in the Event Pattern 1 field, but they do not support criteria in multiple Event Pattern fields. Any criteria in other fields is combined with the Pattern 1 field.
5. Click Search when finished.

The results appear in the table above the Details tab. You can filter the results by entering a property value in the Filter field.
The results button to the upper left of the table indicates whether the search was successful, or if errors occurred. If the button is green, the search completed successfully and it can be deployed as a policy (see step 7). Yellow or red color can still mean the search returned a valid result but policy deployment is not possible. Click the button to view the returned error messages. If you receive an error message that you need help interpreting, see Error Messages.

6. (Optional) Select a returned event.
   All raw event properties and values for the event appear in the Details tab.

   NOTE
   Other properties may appear that are not true raw event properties. For help distinguishing true raw properties from temporary or internal properties that are also returned by raw event searches, see Raw Event Properties in Normalization Actions.

7. (Optional) Click Map Events to save the search or create an event policy with a normalization action based on the search.

Event Search Syntax Guidelines and Best Practices

Contents

Event Management uses the predicate expression syntax that is defined in the XPath 2.0 and XQuery 1.0 XML language specifications as the standard syntax for event searches. Searches must adhere to the predicate expression syntax to work.

NOTE
The Event Management engine supports conversion of only certain XPath functions for use in real-time policies, as described in the Functions section.

When constructing your search patterns, take advantage of the right-click menu, which helps you format the syntax correctly.

The following sections provide guidelines to help ensure that your search is successful:

- Event Properties and Values
- Operators and Expressions
- Functions
- Special Characters
- Scoping
- Raw Event Searches Limitations
- Normalized Event Searches Limitations

Event Properties and Values

- Use only the valid properties and property values defined in Event Properties and Event Information for normalized event searches. Use the right-click menu to automatically populate pattern fields with the appropriate property names and enumerated values.
- All event properties and values are case-sensitive. Using the right-click menu helps ensure the correct case and format for properties and enumerated values.

Syntax

Use the following syntax to complete a basic search pattern:

```
property[operator]'value'
```

NOTE
For deployed policies and when performing searches using string values, property values must be delimited by single quotes. An error message appears when a property is missing a quote character on either side. For example, to enter a pattern that returns events with a severity of Critical, you would enter Severity='Critical'.

Alternatively, leave the search patterns empty to return a console view of all events for the defined scope.

Event Properties

- Performing actions on alerts in the Operations Console does not affect related event properties, such as IsAcknowledged.
- The MdrProduct, MdrProdInstance, and OccurrenceTimestamp properties are automatically leveraged for scoping. Therefore, using these properties in event searches is typically redundant and unnecessary.
- The ID properties (MdrElementID and AlertedMdrElementID) and the properties that use the ID properties as part of their values are unique values that are difficult to derive without looking directly in the Event Store. Typically, the best use of these properties in searches is through the question mark (?) substitution character described in the Special Characters section.
- Always use the right-click menu when entering a value for the AlertedMdrProduct or MdrProduct properties (not available for raw events). The right-click menu converts the displayed connector name values into the valid numeric values for these properties defined in the USM schema. Entering a connector name in an event pattern instead of the numeric value causes the search to fail.

Operations and Expressions

Use any of the operators listed under Properties, Operators in the right-click menu to define how a property relates to the specified value.

You can use the conditional operators AND and OR to enter complex patterns for a single event type. This example returns all events occurring on server1.ca.com with a severity of Critical:

```
Severity='Critical' and MdrProdInstance='server1.ca.com'
```

All operators are case-sensitive. Using the right-click menu ensures the correct case and format for operators.

Functions

This section lists some best practices for the 'not', 'matches', and 'fn:Parse()' functions.

Not

- Use the 'not' function to return events that do not match the entered pattern. This example returns all events that have a severity other than Critical:
  ```
  not(matches(Severity,'Critical')
  ```
- You can also use the 'not' function to detect a missing event. This example, when paired with a pattern detecting a 'backup started' event, detects when the expected 'backup stopped' event did not occur:
  ```
  not(matches(Summary,'backup stopped'))
  ```

For more information about detecting a missing event, see Event Search Examples: Advanced Search Techniques.

Matches

- Use the 'matches' function to return events based on partial match criteria. Use regular expressions within this function to construct the match criteria. This example returns events with a severity that starts with M and a message that contains the phrase 'service has stopped':
  ```
  matches(Severity,'^M.*') and matches(Message,'service has stopped')
  ```
- Use the 'not' and 'matches' functions in combination with one another to return events that do not match partial criteria. This example returns events with a severity that does not start with M:

```
not(matches(Severity,'M.*'))
```

fn:Parse()

- Correlate events based on property fragments using the fn:Parse() function. This example returns events with the server name server1 in a specific place in the event Message property:
  ```
  fn:Parse(Message,'device=(.*?).ca.com')='server1'
  ```
- Use regular expressions to construct the parsing function.

For more information about parsing property values to use fragments in event searches, see Event Search Examples: Advanced Search Techniques.

Special Characters

This section lists some best practices for special characters.

Question Mark

- Use a question mark (?) to denote that you want a property value to be the same across event types. This example returns correlated events that have the same AlertedMdrElementID value and summaries that contain 'service has started' and 'service has stopped':
  ```
  AlertedMdrElementID=? and matches(Summary,'service has started')
  AlertedMdrElementID=? and matches(Summary,'service has stopped')
  ```
  The results are organized into groups based on their correlated value.
- Only use the question mark character for correlation-based searches where each event pattern contains another condition connected with the AND operator (as in the preceding example). Using the character under any other conditions may return an error or unusual results.
- You can use multiple question mark characters to create multiple correlated conditions in a single event search.

Exclamation Mark

- Use an exclamation mark (!) to denote that you want a property to not match the specified value. This example returns events with all Alert Types other than Informational:
  ```
  AlertType!='Informational'
  ```
- Use the ! character for simple expressions and the 'not' function for more complex criteria, such as a multi-condition expression.

Scoping

Select the source in the Data Source list on the Events tab to scope to one data source. Click the Source button to select multiple sources for scoping.

A search using the default scoping (asterisk in the Source field), which includes all data sources, returns duplicate events. To search events from all sources, perform a search scoped to the Mid-tier connector.

Exclude the Mid-tier connector from scoping when searching events from specific sources.

The search scope must be limited to 25,000 events per connector. If any included data source exceeds this limit, an error message appears. Reduce the time interval so that the returned number of events is within the limit for the search to succeed.

Raw Event Search Limitations

The following table shows the operators and functions that you can use for raw event searches and which of them will work in a normalization policy.

	Search	Normalization Policy
Operators	All	Only AND, =, !=
'fn:Parse' function	Yes	No
'not' function	Yes	Yes

The following three examples show the only raw event search patterns that are supported for normalization policy deployment:

```
Severity='Critical' and AlertType='Health'
matches(Severity,'Critical|Fatal')
Severity='Critical' and not(SeverityTrend='Unknown')
```

Note: The use of the not function has the following limitations:

- The not function does not support two operands. For example, not(mdr_dept='Finance' and mdr_size='11') is not supported. However, not(mdr_dept='Finance') and not(mdr_size='11') is supported.
- Additionally, parentheses do not support two operands even if the 'not' function is not present. For example, (mdr_dept='Finance' and mdr_size='11') displays an error message "Unable to Resolve". However, mdr_dept='Finance' and mdr_size='11' is supported.

When constructing search patterns, consider the following:

- If you do not have a list of the internal properties for the domain manager, run a search for all raw events from the data source. Then you can select the raw event properties returned by the search for use in search patterns using the right-click menu. For help with distinguishing true raw properties from temporary or internal properties that are also returned by raw event searches, see Raw Event Properties in Normalization Actions.
- Raw event properties are also case-sensitive in searches.
- Raw event searches only support a single event pattern with no additional time or occurrence-based criteria. Enter all search criteria in the Event Pattern 1 field.

Normalized Event Search Limitations

The following table shows the operators and functions that you can use for normalized event searches and which of them will work in a deployed policy.

	Search	Deployed Policy
Operators	All	All except >, <, >=, <=
'not' function	Yes	No
'fn:Parse' function	Yes	Yes
nested or embedded functions	Yes	No
'contains', 'ends-with', and 'starts-with' functions	Yes	No

Note:

- The not function does not support two operands. For example, not(mdr_dept='Finance' and mdr_size='11') is not supported. However, not (matches(Severity,'Major')) and not (matches(Assignee,'Smith')) is supported.
- A limit of ten fn:Parse statements can appear in a deployed event policy.

Configure Event Search Settings

You can configure whether to search archived Event Store files and how many groups to return in search results.

Follow these steps:

1. Open the <SOI_HOME>/tomcat/lib/eventManagerClientConfig.xml file and change the value of the following property if necessary:
   – **EQUERY_UNZIP_ARCHIVE=**
     Determines whether to unzip and search archived Event Store files for event searches that are not time scoped. Enter 1 to search archived files or 0 to never search archived files.
     Default: 0
2. Open the <SOI_HOME>/jsw/conf/EventManager-wrapper.conf file in a text editor and change the value of the following properties if necessary:
   – **EQUERY_MAX_QGROUP=**
     Determines the maximum number of groups to return in search results.
     Default: 10
   – **EQUERY_MAX_RETCOUNT=**
     Defines the maximum amount of events returned by a search per connector multiplied by 1000. For example, the default value of 25 returns a maximum of 25,000 events from a search on a connector.
     Default: 25
3. Restart the CA SAM Application Server service.
   Subsequent event searches use the configured settings.

Event Search Examples

This section provides examples for common event search types.

Event Search Examples: Time-Based Correlation

Select 'ALL events occur within *N* seconds' in the Additional Criterion pane and enter search criteria in at least two of the Event Pattern fields to create a search that correlates and groups sets of events according to their occurrence within a specified time interval. Select the optional Sequence enforced check box to match only on event groups that occur in the order of the event patterns.

Example: Detect increase in severity with sequence enforced

Pattern 1:

```
AlertedMdrElementID=? and matches (Severity,'^M.*')
```

Pattern 2:

```
AlertedMdrElementID=? and (Severity='Critical' or Severity='Fatal')
```

This example searches for a combination of events that have the same AlertedMdrElementID, which have therefore been generated from the same connector on the same CI. The first event must have a severity that starts with M, which would be minor or major. The second event must have a severity of critical or fatal. For Additional Criterion, select 'ALL events occur within 600 seconds' and select Sequence enforced. This search detects when the severity of an event on the same CI has increased from a previous event within the last ten minutes.

Example: Correlate service shutdown with sequence enforced

Pattern 1:

```
AlertedMdrElementID=? and matches (Summary,'service has started')
```

Pattern 2:

```
AlertedMdrElementID=? and matches (Summary,'service has stopped')
```

This example searches for a combination of events that have the same AlertedMdrElementID, which have therefore been generated from the same connector on the same CI. The first event must contain 'service has started' in its summary, and the second event must contain 'service has stopped' in its summary. For Additional Criterion, select 'ALL events occur within 30 seconds' and select Sequence enforced. This search detects a service that is crashing every time it starts. You could scope this search to the Mid-tier connector to search all events or on a subset of connectors for a targeted search.

> **NOTE**
> For an end-to-end scenario using this search pattern, see Event Management Scenarios.

Event Search Examples: Occurrence Frequency

Select 'OCCURS *N* times within *N* seconds' and enter search criteria in one event pattern field to create a search that returns a matching event that occurs a specified number of times within a specified time interval.

Example: Detect CPU usage spikes

```
matches (Message,'server1') and matches (Summary,'CPU usage high')
```

This example searches for events that contain the same server name in their message and a summary that contains the text 'CPU usage high'. For Additional Criterion, select 'OCCURS 3 times within 180 seconds'. This search can detect whether CPU usage is spiking every two minutes on a CI.

Example: Detect unacknowledged authentication failures

```
isAcknowledged='false' and not (Severity='Informational' and Severity='Normal') and matches
 (Summary,'Authentication failure')
```

This example searches for unacknowledged events with a severity higher than Normal that contain the text 'Authentication failure' in their summary. For Additional Criterion, select 'OCCURS 4 times within 60 seconds'. When scoped to a connector that tracks enterprise security (such as CA Access Control, or the Windows Event Log through the Event connector), this search can detect repeated attempts to breach system security by an unauthorized user.

Event Search Examples: Advanced Search Techniques

These examples show advanced techniques for performing complex correlation that are based on property fragments and for detecting the absence of an expected event.

Example: Detect an event on a specific system by correlating based on property fragments

Pattern 1:

```
fn:Parse(Message,'device=(.*?).ca.com')='server1' and fn:Parse(Summary,'Database Instance:(.*?)
 stopped')='PAYROLL'
```

This search pattern isolates key values embedded in the event Message and Summary properties:

- The server name of the device in the Message property when the Message matches 'device=*servername*.ca.com'
- The database instance name in the Summary property when the Summary matches 'Database Instance: *instancename* stopped'

When these values match server1 and PAYROLL, the event matches. This example shows how you can parse information out of a property and can use that information in search patterns.

Example: Correlate events on the same system that are based on property fragments

Pattern 1:

```
fn:Parse(Message,'device=(.*)')=? and Summary='low memory'
```

Pattern 2:

```
fn:Parse(Message,'device=(.*)')=? and Summary='device unresponsive'
```

This example uses the question mark correlation character and the fn:Parse function to correlate events that have the same server name in the Message property when the property format matches 'device=*servername*'. Events with a Summary of 'low memory' and 'device unresponsive' occurring within a short time interval could indicate low memory as a root cause of device failure.

> **NOTE**
> For a full example using this search pattern, see Create Event Action Examples.

Event Search Examples: Raw Events

Select Raw Events and enter a single search pattern in the Event Pattern 1 field. This pattern searches for raw events that retain the properties of their event source.

Example: Search for Windows Event Log events from the Security log

```
syslog_source='Security'
```

The Windows Event Log contains multiple logs that collect different types of events, such as Security, System, and Application. Normalized Windows Event Log events give no indication of the source event log. This example isolates this information in the raw event property syslog_source and returns all events in the Security log. You can then create a policy that normalizes all security events from the Windows Event Log.

Example: Search for traps with specific information in the variable bindings

Events from the SNMP connector or the SNMP adaptor that are provided with the Event connector have their variable bindings split into separate properties in the Event Store. Therefore, you can search based on a specific varbind value. Varbind properties are prefixed with 'varbind-' and then the OID number. The following example searches for CA Workload Automation traps with a specific job name:

```
snmp_enterprise="1.3.6.1.4.1.11203" and varbind-1.3.6.1.4.1.11203.9="Disk Mount Job"
```

This pattern first searches for traps with an enterprise OID of 1.3.6.1.4.1.11203, which narrows the results to CA Workload Automation traps. The pattern then searches for events that match Disk Mount Job in the variable binding 1.3.6.1.4.1.11203.9, which contains the CA Workload Automation job name. You can use this search to view all events that are related to that job. You can create policy that normalizes the messages that are related to the job so that they appear as alerts in the Operations Console. The varbinds are mapped to the appropriate properties.

Event Search Examples: Moving from Simple to Complex

The following example scenario shows how you can move from a simple search to more complex occurrence and correlation searches. You can use the searches in event policies.

Consider a situation where a database server is having performance problems. Several domain managers are monitoring the server, and you cannot pinpoint the cause of the problems. You could begin with the following simple search in the Event Pattern 1 field with 'ANY event occurs' selected:

```
matches (Summary,'query error')
```

This search returns all events that contain the phrase 'query error' in the event summary. The search shows many failed database queries coming from various connectors managing resources that are querying the database. To refine the search to query failures on the problematic database server, you could run an additional search as follows:

```
matches (Summary,'dbserver1') and matches (Summary,'query error')
```

This search only returns database query failure events that include the name of the problematic database server in the event summary. If you still see many events that are returned, you can select OCCURS and specify 10 times within 60 seconds. This search returns matching events that occurred ten times within one minute of each other, which indicates that the query failures are persistent. If you suspect that persistent query failures are draining the database server memory, you can further refine the search by entering the following two search patterns:

```
matches (Summary,'dbserver1') and matches (Summary,'query error')
matches (Summary,'dbserver1') and matches (Summary,'memory usage high')
```

Selecting 'ALL events occur within 120 seconds' returns sets of events where at least one query failure occurred and a high memory usage event occurred within two minutes of each other. The results of this search could indicate a correlation between persistent database query failures and high memory usage on the database server, which could be degrading its performance. With this knowledge, you can generate a plan of action for fixing the problem and future occurrences of the same problem by doing any or all of the following:

- Including the search patterns in a create event policy that could raise the severity of the query error events and modify the message to describe the correlated condition
- Filtering repeated database query error events so that technicians can focus on the created event with elevated severity
- Creating escalation policy based on the new event that emails the database server technician or the technician responsible for the application sending the bad database queries

Working with Event Policies and Actions

This section describes how to create all types of event policies and create manually policies for scenarios not covered out of the box.

Event Policy with Actions

Contents

An *event policy* is a combination of event search patterns and an action to perform when the patterns match. You can do the following with event policies:

- Save them for an on demand view of events that match the search patterns
- Deploy them to evaluate incoming events dynamically based on the search criteria and perform the specified action in response to matches

Event policies let you manage when and how events become alerts in CA SOI. The following action types are available:

- **Filter**
  Excludes events that match search patterns from becoming alerts. For example, you can discard all events with a severity that is less severe than Major so that only events with high severities appear as alerts. You can also explicitly 'include' matching events rather than discarding them.
- **Create new event**
  Creates a new event when a match occurs. You specify all property values for the new event, which can be custom values or based on values in the matching events. For example, you can create a new event based on a correlated set of events that, when occurring together, indicates a more severe problem.
- **Enrichment**
  Adds information to an event from outside sources when a match occurs. For example, you can add contact information to events from an external database or use the Map only feature and add static information.
- **Normalization**
  Configures custom mappings from raw event properties to USM alert properties. For example, you can normalize SNMP traps from a specific source so that their variable bindings map to their appropriate USM properties.

Event policy helps ensure that as events become alerts that appear in the Operations Console, they represent a consolidated, high quality, actionable set of conditions.

Event Policy Best Practices

To decide which event action use in a policy, use the following guidelines:

- Use an isolated filter action when filtering events not associated with another action. If a create event action requires you to filter the original events, create a filter action that filters based on the same search criteria.
- Use a create event action when multiple events correlate to require a separate event message summarizing the correlated condition. If one or two properties in a single event require more or different information, use enrichment or normalization instead.
- Enrichment and normalization actions both let you modify event property values.
  Use enrichment in the following situations:
  - When an event requires extra information in a property available for enrichment. Most optional properties are available for use in enrichments. Required properties are not supported for enrichments in the user interface. The enrichment source can return an invalid required property value and can break the policy.
  - When an event requires extra information stored in some other external source. Only enrichments can extract information from external sources. Assign the enrichment information to one of the properties that the enrichment action supports.

  Use normalization in the following situations:
  - When a required event property requires new or more information. Any information that you enter in a normalization action overwrites the mapping for that property in the default policy. Normalization actions enforce valid values for enumerated properties and values for all required properties. You can then interact with these properties with less potential for error.
  - When events from a raw event source require more detailed normalization rules to become manageable alerts.

To ensure a valid policy action, use the following best practices:

- Run an event search before creating any non-filter policy, so that the search results are available in the Event Log table for previewing changes.
- Verify that search patterns are valid and provide expected results before creating policy based on them.
- Test all enrichment connection information before assigning enrichment values.
- Do not use the &, <, and ' characters on the Create New Event, Normalize Event, and Enrichment Policy pages.
- Read the Confirm page carefully before finalizing the policy to verify that all settings are correct.
- CA Catalyst connectors do not support database enrichments. Deploy database enrichments on the Mid-tier connector to enrichment CA Catalyst connectors with database information.

Event Policy Deployment

Contents

When you create an event policy, you deploy the policy for the specified event action to occur when the search patterns match. You can deploy policies on an individual connector, a subset of connectors, or on the Mid-tier connector. The deployed policy dynamically merges with the appropriate connector policy to detect events matching the search criteria in real-time and perform the resultant actions.

Mid-tier Connector

The *Mid-tier connector* is an intermediate processing layer through which you can deploy cross-domain event policy for automated processing on events from all connectors. The Mid-tier connector is automatically installed on the SA Manager system and appears in the Data Source list of the Event Policy dialog Events tab as follows:

```
Event Management MidTier Connector_SAManagerserver@connectorserver
```

All events flow through the Mid-tier connector before reaching the Operations Console as alerts.

Deploy event policies to the Mid-tier connector in the following situations:

- You want to correlate events across domain managers. When you deploy policy on a set of connectors, correlation occurs only within each connector source. The Mid-tier connector, due to its position in the event flow, can correlate across all data sources.
- You want to perform actions on events from all domain managers. The Mid-tier connector performs actions on matching events from all data sources by default when you deploy policy to it.
- You want to perform an out of the box enrichment. These enrichments are only supported on the Mid-tier connector.
- You want to perform an enrichment that you cannot perform directly on CA Catalyst connectors, such as JDBC enrichment.

Deployment Best Practices

Use the following guidelines to understand when to deploy policies on the Mid-tier connector or on a single connector or subset of connectors:

- Use the Mid-tier connector any time the situations previously described occur: you want to correlate across data sources, or you want to perform actions on all data sources. For action processing, using the Mid-tier connector is a more efficient operation than deploying a policy to all connectors, because the processing occurs in one location.
- Do not use the Mid-tier connector in most cases if you are limiting the matching events to those from a subset of connectors (unless you require cross-domain correlation). By rule, always deploy policies at the lowest level possible, which can be for one specific connector or a subset of connectors.
- Do not deploy a policy on a subset of connectors and the Mid-tier connector, because this creates a duplicate action in most cases.
- Deploy a policy with a normalization action on one connector only. Normalization actions do not support deployment on multiple connectors or the Mid-tier connector.
- Deploy out of the box enrichments (CA Spectrum and CA CMDB) on the Mid-tier connector only. Other connectors do not support these enrichments.
- If you want to perform a JDBC enrichment on CA Catalyst connectors, deploy it on the Mid-tier connector. CA Catalyst connectors do not support direct JDBC enrichment deployment.

Create an Event Policy with a Filter Action

Contents

You can create and deploy an event policy that filters normalized or raw events that match the search patterns. The filter action can do either of the following:

- Discard events that match the pattern to prevent them from appearing as alerts in the Operations Console
- Immediately include matching events (without regard for subsequent exclude filters) as alerts in the Operations Console

Combinations of filter actions are necessary in situations where you want to discard most events from any or all data sources but explicitly include a small portion of those events. However, the user interface does not support multiple filters in one policy. Built-in sequencing rules handle the most common filter combination use cases, but for some use cases, you must manually refine the policy to configure the appropriate sequence of filter evaluations.

> **NOTE**
> For an example end-to-end deployment scenario using a filter action, see Event Management Scenarios. For an example of an event policy manually refined to include a combination of include and exclude filters not supported by the default filter sequencing rules, see Manually Refining Event Policy.

Follow these steps:

1. From the Operations Console, select any item in the Navigation pane, and select Tools, Event Policies.
   The Event Policy dialog opens.
2. Run an event search with the patterns and criteria that you want to use for the policy.

   > **NOTE**
   > The Scope options do not apply to event policy.

   The search results display.
3. Click Create Policy.
   The Create Event Policy wizard opens and displays the New Policy page.

   > **NOTE**
   > Only one user can create an event policy at one time. If any other logged in user has the Create Event Policy wizard open, an error message appears saying that another user has a lock on the functionality. The user must close the dialog before you can create event policies.

   If the search patterns would cause an event policy deployment error, a corresponding error message appears that might prevent you from creating the policy. For more information, see Error Messages.
4. Enter a name in the Policy Name field, select Filter Events and one of the following subselections, and click Next:
   - **Exclude**
     Excludes events matching the search pattern from appearing as alerts. Use this operation to discard events that do not require management as alerts.
     An exclude filter does not prevent matching events from being evaluated as a part of other policies. For example, another policy could detect a pattern where five logins failed within a certain time period, even if those events meet the criteria for an exclude filter.
     In cases where more than one filter is deployed to a connector, the filters are sequenced according to default sequencing rules.
   - **Include**
     Includes matching events as alerts in the Operations Console and ignores subsequent exclude filters on those events. The include operation is useful if you have subsequent filter policies that discard all events or all events of a certain type, and you want to manage only certain events as alerts. By default, include filters take precedence over exclude filters deployed on the same connector.

     > **NOTE**
     > The policy name cannot have more than 128 characters and cannot contain any characters listed as prohibited on the tooltip that appears when you hover over the Policy Name text.

   The Select Data Sources page opens.
5. Do one of the following:
   - Select Save policy only and click Finish.
     The policy is saved but not deployed, and it appears in the Saved Policies section of the Events tab. This policy does not dynamically evaluate and process events according to its defined search patterns and resultant actions until you deploy it. Skip the rest of this procedure if you want to save the policy without deploying.
   - Select Save and Deploy policy.
     The list of available connectors becomes available.
6. Select the connectors on which to deploy the policy according to the guidelines in Deployment Best Practices, move them to the Selected Data Sources pane, and click Next.
   The Confirm page opens and displays the following information:
   - Data sources on which to deploy the policy
   - Time scope
   - Search patterns for policy criteria
   - Event action type and action details
7. Click Finish.

The policy is created, and it appears in the Deployed Policies section of the Events tab and under the appropriate data source in the Data Source section.

Filter Action Examples

Low-level event sources, such as the Windows Event Log in these examples, can generate informational events. Depending on their severity, these events could become alerts in CA SOI, even though they have no impact on services or managed CIs. These examples show how you can use filter policies to control which events from a raw event source such as the Windows Event Log become alerts.

Example: Exclude login failures with non-normal severities

This example discards login failure events with a severity other than normal so that they do not become alerts:

- Enter the following search pattern in the Event Pattern 1 field:
  ```
  matches(Message,'User .* login failed.') and Severity!='Normal'
  ```
  This search pattern matches events that contain the message 'User * login failed' (with the asterisk denoting that any user name value is acceptable) and have a severity other than normal.
- Select Filter Events and Exclude on the New Policy page of the Create Event Policy dialog.
- Deploy the policy on the Windows Event Log data source.
  All events with a severity of normal and informational are automatically prevented from becoming alerts. This policy further excludes Windows Event Log events that are informational in nature but have a misleading severity (such as minor) that would otherwise become alerts.

An exclude filter does not prevent the events from being evaluated as a part of other policies. For example, another policy could detect a pattern where five logins failed within a certain time period, even if those events meet the criteria for an exclude filter.

Example: Include login failure events for a specific user

You typically use include filters in combination with an exclude filter to explicitly include certain events that would otherwise be discarded by an exclude filter. This example builds on the previous example by including login failure events for a specific user:

- Enter the following search pattern in the Event Pattern 1 field:
  ```
  Message='User Jeff login failed'
  ```
  This search pattern matches events that contain the message 'User Jeff login failed'. It isolates failure events for the user Jeff.
- Select Filter Events and Include on the New Policy page of the Create Event Policy dialog.
- Deploy the policy on the Windows Event Log data source.
  This policy explicitly includes Windows Event Log events with the message text 'User Jeff login failed' so that they become alerts. Without the include filter, these events would be discarded by the previously defined exclude filter.

Include filters let you specify the specific events that you want to become alerts, regardless of the existence of a matching exclude filter. You could take this example further for a low-level event source like the Windows Event Log by creating an exclude filter for all events from the source, and then using include filters to include exceptions that you want to manage as alerts.

You could deploy an event policy for each of these filters on the same connector. According to the default sequencing rules, Event Management evaluates the include filter first and the exclude filter second, so that the exclude filter does not exclude important events before the include filter detects them.

> **NOTE**
> The alert management engine cannot display alerts with a severity value of normal or informational on the Operations Console. Therefore, even if you create an include filter action for events with normal or informational severities, these events do not appear as alerts on the Operations Console.

To deploy filter combinations on the same connector that would not work using the default sequencing rules, manually refine the policy by adding a seqnumber attribute to configure the appropriate sequence of filter evaluations. For an example of how to manually sequence filter combinations, see Manual Policy Scenario: Sequencing Exclude and Include Filter Combinations.

Filter Sequencing Rules

Event policies support one filter action each. However, most filtering use cases require a combination of multiple filters to exclude certain events and include only an important subset of those events. Combining multiple filters requires a separate event policy deployment for each filter. Event Management uses the following default sequencing rules to determine in what order to evaluate multiple filters that are deployed on the same connector:

- Include filters are always evaluated before exclude filters.
- Multiple include filters on the same connector are evaluated in random order.
- Multiple exclude filters on the same connector are evaluated in random order.

After an event matches a filter, subsequent filters do not evaluate the event, which is why the sequence must be correct. The default rules enable support for the most common filter combination use cases in the user interface. For example, if you want to exclude all events from the SNMP connector except for events from a specific data source, the following process occurs:

- You deploy the include and exclude filter policies on the SNMP connector in any order.
- The event policy evaluates the include filter first and includes any events from the specified data source. The included events are not evaluated by the exclude filter, and therefore are not excluded incorrectly.
- The event policy evaluates the exclude filter for events that did not match the include filter and excludes any matching events.

The default sequencing rules do not support all potential use cases. The following filter combinations on the same connector require manual policy refinements to change the filter evaluation sequence:

- You want an exclude filter to take precedence over an include filter.
- You want a specific sequence of multiple exclude or include filters.

Change Filter Evaluation Sequence

When you deploy multiple event policies with filter actions on the same connector, you can manually modify the filter evaluation sequence when any of the following situations apply:

- An exclude filter requires evaluation before an include filter
- More than two filters exist of different types that require sequencing that differs from the default sequencing rules

For example, consider a situation where you define the following filters on the same connector:

- A broad exclude filter
- An include filter that includes an important subset of events that would match the exclude filter
- Another exclude filter that excludes a subset of the events that would match the include filter

These filters would require evaluation from the most granular (the specific exclude filter) to the least granular (the broad exclude filter) to help ensure that the policy does not erroneously exclude or include any events. Because the default sequencing rules always evaluate include filters first, this example would require you to change the filter evaluation sequence manually.

> **NOTE**
> For a detailed manual filter sequencing scenario, see Manual Policy Scenario: Sequencing Exclude and Include Filter Combinations.

Follow these steps:

1. Deploy the appropriate filter policies to the same connector.
2. Navigate to the files on the connector system where you deployed the policies, and back up all policy files that require manual edits (<SOI_HOME>\resources\Core\Catalogpolicy\extensions).
3. Open the policy file for the filter that you want the policy to evaluate first, add the property seqnumber='1' to the Field attribute in the <FilterPostN> section as shown in the following example, and save the file:

   ```
   <FilterPostN>
       <Field input='internal_suppresseventExclude Filter 1' pattern='^true$' type='exclude' seqnumber='1' />
   </FilterPostN>
   ```

   The policy evaluates the filter with the lowest seqnumber value first. Any filter with a seqnumber property takes precedence over filters without one.
4. Copy the edited policy file to the <SOI_HOME>\resources\EventManagement\externalPolicies directory on the SA Manager system, change the file extension from '.xml' to '.policy', and change the file name so that it matches the name of the corresponding file in the <SOI_HOME>\resources\EventManagement\Policies directory.
   The policy now appears under External Policies in the Events tab. When you edit a policy file and move the corresponding SA Manager record of that policy to the externalPolicies directory, it appears as an external policy in the user interface.
5. Right click the policy, select Deploy Policy, select the connectors on which to deploy, and click OK.
   The updated policy redeploys.
6. Repeat Steps 2-5 on subsequent policies if necessary, adding seqnumber properties to each file to configure the correct evaluation order.
   The sequencing change takes effect.

Filtering Original Events

When generating a new event from an incoming event, you can decide whether you want to filter the original events. You can do so by using a configuration setting, CLEAR_ALERT. By filtering original events, you remove the excessive number of extraneous events that you do not need in your infrastructure. This ability helps you create a more organized and efficient event management process in your organization.

The filtering of the original events functionality supports both individual and multiple incoming events participating in a rule. OR, AND, and frequency threshold rules are also supported. The following examples help you understand how specific rules are supported in filtering original events when multiple incoming events participate in a rule:

- Consider a scenario where the policy condition specifies if an event A or an event B occurs, you want to create an event C. This scenario represents the OR rule example. The original events A and B are cleared after the event C is generated.
- Consider a scenario where the policy condition specifies if an event A and an event B occur within 5 minutes, you want to create an event C. This scenario represents the AND rule example. The original events A and B are cleared after the event C is generated.
- Consider a scenario where the policy condition specifies if an event A occurs three times within 5 minutes, you want to create an event C. This scenario represents the frequency threshold rules example. All original A events are cleared after the event C is generated.

Set Additional Fields for Cleared Events

If filtering of events (that create new events) is enabled, all events that trigger rules are filtered and are not published. For example, in case of events participating in OR or ANY pattern, all events trigger the rule, so these events are filtered. However, multiple events that pass through the system unfiltered (for example, events participating in the AND or OCCURS pattern) do not trigger the rule and are published. These published events (that had already been sent to CA SOI) are *cleared* by creating a similar event with the severity of Normal. For these cleared events, you can also set more values using the FormatPostN section in the policy file. Only in case of the AND pattern and the OCCURS pattern that the clear events are created.

Using this functionality, you can, therefore, also filter events that are based on the field values. You manually update the event policy file and specify the value for the additional fields. You specify the value for the field (for example, userAttribute2) in the FormatPostN section under the createEvent section of the policy file.

Follow these steps:

1. Open the event policy file (<SOI_HOME>\resources\EventManagement\Policies*policyname*.policy) in a text editor.
2. Locate the createEvent section in the file.
3. Change the value of the field in the FormatPostN section.
   An example of the snippet that contains the additional field userAttribute2 is as follows:
   ```
   <EventClass name='<policyname>createevent_suppression' extends='<deployedeventclass>'>
     <FormatPostN>
       <Field output='userAttribute2' format='userassignedvalue' input='' />
     </FormatPostN>
   </EventClass>
   ```
4. Restart the CA SAM Integration Services service.
   The value is set.

Specify the Filtering Original Events Configuration Setting

You can use the CLEAR_ALERT configuration setting to specify whether you want to filter original events when a new event is created.

Follow these steps:

1. Navigate to the <SOI_HOME>\jsw\conf folder and locate the SAM-IntegrationServices.conf file.
2. Find the line that contains the CLEAR_ALERT configuration setting; for example, wrapper.java.additional.<*number*>=-DCLEAR_ALERT=true.
3. Set the value of the configuration setting as true or false:
   - **true**
     Specifies that the original events are filtered.
   - **false**
     Specifies that the original events are not filtered. This value is the default value.
4. Restart the CA SAM Integration Services service.
   The value for the CLEAR_ALERT configuration is set.

Create an Event Policy with a Create Event Action

Contents

You can create and deploy an event policy that creates a new event to respond to a condition identified by events that match a defined search pattern. Creating a new event can be useful in the following situations:

- You are receiving multiple events indicating the same condition and want to create one representative event to display as an alert while filtering out the duplicates
- Multiple event conditions are indicative of a different or more serious condition that you want to capture in a new event

It is often sensible to use the create event action in combination with other actions, primarily a filter. For example, creating a new event may justify discarding the original events that matched the search pattern. To do so, create a separate filter action using the same search criteria.

> **NOTE**
> For an example end-to-end deployment scenario using a create event action, see Event Management Scenarios.

Follow these steps:

1. From the Operations Console, select any item in the Navigation pane, and select Tools, Event Policies.
   The Event Policy dialog opens.
2. Run an event search with the patterns and criteria that you want to use for the policy.

   > **NOTE**
   > The Scope options do not apply to event policy.

   The search results display. Search results are required for events to appear in the Event Log table for previewing policy changes. Even though scope does not affect the policy, scope the search if it makes the results more accurate.
3. Click Create Policy.
   The Create Event Policy wizard opens and displays the New Policy page.

   > **NOTE**
   > Only one user can create an event policy at one time. If any other logged in user has the Create Event Policy wizard open, an error message appears saying that another user has a lock on the functionality. The user must close the dialog before you can create event policies.

   If the search patterns would cause an event policy deployment error, a corresponding error message appears that might prevent you from creating the policy. For more information, see Error Messages.
4. Enter a name in the Policy Name field, select Create New Event, and click Next.

   > **NOTE**
   > The policy name cannot have more than 128 characters and cannot contain any characters listed as prohibited on the tooltip that appears when you hover over the Policy Name text.

   The Create New Event page opens.
5. (Optional) Select Reevaluate to reprocess the newly created event and all events matching the search pattern after completion of the create event action.
   This option allows for other actions to occur on the new event. For example, you can enrich the new event with additional information using an enrichment action.

   > **NOTE**
   > For a full scenario using reevaluation, see Event Management Scenarios.
6. Specify values for the USM properties of the new event as follows and click Next:
   - Use the substitution strings displayed by default in each field to use the properties from an event that matches one of the specified search patterns. For example, ${pattern1.AlertedMdrProduct} indicates that the new event acquires the AlertedMdrProduct value of the event that matched Event Pattern 1. By default, all properties except the custom User Attributes use the corresponding properties from the event that matched Event Pattern 1. Change the number in the substitution string to switch to a different event pattern.
   - Right-click a Value cell to select property values from any of the defined event patterns and functions to perform common data conversions, such as fully qualified domain name, time, IP address, and case.
     For example, fx:ip(${pattern1.AlertedMdrProdInstance}) converts the AlertedMdrProdInstance value to the system IP address.
   - Select events in the Event Log table to preview property values based on events returned by the search you ran in Step 2. You must run an event search to get events in the Event Log table for previewing the new events based on existing event content.
   - Highlighted properties are required USM alert properties. These properties must contain values to create a valid alert.
   - Enter custom values in any of the attribute fields as necessary. For example, you may want the new event to have a higher severity than the previous events. You can also use a combination of custom values and substitution patterns for certain situations, such as when you want to append the summary from a certain pattern with more descriptive information for the new event.

For example, entering 'Recurring condition - ${pattern1.Summary}' prefaces the Summary value from the event that matched pattern 1 with clarification that the condition has occurred multiple times.

- Custom values must be valid values for that property to avoid errors. If you enter an invalid value for a property with enumerated values (like Severity), an error message appears at the bottom of the dialog that prevents you from proceeding. For more information about valid enumerated values, see Event Properties and Event Information.
- Use the Service right-click menu selection to assign the created event to a service by populating the AlertedMdrProduct, AlertedMdrProdInstance, and AlertedMdrElementID values based on the service you select. You can also assign the event to the same CI as one of the matched events using the values for these properties in one of the event patterns (for example, ${pattern1.AlertedMdrElementID}. If you enter a custom value for AlertedMdrElementID, AlertedMdrProdInstance, and AlertedMdrProduct, verify that the values match an existing CI so that the created event associates with a valid CI.
- If you enter regular expression patterns for a value, consider that the deployed policy assumes a '^.' at the beginning of each expression and a '&' at the end. When these do not exist, the policy adds a '.*' in their place. To work around this issue, add the expected characters at the beginning and end of expressions.
- Populate the User Attribute properties with custom values that do not fit in other properties, if necessary. These properties appear under their original names in the Event Policy dialog, even if you renamed them. Assigning values to these original names properly displays the values under the renamed properties in the Operations Console.
- If subsequent events require information about the created event to eventually clear it, consider manually setting the MdrElementID value.

  NOTE
  MdrElementID is a unique key to each alert, per connector. If you use the same MdrElementID value for multiple alerts from the same connector, the policy updates the existing alert with the new values instead of creating a new one.
- Right-click a column name for additional help information.

The Select Data Sources page opens.

7. Do one of the following:
   - Select Save policy only and click Finish.
     The policy is saved but not deployed, and it appears in the Saved Policies section of the Events tab. This policy does not dynamically evaluate and process events according to its defined search patterns and resultant actions until you deploy it. Skip the rest of this procedure if you want to save the policy without deploying.
   - Select Save and Deploy policy.
     The list of available connectors becomes available.
8. Select the connectors on which to deploy the policy according to the guidelines in Deployment Best Practices, move them to the Selected Data Sources pane, and click Next.
   The Confirm page opens and displays the following information:
   - Data sources on which to deploy the policy
   - Time scope
   - Search patterns for policy criteria
   - Event action type and action details
9. Click Finish.
   The policy is created, and it appears in the Deployed Policies section of the Events tab and under the appropriate data source in the Data Source section.

Create Event Action Examples

The following examples show how you can use the create event action to create new events when search patterns match:

NOTE
For detailed end-to-end scenarios that use the create event action, see Event Management Scenarios.

Example: Create new event to indicate a pattern of CPU usage spikes

This example creates a new event that summarizes a condition detected through event correlation: in this case, a pattern of CPU usage spikes on a system:

- Enter a search pattern in the Event Search tab similar to the first example in Event Search Examples: Occurrence Frequency, and click Search to obtain results for previewing the new event based on existing events.
- Change the following values in the Assigned Value column:
  - Set the Severity to Critical using the right-click menu.
  - Set the Summary to 'CPU Usage Spiking on ${pattern1.AlertedMdrElementID}.
  - Set the Severity Trend to Increasing.
  - mdrElementID: fx:uniqueidentifier()

  The created event inherits all other properties from the original events, with an elevated severity and a summary that clarifies the correlated condition.
- Deploy the policy on the Mid-tier connector to include events from all connectors in the policy, or on specific connectors only to limit the policy to those connectors.
- Create and deploy a separate filter action using the same search criteria to filter the original events.

Example: Create a new event when physical and virtual memory exceed a threshold within 60 seconds

This example creates a new event when the virtual and physical memory on a system each exceed 80% within a 60 second time span:

- Enter the following search patterns in the Event Search tab, select 'all events occurs within 60 seconds' in the Additional Criterion pane, and click Search to obtain results for previewing the new event based on existing events:
  - Pattern 1: AlertedMdrElementID=? and matches(Summary,'Physical Memory') and Severity='Major'
  - Pattern 2: AlertedMdrElementID=? and matches(Summary,'Virtual Memory') and Severity='Major'
- Change the following values in the Assigned Value column:
  - Severity: Critical
  - Summary: MEMORY SHORTAGE
  - Message: Symptom: ${pattern1.Summary} + ${pattern2.Summary}
  - mdrElementID: fx:uniqueidentifier()

  The created event inherits all other properties from the original events, with an elevated severity, a summary that clarifies the correlated condition, and a message that includes the summary of each original event.
- Deploy the policy on the Mid-tier connector to include events from all connectors in the policy, or on specific connectors only to limit the policy to those connectors.
- Create and deploy a separate filter action using the same search criteria to filter the original events.

Example: Create a new event when an expected backup stopped event does not occur after a backup started event

This example creates a new event when an expected event is not occurring. In this case, an event signifying the completion of a backup job should occur within 120 seconds after an event that signifies the start of a backup job. When the completion event does not occur, this example creates a new event that describes the condition:

- Enter the following search patterns in the Event Search tab, select 'ALL events occurs within 120 seconds' and Sequence enforced in the Additional Criterion pane, and click Search to obtain results for previewing the new event based on existing events:
  - Pattern 1: AlertedMdrElementID=? and matches(Summary,'Backup started')
  - Pattern 2: not(AlertedMdrElementID=? and matches(Summary,'Backup stopped'))
- On the Create New Event page, change the following values in the Assigned Value column:
  - Severity: Critical
  - Summary: BACKUP NOT COMPLETED IN TIME
  - mdrElementID: fx:uniqueidentifier()

The created event inherits all other properties from the backup started event with an elevated severity and a summary that clarifies the condition implied by the missing backup completed event.

- Deploy the policy on the applicable connector to limit the policy to that connector.

Preventing Duplicate Create Event Actions

When generating a new event from an incoming event, you can prevent any duplicates of the event from getting generated during the new event creation process. You can do so with the help of a configurable create event action reset interval. The create event action reset interval does not allow create event actions to repeat over a specified period. By default, create event actions occur every time that the policy detects a criteria match. However, you can set the action reset interval to prevent duplicate actions. For example, you can set a reset interval of 90 seconds, which prevents duplicate create event actions from recurring within a 90-second window. As a result, only required events are forwarded into the system, decreasing the time that is required to interpret and resolve critical alerts. This ability also gives you more control over your event management system.

Define a Create Event Action Reset Interval

You can set an action reset interval to prevent duplicate create event actions, if necessary.

Follow these steps:

1. Navigate to the <SOI_HOME>\jsw\conf folder and locate the SAM-IntegrationServices.conf file.
2. Open the file in a text editor.
3. Add the following property:
   ```
   wrapper.java.additional.n=-DACTION_RESET_INTERVAL=interval
   ```
   – *n*
     Use the appropriate sequential number for *n* in the path so that no numbers are duplicated or skipped.
   – *interval*
     Defines the number of seconds before the action repeats.

   For example, to set the event action reset interval to five seconds where the sequential property number is 108:
   ```
   wrapper.java.additional.108=-DACTION_RESET_INTERVAL=5
   ```
4. Save the changes.
5. Restart the CA SAM Integration Services service.
   The value is set.

Create an Event Policy with an Enrichment Action

You can create and deploy an event policy that enriches events with useful information from outside sources that is not available in the original event. This functionality replaces the event enrichment functionality from previous releases. Deploying an event policy with an enrichment action on the Mid-tier connector reproduces the functionality of event enrichment (enrichments enacted on all events).

The following types of enrichment are available:

- **Database enrichment**
  Extracts information from a database and adds the information to the event. Perform this enrichment on the Mid-tier connector only. CA Catalyst connectors do not support direct database enrichment policy deployment.
- **Java method enrichment**
  Runs a Java method and add its output information to the event.
- **Script enrichment**
  Runs a script and adds its output information to the event.
- **Map enrichment**

Enriches events with information that does not require a connection to any external data source. You configure map enrichment values manually using a combination of static text, functions, and other event property values.

Event search patterns let you limit the events that are enriched, and you can define multiple enrichments using multiple policies. The following examples show how enrichment can be useful:

- Enriching events from a specific domain manager with links to that domain manager's knowledge base for more information
- Enriching events with contact information based on an external database, which would make alerts easier to assign in the Operations Console
- Enriching events with any information that would facilitate escalation, so that you can define alert escalation policy that occurs based on enriched information
- Enriching events with information that you can use as criteria for grouping alerts into queues

You can perform enrichments on a subset of optional event properties. Leveraging user attribute values for enrichment is a common usage of the user attributes. If you require more user attributes than are provided, as a best practice, you can assign multiple different values to a user attribute and use Matches Regex logic in escalation policy to filter out values and take action based on a certain type of value in the attribute.

To modify the required event properties, use a normalization action instead.

> **NOTE**
> For an example end-to-end deployment scenario using a database enrichment action, see Event Management Scenarios.

Follow these steps:

1. From the Operations Console, select any item in the Navigation pane, and select Tools, Event Policies.
   The Event Policy dialog opens.
2. Run an event search with the patterns and criteria that you want to use for the policy.

   > **NOTE**
   > The Scope options do not apply to event policy.

   The search results display. Search results are required for events to appear in the Event Log table for previewing policy changes. Even though scope does not affect the policy, scope the search if it makes the results more accurate.
3. Click Create Policy.
   The Create Event Policy wizard opens and displays the New Policy page.

   > **NOTE**
   > Only one user can create an event policy at one time. If any other logged in user has the Create Event Policy wizard open, an error message appears saying that another user has a lock on the functionality. The user must close the dialog before you can create event policies.

   If the search patterns would cause an event policy deployment error, a corresponding error message appears that might prevent you from creating the policy. For more information, see Error Messages.
4. Enter a name in the Policy Name field, select Enrich Event, and click Next.

   > **NOTE**
   > The policy name cannot have more than 128 characters and cannot contain any characters listed as prohibited on the tooltip that appears when you hover over the Policy Name text.

   The Enrichment Configuration page opens.
5. (Optional) Select Reevaluate to reprocess the event after enrichment occurs.
   This option allows for other actions to occur on the enriched event.

   > **NOTE**
   > For a full scenario using reevaluation, see Event Management Scenarios.

6. Select one of the following, and complete the connection information and enrichment assignment as described in the linked sections:
   - JDBC
   - Java method
   - Script
   - Map only

   The Select Data Sources page appears after you finish configuring the enrichment.
7. Do one of the following:
   - Select Save policy only and click Finish.
     The policy is saved but not deployed, and it appears in the Saved Policies section of the Events tab. This policy does not dynamically evaluate and process events according to its defined search patterns and resultant actions until you deploy it. Skip the rest of this procedure if you want to save the policy without deploying.
   - Select Save and Deploy policy.
     The list of available connectors becomes available.
8. Select the connectors on which to deploy the policy according to the guidelines in Deployment Best Practices, move them to the Selected Data Sources pane, and click Next.
   The Confirm page opens and displays the following information:
   - Data sources on which to deploy the policy
   - Time scope
   - Search patterns for policy criteria
   - Event action type and action details
9. Click Finish.
   The policy is created, and it appears in the Deployed Policies section of the Events tab and under the appropriate data source in the Data Source section.

Create a Database Enrichment Action

Contents

Create an event policy with a database enrichment action to query a database based on specified event properties and add returned information from the database to specific areas in the event. The database must be able to use a JDBC connection.

For example, you could query an internal database of contacts for all resources in your enterprise, and add the necessary contact to the event to accelerate alert assignment and resolution.

Database enrichments require the following information:

- Connection information for the database
- Match criteria that pinpoints a database row from which to extract the enrichment value based on event properties
- The database column value to use for the enrichment output value

Deploy database enrichments on the Mid-tier connector only. CA Catalyst connectors do not support direct database enrichment policy deployment. Deploying them through the Mid-tier connector enriches CA Catalyst connector events without deploying directly on the connectors.

Follow these steps:

1. Create an event policy based on a search pattern, and select Enrich Event as the action type.
   The Enrichment Configuration page opens.
2. Select JDBC in the Type drop-down list.
   Fields appear for entering JDBC connection information.
3. (Optional) Select the appropriate database type in the Templates drop-down list.

Template connection information for the database type appears in the fields. Edit the template text with database-specific information, such as full path information, database server, and database name.

4. Enter information or edit template text in the following fields:
   Note: The Test button does not work when using the SQL Server JDBC driver (sqljdbc<version>.jar) for SQL Server connection, even though the enrichment does work with the SQL Server JDBC driver. Use a JTDS driver instead for SQL Server connections to enable the Test button.
   - **Class Path**
     Defines the path and file name of the JDBC driver.
     Example: C:\Program Files\Oracle\ojdbc6.jar
   - **Class Name**
     Defines the JDBC driver class name of the database to use for the enrichment.
     Example: oracle.jdbc.driver.OracleDriver
   - **Connection**
     Defines the JDBC connection string of the database to use for the enrichment. Do not include credentials in the connection string; use the User and Password fields instead so that the password is encrypted.
     Example: jdbc:oracle:thin:@server01:1521:MyDB
   - **User**
     Defines the database user name.
   - **Password**
     Defines the password for the specified database user name.
   - **Table or View**
     Defines the database table from which to extract information for the enrichment. This field is case-sensitive and must exactly match the table name in the database.

     NOTE
     The text at the bottom of the page indicates if any required information is missing. The examples in this step are for an Oracle database. For connection examples for other database types, see JDBC Connection Examples.
5. (Optional) Click Test.
   A confirmation dialog opens.
6. (Optional) Click Yes.
   The database connection is verified. The Configuration Test Result dialog indicates whether the connection was successful. If an error occurred the displayed error message attempts to isolate the reason for the problem (for example, if the database table does not exist).

   NOTE
   If you have to change this information after deploying the policy, restart the CA SAM Integration Services service on the system to ensure that the change takes effect immediately. For information about how to configure enrichment value caching, see Configure Enrichment Cache Timeout.
7. Click Next.
   The Enrichment Policy page opens. Right-click each column on this page for additional help information.
8. Enter the following in the Parameter Configuration table to construct the WHERE clause of a SQL query that determines how the input parameters to the enrichment process are assigned according to database column values and event properties:
   - **Input Parameter**
     Defines the database column on which to search for the appropriate input value. Right-click in a cell to select from the available columns in the defined database table.

     NOTE
     If the right-click menu does not appear, there could be problems with the database connection. Return to the Enrichment Configuration page and test the database connection before proceeding. If you changed

the table name and returned to this page, click on a different cell before right-clicking to avoid seeing columns from the previously entered database table.

– **Assigned Value**
  Defines the event property value to use to query the database for a value that matches the specified column in corresponding Input Parameter cell. Use the right-click menu to assign the value of a property from any matching pattern. For example, you can search a database column named HostName based on the value of the MdrProdInstance property. The search value for each database column can take any of the following forms:
  - A full event property
  - Multiple combined event properties
  - Part of an event property
  - Modified event properties

  Use the right-click menu to add provided functions to perform common data conversions on the search value to use in the database query.

  NOTE
  Querying a database that uses fixed columns (which often applies with Oracle databases) may require you to pad the assigned value. Use a SQL function such as Rpad to add the additional spaces to ensure that the queried value is found. For example, you would need to enter the assigned value as follows if the column on which you want to match has a fixed width of 64 characters: rpad(${AlertedMdrProdInstance},64).

The Preview cell displays the result of the entered value based on the selected event in the Event Log table. You must run an event search before creating the policy to get its results in the Event Log table for previewing enrichment values based on existing event content.
You can include multiple columns in the Parameter Configuration table to use in the WHERE clause. If no match occurs for an event, the enrichment does not occur for that event. For example, if you want to enrich when the value of a HostName column matches that of the event's MdrProdInstance value, no enrichment occurs when no match is found.

9. Enter the following in the Enrichment Property Assignment table to specify how enrichment output values are assigned to event properties, and click Next:
   – **Assigned Value**
     Defines the database column values to assign to the event properties in the Event Property column. This value completes the database query started in the Parameter Configuration pane. It is the SELECT statement that uses the WHERE clause constructed in the Parameter Configuration pane to select the specified column value to use for the enrichment output from the appropriate row.
     Right-click in a cell to select from the available columns in the defined database table. If you enter columns manually, references to database columns must be in the following format: ${*columnname*}. For example, ${hostname} uses the returned value from the hostname database column for the enrichment. Any values entered without this format appear directly in the event as written. You can add enrichments to as many event properties as necessary.

     NOTE
     You can change the names of the User Attribute properties if you want them to accurately represent the enrichment properties that you assign to them. However, these properties appear under their original names in the Event Policy dialog, even if you renamed them. Assigning values to these original names properly displays the values under the renamed properties in the Operations Console.

     The column-based value can be a single column value, multiple values, or a modified column value. Use the right-click menu to add provided functions to perform common data conversions on the enrichment value before assigning it to the specified property.

     NOTE
     Only the properties that support enrichment value assignment appear in the Event Property column.

   The Select Data Sources page opens.

10. Save or deploy the policy.

JDBC Connection Examples

The following list provides the common values that you can use as templates for establishing connections with the database types supported for enrichments. You can also leverage the information in the Templates drop-down list to populate the connection fields with the template text.

Microsoft SQL Server

- Class Name: net.sourceforge.jtds.jdbc.Driver
- Connection: jdbc:jtds:sqlserver://server01:1433/SAMStore

OR

- Class Name: com.microsoft.sqlserver.jdbc.SQLServerDriver
- Connection: jdbc:sqlserver://server01:1433;databaseName=MyDB

Oracle

- Class Name: oracle.jdbc.driver.OracleDriver
- Connection: jdbc:oracle:thin:@server01:1521:MyDB

> **NOTE**
> Oracle 10.x uses port 1521, while Oracle 11.2 uses port 1045.

MySQL

- Class Name: com.mysql.jdbc.Driver
- Connection: jdbc:mysql://server01:3306/MyDB

Database Enrichment Examples

The following examples show how you can use the enrich event action to enrich events with information from an external database when search patterns match:

> **NOTE**
> For detailed end-to-end scenarios that use the database enrichment action, see Event Management Scenarios.

Example: Enrich events with database contact information

This example enriches events with contact information stored in an external database. This contact information could be useful for alert assignment and problem resolution. You could use the contact information to create alert queues based on the assigned technician or to automate emailing the assigned technician as part of an escalation policy.

Note: This example uses sample database server information that you can replace.

- Select JDBC on the Enrichment Configuration page, and enter database connection information according to the database type, database server, and database table. This example uses the following:
  - The Microsoft SQL Server conventions described in JDBC Connection Examples and accessible from the MS SQL entry in the Templates drop-down list.
  - The database server dbserver1 in the Connection string
  - The database name ResourceCatalog in the Connection string
  - The database table Contacts in the Table or View field
- Do the following on the Enrichment Policy page:
  - Select HostName from the right-click menu in the Input Parameter column and enter ${pattern1.AlertedMdrProdInstance} in the corresponding Assigned Value cell.

This parameter configuration queries the HostName column of the database based on the value of the AlertedMdrProdInstance property.
  - Select Name from the right-click menu in the Assigned Value cell corresponding to the User Attribute 1 Event Property cell.
  - Select Email from the right-click menu in the Assigned Value cell corresponding to the User Attribute 2 Event Property cell.

  This enrichment queries the Contacts table of the ResourceCatalog database using a SQL statement similar to the following:

  ```
  SELECT Name, Email WHERE HostName=${pattern1.AlertedMdrProdInstance}
  ```

  The enrichment uses the AlertedMdrProdInstance value of each event and searches for a match in the HostName column of the database. If there is no match, the enrichment does not occur. If there is a match, the Name and Email column values are returned from the matching row and assigned to the User Attribute 1 and User Attribute 2 properties in the enriched event.
  For example, consider an event with the AlertedMdrProdInstance value of server1. The server1 value matches a HostName database column value. The Name and Email values in the matching row are Dave and dave@ca.com, and these values appear in the enriched event for User Attribute 1 and User Attribute 2.
- Deploy the policy on the Mid-tier connector to include events from all connectors in the policy.

Example: Enrich events with maintenance schedule information

This example enriches events with maintenance information stored in an external database. CA SOI automatically synchronizes maintenance information from several connectors. However, for connectors that do not synchronize maintenance, you could use an enrichment to pull the maintenance status and schedule from the domain manager database so that you can determine whether maintenance is the cause of an alert and enter a corresponding maintenance schedule for the CI in CA SOI.

> **NOTE**
> This example uses sample database server information that you can replace.

- Select JDBC on the Enrichment Configuration page, and enter database connection information according to the database type, database server, and database table. This example uses the following:
  - The Oracle conventions described in JDBC Connection Examples and accessible from the Oracle entry in the Templates drop-down list.
  - The database server dbserver2 in the Connection string
  - The database name ProductDB in the Connection string
  - The database table Maintenance in the Table or View field
- Do the following on the Enrichment Policy page:
  - Select HostName from the right-click menu in the Input Parameter column and enter ${pattern1.AlertedMdrProdInstance} in the corresponding Assigned Value cell.
    This parameter configuration queries the HostName column of the database based on the value of the AlertedMdrProdInstance property.
  - Select Status from the right-click menu in the Assigned Value cell corresponding to the User Attribute 1 Event Property cell.
  - Enter fx:xsdateTime(${StartTime}), using the right-click menu to select the function and embedded column value, in the Assigned Value cell corresponding to the User Attribute 2 Event Property cell.
  - Enter fx:xsdateTime(${EndTime}), using the right-click menu to select the function and embedded column value, in the Assigned Value cell corresponding to the User Attribute 3 Event Property cell.
  - Enter fx:fdqn(${Backup}), using the right-click menu to select the function and embedded column value, in the Assigned Value cell corresponding to the User Attribute 4 Event Property cell.

  This enrichment queries the Maintenance table of the ProductDB database using a SQL statement similar to the following:

  ```
  SELECT Status, StartTime, EndTime, Backup WHERE HostName=${pattern1.AlertedMdrProdInstance}
  ```

The enrichment uses the AlertedMdrProdInstance value of each event and searches for a match in the HostName column of the database. If there is no match, the enrichment does not occur. If there is a match, the Status, StartTime, EndTime, and Backup column values are returned from the matching row and assigned to the User Attribute 1-4 properties in the enriched event.
For example, consider an event with the AlertedMdrProdInstance value of server5. The server5 value matches a Host Name database column value. The values in the matching row are as follows:
- Status: Maintenance
- StartTime: 05 09 2011 5:00:00
- EndTime: 05 12 2011 9:00:00
- Backup: 12.543.34.58

The Maintenance value appears in the User Attribute 1 property of the enriched event, and you can set the maintenance status of the CI accordingly in CA SOI. The StartTime and EndTime column values are converted to dateTime strings and appear in the User Attribute 2 and 3 properties of the enriched event. You could use these values to create a corresponding maintenance schedule for the CI in CA SOI. The IP address in the Backup column appears in the User Attribute 4 property of the enriched event converted to a fully qualified domain name, so that you can be aware of the CI performing the same function while the primary CI is in maintenance.

- Deploy the policy on the Mid-tier connector.

Create a Java Method Enrichment Action

Contents

Create an event policy with a Java method enrichment action to run a Java method and enrich an event based on the method output values. The Java method must return a comma-separated list of properties and returned values as follows for the enrichment to work:

propertyname,*value*,*propertyname*,*value*...

Java method enrichments require the following information:

- Method name and class path
- Values to use for the method parameters based on event properties
- Script output values to assign as enrichment output values

The method must exist on the SA Manager and on every connector system to which you want to deploy the event policy.

Follow these steps:

1. Create an event policy based on a search pattern, and select Enrich Event as the action type.
   The Enrichment Configuration page opens.
2. Select Java method in the Type drop-down list and enter information in the following fields:

   NOTE
   The text at the bottom of the page indicates if any required information is missing.

   - **Java Class Path**
     Defines the full class path and jar file of the method to use for the enrichment.
     Example: <C:\Program Files\CA\SOI\lib\ivy\em.event-plus-catalog-4.2.0.jar>
   - **Class Name**
     Defines the class name of the Java method, including the package, to use for the enrichment.
     Example: com.ca.eventplus.catalog.methods.CMDBEnrich
   - **User**
     (Optional) Defines the user name to run the Java method, if necessary.
   - **Password**
     (Optional) Defines the password for the specified Java method user name, if necessary.

NOTE
If the method requires user authentication in its parameters, enter the credentials here and reference them on the following page to ensure that the data is protected.

– **Method**
Defines the name of the Java method to run from the referenced class.
Example: performCMDBEnrichment_v2

3. (Optional) Click Test.
A confirmation dialog opens.
4. (Optional) Click Yes.
The Java method connection is verified. The Configuration Test Result dialog indicates whether the connection was successful.

NOTE
If you have to change this information after deploying the policy, restart the CA SAM Integration Services service on the connector system to ensure that the change takes effect. For information about how to configure enrichment value caching, see Configure Enrichment Cache Timeout.

5. Click Next.
The Enrichment Policy page opens. Right-click each column on this page for additional help information.
6. Enter the following in the Parameter Configuration table to determine how the input parameters to the enrichment process are assigned according to method parameter values and event properties:
   – **Input Parameter**
   Defines placeholder names for each required method input parameter. The enrichment always reads the parameters sequentially; therefore, the names that you enter for each parameter can be anything (param1, param2, and so on). Create an entry for each required input parameter to ensure that the method runs successfully.
   – **Assigned Value**
   Defines the event property or other value to use for the corresponding method parameter value. Use the right-click menu to assign the value of a property from any matching event pattern. The value for each method parameter can take any of the following forms:
   - A full event property
   - Multiple combined event properties
   - Part of an event property
   - Modified event properties

   Use the right-click menu to add provided functions to perform common data conversions on the search value to use for each method parameter.
   To enter user credentials, use the following substitution characters to reference the credentials entered on the previous page:
   ```
   ${user}
   ${password}
   ```

   NOTE
   Entering a password value manually on this page creates an unencrypted record of the password.

   The Preview cell displays the result of the entered value based on the selected event in the Event Log table. You must run an event search before creating the policy to get its results in the Event Log table for previewing enrichment values based on existing event content.
   Include all required parameters for the method to run. If the Java method does not run successfully based on the entered parameters or does not return a comma-separated list of properties and values, the enrichment does not occur for that event.
7. Enter the following in the Enrichment Property Assignment table to specify how enrichment output values are assigned to event properties, and click Next:
   – **Assigned Value**

Defines the Java method output property values to assign to the event properties in the Event Property column. This value determines the property value to use for the enrichment from the comma-separated list of properties and values that the method returns.
References to output properties must be in the following format: ${*propertyname*}, where propertyname is the name of the property in the comma-separated output list whose value you want to return. For example, for a method that returns the string 'user,*value*,role,*value*,department,*value*', ${role} uses the returned value from the role output property for the enrichment. Any values entered without this format appear directly in the event as written. You can add enrichments to as many event properties as necessary.

NOTE

You can change the names of the User Attribute properties if you want them to accurately represent the enrichment properties that you assign to them. However, these properties appear under their original names in the Event Policy dialog, even if you renamed them. Assigning values to these original names properly displays the values under the renamed properties in the Operations Console.

The method property-based value can be a single property value, multiple values, or a modified property value. Use the right-click menu to add provided functions to perform common data conversions on the enrichment value before assigning it to the specified property. The return value cannot contain an embedded comma.

NOTE
Only the properties that support enrichment value assignment appear in the Event Property column.

The Select Data Sources page opens.

8. Save or deploy the policy.

Java Method Enrichment Examples

The following examples show how you can use the enrich event action to enrich events with information from a Java method when search patterns match. The examples uses CA CMDB and CA Spectrum enrichments provided with the Mid-tier connector. The enrichments could be useful in CA SOI for situations such as the following:

- You want to use CI properties in CA CMDB or CA Spectrum in alert queue criteria or for escalation policy
- You are using custom properties in CA CMDB or CA Spectrum that are not imported into CA SOI
- You want to use information from CA CMDB or CA Spectrum in alerts or CIs that are not managed in CA CMDB or CA Spectrum
- You want to parse partial information from a property for use in a different context

WARNING
The enrichments use .jar files that are only available with the Mid-tier connector. Deploy these provided enrichments on the Mid-tier connector only.

Example: Enrich events with location information from CA CMDB

This example enriches events with location information stored in CA CMDB. The information could help you create alert queues by location or add location-based criteria to escalation policy.

- Select Java Method on the Enrichment Configuration page, and select CMDB in the Templates drop-down list.
- Use the User and Password fields to enter valid credentials for the CA CMDB server, and leave the default values in all other fields.
- Do the following on the Enrichment Policy Configuration page:
  - Enter values for the provided method parameters in the Input Parameter column in the Assigned Values column:
    a. endpointref: http://<cmdbserver>:8080/axis/services/USD_R11_WebService?wsdl
    b. userid: ${user}
    c. password: ${password}
    d. propertylist: location.address,location.city
    Note: For CA CMDB r12 and above, use location.address1 instead of location.address.

e. selectquery: dns_name like "%s"
f. node: ${pattern1.AlertedMdrProdInstance}

This parameter configuration queries the defined CA CMDB instance for CIs with a dns_name property that matches the event AlertedMdrProdInstance property value and returns the location.address and location.city properties of the matching CI. It uses substitution strings for the required CA CMDB credentials (referencing the credentials entered on the previous page) to avoid entering the information unencrypted.

– Enter ${location.address} in the Assigned Value cell corresponding to the User Attribute 1 Event Property cell.

NOTE
For CA CMDB r12 and above, use location.address1 instead of location.address.

– Enter ${location.city} in the Assigned Value cell corresponding to the User Attribute 2 Event Property cell.

This enrichment queries the defined CA CMDB instance for the location.address and location.city properties of the CI with a dns_name property value that matches the event AlertedMdrProdInstance property value. If there is no match, the enrichment does not occur. If there is a match, the location.address and location.city values are returned from the matching CI and assigned to the User Attribute 1 and User Attribute 2 properties in the enriched event.
For example, consider an event with the AlertedMdrProdInstance value of server4. The server4 value matches a CA CMDB CI DNS name. The location.address and location.city values in the matching CI are 453 Elm St and Los Angeles, and these values appear in the enriched event for User Attribute 1 and User Attribute 2.

- Deploy the policy on the Mid-tier connector.
- (Optional) If you want enriched events to contain a link to the CI in CA CMDB, create a separate create event policy to add the following URL in the User Attribute 1 field to events that have been enriched:

```
http://<cmdbserver>:8080/CAisd/pdmweb.exe?OP=SEARCH+FACTORY=nr+SKIPLIST=1+QBE.EQ.id=
${pattern1.AlertedMdrProdInstance}
```

Example: Enrich events with CA Spectrum model attributes

This example enriches events with model attributes stored in CA Spectrum. The information could help you add CA Spectrum-specific information to CA SOI services.

NOTE
This example works with CA Spectrum r9.2.

- Select Java Method on the Enrichment Configuration page, and select Spectrum (By Name) in the Templates drop-down list.
- Use the User and Password fields to enter valid credentials for the CA Spectrum server, and leave the default values in all other fields.
- Do the following on the Enrichment Policy Configuration page:
  – Enter values for the provided method parameters in the Input Parameter column in the Assigned Values column:
    a. i1: -searchmethod=by_name
    b. i2: -broadcast=yes
    c. i3: -landscapeout=*spectrumserver*
    d. i4: landscapeuser=${user}
    e. i5: -modeltype=Host_Device
    f. i6: modelname=${AlertedMdrProdInstance }
    g. i7: modelattributes=12bfd,12bfe

    This parameter configuration queries the defined CA Spectrum server for models with a name that matches the AlertedMdrProdInstance property value and returns the owner and organization properties of the matching model. You can enter the CA Spectrum hex codes for any model attribute.
  – Enter ${12bfd} in the Assigned Value cell corresponding to the User Attribute 1 Event Property cell.
  – Enter ${12bfe} in the Assigned Value cell corresponding to the User Attribute 2 Event Property cell.

  This enrichment queries the defined CA Spectrum server for the owner and organization attributes of the model with a name that matches the AlertedMdrProdInstance property value. If there is no match, the enrichment does not occur.

If there is a match, the owner and organization values are returned from the matching CI and assigned to the User Attribute 1 and User Attribute 2 properties in the enriched event.
For example, consider an event with the AlertedMdrProdInstance value of server4. The server4 value matches a CA Spectrum model name. The owner and organization values in the matching CI are Dave and Spectrum, and these values appear in the enriched event for User Attribute 1 and User Attribute 2.

- Deploy the policy on the Mid-tier connector.

Create a Script Enrichment Action

Contents

Create an event policy with a script enrichment action to run a script and enrich an event based on its output values. The script must return a comma-separated list of properties and returned values as follows for the enrichment to work:

propertyname,*value*,*propertyname*,*value*...

Script enrichments require the following information:

- Script path and name
- Values to use for the script parameters based on event properties
- Script output values to assign as enrichment output values

The script must exist locally on every connector system to which you want to deploy the event policy.

Follow these steps:

1. Create an event policy based on a search pattern, and select Enrich Event as the action type.
   The Enrichment Configuration page opens.
2. Select Script in the Type drop-down list and enter information in the following fields:

   NOTE
   The text at the bottom of the page indicates if any required information is missing.

   NOTE

   If the script requires user authentication in its parameters, enter the credentials here and reference them on the following page to ensure that the data is protected.

   – **Script Path**
     Defines the directory path of the script to run for the enrichment, excluding the script name.
     Example: C:\Program Files\myscripts
   – **Script Name**
     Defines the script name to use for the enrichment. Do not add command line arguments to the script name.
     Example: GetObjectProperties.exe
   – **User**
     (Optional) Defines the user name for running the script, if necessary.
   – **Password**
     (Optional) Defines the password for the specified script user name, if necessary.
3. (Optional) Click Test.
   A confirmation dialog opens.
4. (Optional) Click Yes.
   The script connection is verified. The Configuration Test Result dialog indicates whether the connection was successful.

NOTE
If you have to change this information after deploying the policy, restart the CA Catalyst Container service on the connector system or the appropriate plugin service to ensure that the change takes effect. For information about how to configure enrichment value caching, see Configure Enrichment Cache Timeout.

5. Click Next.
   The Enrichment Policy page opens. Right-click each column on this page for additional help information.
6. Enter the following in the Parameter Configuration table to determine how the input parameters to the enrichment process are assigned according to script parameter values and event properties:
   – **Input Parameter**
     Defines placeholder names for each required script input parameter. The enrichment always reads the parameters sequentially; therefore, the names that you enter for each parameter can be anything (param1, param2, and so on). Create an entry for each required input parameter to ensure that the script runs successfully.
   – **Assigned Value**
     Defines the event property or other value to use for the corresponding script parameter value. Use the right-click menu to assign the value of a property from any matching event pattern. The value for each script parameter can take any of the following forms:
     - A full event property
     - Multiple combined event properties
     - Part of an event property
     - Modified event properties
     - An associated CI property if you have enabled Persistent Store enrichment

     Use the right-click menu to add provided functions to perform common data conversions on the search value to use for each script parameter.
     To enter user credentials, use the following substitution characters to reference the credentials entered on the previous page:
     ```
     ${user}
     ${password}
     ```

     NOTE
     Entering a password value manually on this page creates an unencrypted record of the password.

   The Preview cell displays the result of the entered value that is based on the selected event in the Event Log table. You must run an event search before creating the policy to get its results in the Event Log table for previewing enrichment values based on existing event content.
   Include all required parameters for the script to run. If the script does not run successfully based on the entered parameters or does not return a comma-separated list of properties and values, the enrichment does not occur for that event.
   Example
   The following script adds a support person contact details depending on alert severity.
   ```
   @echo off
   if %1==Critical
       (
       echo lname,Scott,fname,Sue,email,email01@company.com,phone,631-001-0001,severity,%1
       )
   else (if %1==Major
        (
        echo lname,Black,fname,Bill,email,email02@company.com,phone,631-002-0002,severity,%1
        )
   else (if %1==Minor
        (
        echo lname,Unum,fname,Sven,email,email03@company.com,phone,631-003-0003,severity,%1
        )
   ```

))

In the Parameter Configuration table, assign a variable whose value the input parameter in the script (%1) will use:

Input Parameter	Assigned Value
Severity	${pattern1.Severity}

- Enter the following in the Enrichment Property Assignment table to specify the enrichment output values to assign to event properties, and click Next:
  - **Assigned Value**
    Defines the script output property values to assign to the event properties in the Event Property column. This value determines the property value to use from the comma-separated list of properties and values that the script returns. References to output properties must be in the following format: ${*propertyname*}, where propertyname is the name of the property in the comma-separated output list whose value you want to return. For example, for a script that returns the string 'city,*value*,state,*value*,zip,*value*', ${city} uses the returned value from the city output property for the enrichment. Any values entered without this format appear directly in the event as written. You can add enrichments to as many event properties as necessary.
    Note: You can change the names of the User Attribute properties if you want them to accurately represent the enrichment properties that you assign to them. However, these properties appear under their original names in the Event Policy dialog, even if you renamed them. Assigning values to these original names properly displays the values under the renamed properties in the Operations Console.
    The script property-based value can take any of the forms previously described for the input value: a single property value, multiple values, or a modified property value. Use the right-click menu to add provided functions to perform common data conversions on the enrichment value before assigning it to the specified property. The return value cannot contain an embedded comma.

    NOTE
    Only the properties that support enrichment value assignment appear in the Event Property column.

  The Select Data Sources page opens.
- Save or deploy the policy.

Script Enrichment Examples

The following examples show how you can use the enrich event action to enrich events with information from a script when search patterns match:

Example: Enrich events with information from a Windows VB script

This example illustrates how you can enrich events with output information from a Windows VB script. Information from the Windows operating system that is not already included with an event could be useful for alert diagnosis and resolution. In this example, the event policy calls a VB script using the event server name as input and returns the location of the associated server:

- Select Script on the Enrichment Configuration page, and enter the following information in the fields:
  - Script Path: C:\Windows\System32
  - Script Name: cscript /NoLogo C:\tools\LocationLookup.vbs

  This script looks up the location of the system provided in the input parameter. The /NoLogo modifier ensures that only a comma-separated list of values are returned by the script, which is required for the enrichment to work.
- Do the following in the Parameter Configuration table:
  - Enter INPUT in the first Input Parameter cell.
  - Enter fn:Parse(${pattern1.AlertedMdrElementID},'.*:(.*)') in the corresponding Assigned Value cell.

  These values control the input parameter for the VB script. The assigned value for the script input is the system on which the event occurred. The parse function extracts the server name from the AlertedMdrElementID property. This

format is valid for AlertedMdrElementID properties from the Universal connector. Output from other connectors may require a different format to return the server name.

- Enter ${retval1} in the Assigned Value cell for the User Attribute 1 property in the Enrichment Property Assignments table.
  This assignment enriches events with the output of the script in the User Attribute 1 property. For example, if an event occurs on a system in Minnesota, the script looks up the location based on the system name, and Minnesota appears in the User Attribute 1 property in the enriched event.
- Deploy the policy on the applicable connector.
  Matching events are enriched with the location of the source system in the User Attribute 1 event property. You can use the enriched location information to configure alert queues, as criteria in escalation policies, or as a way to determine alert assignments.

Create a Map Enrichment Action

Contents

To map values to event properties, create an event policy with a map enrichment action. These values are either static or dependent on functions or other event properties. They do not require a connection to an external data source, such as a database or script.

Use the map enrichments as simple enrichments that do not require a data source connection. Consider the following examples:

- You can add static information such as a company URL to one of the user attributes.
- You can manually enter assignees for each type of event in the Assignee property. Then use that information to group alerts by assignee in alert queues.

Follow these steps:

1. Create an event policy that is based on a search pattern, and select Enrich Event as the action type.
   The Enrichment Configuration page opens.
2. Select Map only from the Type drop-down list, and click Next.
   The Enrichment Policy page opens.
3. Enter enrichment values in the Assigned Value column of the Enrichment Property Assignment table for each event property to enrich, and click Next.

   > **NOTE**
   > The Parameter Configuration table requires input only when you are extracting the enrichment values from an external source.

   Map enrichment values can be any combination of static text, functions, and the values of other event properties. To access available functions and event properties, use the right-click menu. Only the properties that support enrichment value assignments appear in the Event Property column.

   > **NOTE**
   > If you want the user attribute properties to represent the enrichment properties that you assign to them, change the names of the User Attribute properties. However, these properties appear under their original names in the Event Policy dialog, even if you renamed them. Assigning values to these original names properly displays the values under the renamed properties in the Operations Console.

   The Select Data Sources page opens.
4. Save or deploy the policy.

Map Enrichment Examples

The following example shows how you can use the enrich event action to enrich events with custom mapped information when search patterns match:

Example: Enrich events with knowledge base search URL

This example enriches events with an internet search URL based on the CI name:

- Enter no search patterns in the Event Search tab, so that all events are included in the policy. Click Search if you want events to appear on the Enrichment Policy page for previewing changes.
- Select Map only as the enrichment type on the Enrichment Configuration page.
- Enter the following URL in the Assigned Value field for User Attribute 1 in the Enrichment Property Assignment table on the Enrichment Policy page:
  ```
  http://www.google.com/search?hl=en&#38;q=${pattern1.AlertedMdrElementID}
  ```
- If you use the following characters in the User Attributes, then encode these characters:
  - **&** to **&amp**;
  - **<** to **&lt**;
  - **>** to **&gt**;
  - **"** to **&quot**;
  - **'** to **&#39**;
- This string is a URL for a Google search that is based on the AlertedMdrElementID value. The value appears as the User Attribute 1 value in the enriched event. You can search based on the value of any event property, and you can change the URL to a company knowledge base.
- To include the events from all connectors in the policy, deploy the policy on the Mid-tier connector.

Configure Enrichment Cache Timeout

By default, Event Management caches enrichment connection information for 120 seconds. If you change enrichment connection values before the cache expires, the change does not immediately take effect. You can configure a custom timeout value for enrichment caching.

Follow these steps:

1. Navigate to the <SOI_HOME>\resources\configurations*connectorname*.conf file.
2. Open the file in a text editor.
3. Add the following property to the file, and save the file:
   ```
   cachetimeout=120
   ```
   Enter any value to represent the number of seconds that enrichment values remain cached.
4. Repeat Steps 1-3 for all connector configuration files that require the enrichment cache timeout settings.
5. Restart the CA SAM Integration Services service on every affected connector system.
   The changes take effect.

Create an Event Policy with a Normalization Action

Contents

You can create and deploy an event policy that manually normalizes raw events with custom mappings from raw event properties to USM alert properties. Normalizing raw events is useful when the default policy for a connector is only generic in nature. The default policy does not perform a mapping that is specific enough to manage the incoming events effectively as alerts. The following connectors are examples of connectors that have generic policy:

- Event connector (some sources)
- SNMP trap connector
- IBM Tivoli Netcool/OMNibus connector
- Oracle Enterprise Manager Grid Control connector
- IBM Tivoli Enterprise Console connector

You can also deploy a normalization action on event sources that have detailed connector policy to refine how required event properties are normalized. The mappings in the event policy overwrite any default mappings in the default policy file. If you want to add information to optional properties or the user attribute properties, use an enrichment action instead, unless that information exists in raw event properties.

The following situations are common normalization action use cases:

- Raw events contain an important property value that is not mapped to any USM alert property by the default connector policy.
- SNMP traps contain important information in their variable bindings that require mapping to individual properties.
- Large-scale event management sources (like IBM Tivoli Netcool/OMNibus) aggregate events from multiple disparate sources. You want to create specific normalization rules for events from each source.

Normalization actions require a raw event search that is available. Deploy a normalization action on only one source connector (which cannot be the Mid-tier connector).

Follow these steps:

1. From the Operations Console, select any item in the Navigation pane, and select Tools, Event Policies.
   The Event Policy dialog opens.
2. Select Raw Events in the Additional Criterion section.
   The Select Data Source dialog opens.
3. Select one connector to search, and click OK.
   The dialog displays an error if you select more than one data source.
4. Enter a search pattern for the raw events that you want to normalize, and click Search.
   The search results display. Unlike other event policies, you run an event search that returns results for the Normalize Events page to contain events for previewing policy changes and obtaining raw event properties. If no search results exist when you click Map Events, an error message appears.
5. Click Map Events.
   The Create Event Policy wizard opens and displays the New Policy page. The Data Source Type field displays the data source that you selected in Step 3. Deploy the event policy on this data source.

   NOTE
   Only one user can create an event policy at one time. If any other logged in user has the Create Event Policy wizard open, an error message appears saying that another user has a lock on the functionality. The user must close the dialog before you can create event policies.If the search patterns would cause an event policy deployment error, a corresponding error message appears. The error can prevent you from creating the policy. For more information, see Error Messages.
6. Enter a name in the Policy Name field, select Normalize Event, and click Next.

   NOTE
   The policy name cannot have more than 128 characters. Also, the name cannot contain any characters that are listed as prohibited on the tooltip that appears when you hover over the Policy Name text.

   The Normalize Event page opens. The results of the raw event search appear in the Event Log table. This table must have valid results for you to be able to map to the raw event properties.
7. Specify values for the USM properties of the raw event as follows and click Next:
   – Right-click an Assigned Value cell and select Attributes for a list of all raw event properties. Select a property to map that raw event property value to the USM property in the corresponding Event Property cell. For example,

${pattern1.syslog_user} in the Assigned Value cell for Assignee maps the value of the raw event property syslog_user to the Assignee USM property.

- When assigning raw event property values, verify that the properties you assign are actual properties from the raw event source. Assigning other properties does not work. For more information, see Raw Event Properties in Normalization Actions.
- You can map properties that the default connector policy already mapped. The normalization action overwrites the default mapping.
- Highlighted properties are required USM alert properties. These properties must contain values to create a valid alert. Leaving the properties empty prevents you from proceeding to the next page.
- If an event has historical data and a default mapping for a property in its connector policy, ${pattern1.*propertyname*} appears in the field for that property. Leave this string to retain the default mapping. This approach is useful if you only want to change the values of targeted properties, not to enter new mappings for all properties.
- Use the Service right-click menu selection to assign the normalized event. Populate the AlertedMdrProduct, AlertedMdrProdInstance, and AlertedMdrElementID values based on the service you select.
- Populate the User Attribute properties with raw event property values or custom values that do not fit in other properties, if necessary. These properties appear under their original names in the Event Policy dialog, even if you renamed them. Assigning values to these original names properly displays the values under the renamed properties in the Operations Console.
- Right-click an Assigned Value cell and select Functions to use the available functions to perform common data conversions. Conversions include a fully-qualified domain name, time, IP address, and case.
- Right-click an Assigned Value cell for a USM property with enumerated values. Select Map to map raw event property values to valid USM property values. For example, if you are mapping a raw property to the USM Severity property, verify that the raw values map to valid Severity values.
- Right-click an Assigned Value cell for a USM property with enumerated values. Select Values to define a valid value for the property that is not supplied by raw event properties. However, in most cases, the default policy handles populating enumerated properties.
- Enter custom values in any of the Assigned Value cells as necessary. For example, you want the normalized event to have a static value for a certain property. You can also use a combination of custom values and substitution patterns for certain situations. You want to append the summary from a raw event with more descriptive information for the normalized event. For example, a Message field value of ${pattern1.varbind-1.3.6.1.4.11203.6} alert on ${pattern1.varbind-1.3.6.1.4.1.203.9} scheduled on ${pattern1.snmp_agent} combines the CA Workload Automation job status, job name, and server with custom text to create a descriptive message using multiple properties.

  NOTE
  To avoid errors, custom values must be valid values for that property. If you enter an invalid value for a property with enumerated values (like Severity), an error message appears at the bottom of the dialog. The error prevents you from proceeding.
- If you enter regular expression patterns for a value, consider that the deployed policy assumes a '^.' at the beginning of each expression and a '&' at the end. When these do not exist, the policy adds a '.*' in their place. To work around this issue, add the expected characters at the beginning and end of expressions.
- Select events in the Event Log table to preview raw event property values based on events that are currently returned by the search patterns.
- Right-click a column name for additional help information.

The Select Data Sources page opens.

8. Perform one of the following actions:
   - Select Save policy only and click Finish.
     The policy is saved but not deployed, and it appears in the Saved Policies section of the Events tab. This policy does not dynamically evaluate and process events according to its defined search patterns and resultant actions until you deploy it. Skip the rest of this procedure if you want to save the policy without deploying.
   - Select Save and Deploy policy.

The list of available connectors becomes available.

9. Select the connector that appears in the Data Source Type field, move it to the Selected Data Sources pane, and click Next. You are prevented from adding data sources other than the one in the Data Source Type field, which is the connector on which you ran the original raw event search.
   The Confirm page opens and displays the following information:
   – Data sources on which to deploy the policy
   – Time scope
   – Search patterns for policy criteria
   – Event action type and action details
10. Click Finish.
    The policy is created. The policy appears in the Deployed Policies section of the Events tab. The policy also appears under the appropriate data source in the Data Source section.

Raw Event Properties in Normalization Actions

Running a raw event search returns a large set of properties. In normalization actions, only use the properties that originate from the raw event source. Other properties may exist in the raw event record, including temporary properties created during default normalization, properties resembling the USM alert properties, and others. Assigning any properties other than those from the raw event source breaks the event policy.

Use the following guidelines to help ensure that you are using true raw event properties in normalization mapping:

- True raw event properties often are prefixed by their event source names. For example, raw event properties from the SNMP connector are prefixed by 'snmp_'. The Event connector also follows this convention. For example, raw event properties from the Windows Event Log adaptor are prefixed by 'syslog_'. However, some connectors do not follow this convention.
- Variable bindings from the SNMP connector are split into properties prefixed with 'varbind-' followed by the OID number. These properties are acceptable for normalization mapping.
- Do not map to properties prefixed by 'temp_' or 'internal_'. These are properties creating during event processing, and they do not exist when the normalization action runs.
- Do not map to properties prefixed by 'usm_' or those that have the same name as USM properties. These are not raw properties from the event source.
- If you cannot tell from the search results which properties are true raw event properties, see the default policy file for the connector. The raw event properties appear as inputs.

Assign Normalized Events to a Service

Events must be associated with a managed CI when they are normalized to appear as alerts on the Operations Console. Configure the associated CI for a normalized event using the AlertedMdrProduct, AlertedMdrProdInstance, and AlertedMdrElementID properties on the Normalize Event page. These properties must contain valid values to associate the normalized event with a CI.

To populate the necessary properties automatically and to associate the event with a service, use the right-click menu on the Normalize Event page. On the Operations Console, the normalized event appears as an alert generated directly on the defined service CI.

Follow these steps:

1. Create an event policy with a normalization action, and proceed to the Normalize Event page.

   NOTE
   You can also perform this assignment when creating an event policy with a create event action.
2. Right-click the Assigned Value cell for one of the following properties and select Service:

  - Alerted Mdr Product
  - Alerted Mdr Prod Instance
  - Alerted Mdr Element ID

  The Select Service dialog opens.
3. Select the service that you want to associate with the normalized event, and click OK.
   The values for all three properties appear for the selected service on the Normalize Event page. When you deploy the policy, any events that the policy normalized appear as alerts directly on the defined service CI.

   NOTE

   If you know the properties for the exact CI to assign to the alert, you can also manually populate the three values. However, the AlertedMdrElementID value must match a CI on the searched connector for the event to appear as an alert in the Operations Console.

Map Raw Event Property Values to Enumerated USM Property Values

USM properties with enumerated values must contain one of its enumerated values for the resultant alert to be valid. The following properties are examples of ones that require specific enumerated values:

- Severity
- MdrProduct and AlertedMdrProduct
- AlertType

On the Normalize Event page of the Create Event Policy wizard for a normalization action, you must map the values of any raw event properties assigned to enumerated USM properties to valid enumerated values for that property.

For example, if you assign a raw event property to the Severity property, that raw event property could use different terminology, such as Operational, Stopped, Nonfunctional, Degraded, and so on. These terms would create an invalid alert. The values for the raw property would require mapping to the valid Severity values of Unknown, Normal, Minor, Major, Critical, and Fatal.

Follow these steps:

1. Create an event policy with a normalization action, and proceed to the Normalize Event page.
2. Right-click the Assigned Value cell for a USM property with enumerated values and select Map.
   The Map to USM Attribute dialog opens.
   Only USM properties with enumerated values include the Map function in their right-click menu.
3. Do the following in each row that contains an enumerated value in the USM Value column to which you want to map a raw event property value, and click OK:
   - Select the appropriate raw event property using the right-click menu in the Event Property column.
   - Enter the raw event property value that you want to map to the USM property value in the Value column.
   - To assign multiple event properties to the same USM property, use the | character as a delimiter. For example, to assign the Failing and Degraded event properties to the Critical severity value, you would enter Failing|Degraded in the Value column.
   - Use other regular expressions for situations that cannot be solved by direct mapping, such as values that start with the same string but have different ensuing values.
   - The Map function does not support use of embedded functions in the Value column.

   Not every USM value requires a raw event value mapping if, for example, the raw event property does not have as many values.
   The Map function that you created appears in the Assigned Value cell. When you click away from the cell, the Preview cell displays the property value based on the selected event in the Event Log table. The Preview cell does not support map values derived through regular expressions. If the map value uses a regular expression, the Preview cell displays a message 'Mapping not found by preview'. However, the mapping itself occurs as expected in actual event policy.

Normalization Action Examples

The following examples show how you can use the normalize event action to perform custom mappings of raw event properties to USM alert properties.

Example: Normalize Windows Event Log security events to make them easier to categorize

This example illustrates how to normalize events from the Security log in the Windows Event Log. The default policy for Windows Event Log events does not map vital information to USM alerts such as the following:

- Source event log (Security, System, Application, and so on)
- User
- Category

The normalization action in this example makes this information a part of the resultant alert, and you can organize the normalized alerts into queues.

- Run the following raw event search that is scoped to the MS-Syslog source, and click Map Events:
  ```
  syslog_source='Security'
  ```
- Select Normalize Event, and assign the following mappings on the Normalize Event page:
  - **Assignee: ${pattern1.syslog_user}**
    Maps the Assignee property to the internal Windows user information. This information does not appear in alerts that the default policy normalizes.
  - **User Attribute 1: ${pattern1.syslog_source}**
    Maps the User Attribute 1 property to the Windows Event Log source event log. This information does not appear in alerts that the default policy normalizes. You can use it to assign all security events to a specialized queue.
  - **User Attribute 2: ${pattern1.syslog_category}**
    Maps the User Attribute 2 property to the internal event category. This information does not appear in alerts that the default policy normalizes.

  Use the Service right-click menu to assign the AlertedMdr properties to a managed service so that the normalized event appears on that service CI.
  All other properties obtain their values from the default connector policy.
- Deploy the policy on the Windows Event Log data source.
- (Optional) Create an alert queue named Security. Configure the queue to add alerts when the User Attribute 1 property equals Security to group all alerts from the Security log.

Example: Normalize CA Workload Automation traps to assign variable bindings to USM properties

This example normalizes SNMP traps from CA Workload Automation to include important variable binding information in properties. The properties appear on the Operations Console when the event becomes an alert. Default policy for SNMP sources includes all trap varbind values in one property. Event Management splits variable bindings and their values into separate properties in the Event Store. You can map each varbind to its appropriate USM alert property.

> **NOTE**
> This normalization is similar to the default policy for the SNMP connector, which is written for CA Workload Automation traps as an example.

- Run the following raw event search that is scoped to the Generic SNMP Traps source that the SNMP connector provides. Click Map Events:
  ```
  snmp_enterprise="1.3.6.1.4.1.11203"
  ```
  This search returns traps with an enterprise OID that indicates the traps are from CA Workload Automation.
- Select Normalize Event, and assign the following mappings on the Normalize Event page:
  - **Mdr Element ID: ${pattern1.snmpagent}:${pattern1.varbind-1.3.6.1.4.1.203.7}: ${pattern1.varbind-1.3.6.1.4.1.203.9}**

Maps the MdrElementID property to a combination of the varbinds that indicate the source server, application name, and job name.

– **Severity: Use Map Function**
  Maps the Severity property to the varbind that indicates the job status. Map the values for varbind-1.3.6.1.4.11203.6 to valid Severity values using the Map function:
  Value column: USM Value column
  - Unknown|Abandon Submission: Unknown
  - Exec: Informational
  - Complete|Monitor|Ready: Normal
  - Inactive: Minor
  - Overdue|Suberror: Major
  - Failed|Premature|Agent Down: Critical

  NOTE
  The Preview cell does not support map values that are derived through regular expressions. If the map value uses a regular expression, the Preview cell displays a message 'Mapping not found by preview'. However, the mapping itself occurs as expected in actual event policy.

– **Summary: ${pattern1.varbind-1.3.6.1.4.11203.5}**
  Maps the summary property to the trap job status message.

– **Message: ${pattern1.varbind-1.3.6.1.4.11203.6} alert on ${pattern1.varbind-1.3.6.1.4.1.203.9} scheduled on ${pattern1.snmp_agent}**
  Maps the Message property to the following statement: '*jobstatus* alert on *jobname* that is scheduled on *agentserver*'.

– **User Attribute 1: ${pattern1.varbind-1.3.6.1.4.1.203.7}**
  Maps the User Attribute 1 property to the source application name.

Use the Service right-click menu to assign the AlertedMdr properties to a managed service so that the normalized event appears on that service CI.
All other properties obtain their values from the default connector policy.

- Deploy the policy on the SNMP connector data source.
- (Optional) Create alert queues that are based on key identifiers such as the job name or the application name to manage the traps most effectively.

Example: Normalize Windows operating system traps

This example normalizes traps that are collected from the Windows operating system and are related to services starting and stopping.

NOTE
For this example to work, configure Windows to generate traps for Event ID 7036. Use the Windows Event to Trap Translator and send the traps to the SNMP connector system.

- Run the following raw event search that is scoped to the Generic SNMP Traps source that the SNMP connector provides. Click Map Events:
  ```
  snmp_specificTrap="1073748860"
  ```
  This search returns traps that Windows generates for starting and stopping operating system services.
- Select Normalize Event, and assign the following mappings on the Normalize Event page:
  – **Mdr Element ID: ${pattern1.snmp_SpecificTrap}:${pattern1.varbind-1.3.6.1.4.1.311.1.13.1.9999.6.0}**
    Maps the MdrElementID property to the specific trap ID and the affected Windows service.
  – **Severity: Use Map Function**
    Maps the Severity property to the service status. Right-click the cell, select Map, and map the values for varbind-1.3.6.1.4.1.311.1.13.1.9999.7.0 to valid Severity values using the Map function:

    - running: Normal
    - stopped: Critical
  - **Summary and Message: ${pattern1.varbind-1.3.6.1.4.1.311.1.13.1.9999.1.0}**
    Maps the Summary and Message properties to the service message.
  - **Alert Type: Risk**
    Maps the AlertType property to a static value of Risk.
  - **Occurrence Timestamp and Report Timestamp: fx:xsdateTime()**
    Maps the required time-based USM properties to the current time. Find this value by right-click the cell and selecting Functions, fx:xsdateTime-now.

  Use the Service right-click menu to assign the AlertedMdr properties to a managed service so that the normalized event appears on that service CI.
- Deploy the policy on the SNMP connector data source.

Event Action Functions

The create event, enrichment, and normalization actions provide several functions that can perform common data conversions on the following values:

- Output values for the created or normalized event
- Input values on which to base an enrichment
- Output values for enrichment

When you select a function using the right-click menu, a function reference appears in the selected cell with the syntax representing the parameters you must enter, if necessary. If the function requires an input, adhere to the format of the provided syntax for the function to work.

The available functions are as follows:

Host

The following functions are categorized as Host functions:

- **fx:fqdn-conversion**
  Returns the fully qualified domain name (FQDN) based on the IP address parameter. For example, use fx:fdqn(${pattern1.AlertedMdrProdInstance}) to convert the host name of the product from which the alert originated in the first event pattern to a fully qualified domain name. For example, if the property value is server5 (in the ca.com domain), the function would convert the value to server5.ca.com.
- **fx:fqdn-local**
  Returns the fully qualified local host name.
- **fx:ip-conversion**
  Returns the IP address based on the host name parameter. For example, fx:ip(${pattern1.AlertedMdrProdInstance}) converts the AlertedMdrProdInstance value to the system IP address.
- **fx:ip-local**
  Returns the IP address for the local host.
- **fx:localhost**
  Returns the local host name.

The conversion functions convert a string to the function format (fdqn or IP), while the local functions return the local host in the function format.

> **NOTE**
> The IP functions return an IPv4 or IPv6 address, depending on the system IP stack.

Date and Time

The following functions are categorized as Date and Time functions:

- **fx:xsDate-conversion**
  Returns the XML standard date based on the date string and format parameters.
- **fx:xsDate-epoch**
  Returns the current date based on the epoch seconds parameter.
- **fx:xsDate-now**
  Returns the current date.
- **fx:xsDurationFromMillisec**
  Returns an XML schema duration string based on the milliseconds parameter. This constructor function takes a value of milliseconds as an argument; for example, fx:xsDurationFromMilliSec(7545). The return value represents a duration of time; for example, P0DT0H0M7S. The format of the return value is *PnDTnHnMnS*, where *nD* is the number of days, *T* is the separator between date and time, *nH* is the number of hours, *nM* is the number of minutes, and *nS* is the number of seconds.
- **fx:xsDurationFromSec**
  Returns an XML schema duration string based on the seconds parameter. This function takes a value of seconds as an argument; for example, fx:xsDurationFromSec(988). The return value represents a duration of time; for example, P0DT0H16M28S. The format of the return value is *PnDTnHnMnS*, where *nD* is the number of days, *T* is the separator between date and time, *nH* is the number of hours, *nM* is the number of minutes, and *nS* is the number of seconds.
- **fx:xsTime-epoch**
  Returns the current time based on the epoch seconds parameter.
- **fx:xsTime**
  Returns the current time.
- **fx:xsdateTime-conversion**
  Returns the XML standard date and time based on the date and time string and format parameters.
- **fx:xsdateTime-epoch**
  Returns the date and time based on the epoch seconds parameter.
- **fx:xsdateTime-now**
  Returns the current date and time.

String

The following functions are categorized as String functions:

- **fx:toLower**
  Returns a lowercase string based on the mixed case string parameter.
- **fx:toUpper**
  Returns an uppercase string based on the mixed case string parameter.

Other

The following functions do not fall under any of the above categories:

- **fx:toUri**
  Returns a uniform resource identifier based on the file path.
- **fx:uniqueidentifier**
  Returns a unique identifier.
- **fn:Parse**
  Returns a parsed string based on the regex parameter.

For example, use fn:Parse(${pattern1.AlertedMdrElementID},'.*:(.*)') to parse out the first half of the AlertedMdrElementID property in the first event pattern. A property value of SA_Server:UC_Server would appear as simply UC_Server after applying this function.

Using nested functions is not supported. You cannot embed a function within another function.

Concatenating functions in the same cell also works. For example, multiple Parse functions separated by a space include both parsed values in the new event property.

Managing Event Policies

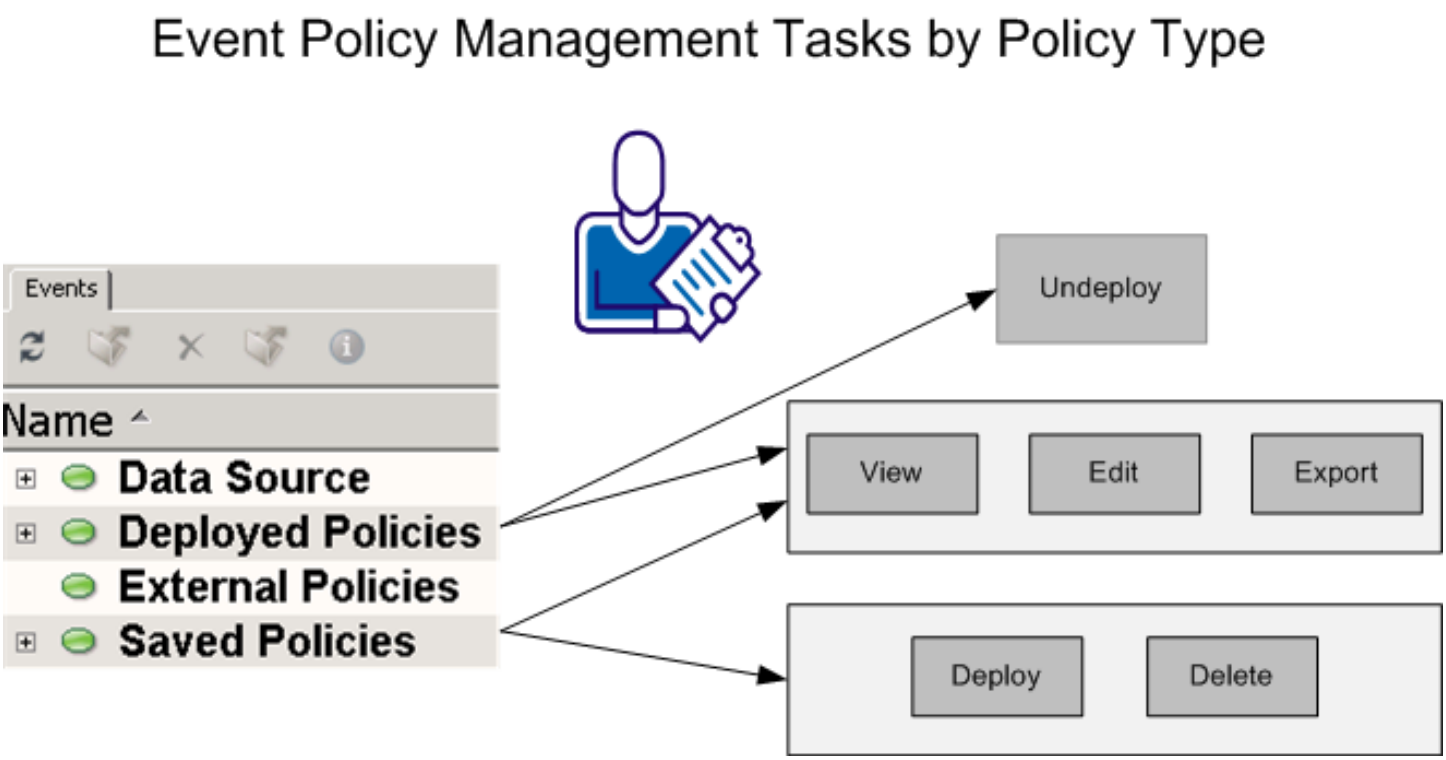
Contents

This section describes how administrators can view and manage event policies.

Event policies appear in one of the following sections in the Events tab of the Event Policies dialog:

- **Deployed Policies**
  Contains policies that are deployed to connectors. Expand the policy name to see the target connectors where the policy is deployed.
- **External Policies**
  Contains policies that are created or refined manually outside of the user interface.
- **Saved Policies**
  Contains saved policies that are currently undeployed.

The following graphic shows the available policy management tasks for each type of event policy:



Administrators can manage event policies in several ways:

- View event policy summary
- Edit an event policy
- Deploy or activate a saved, deactivated, or imported event policy
- Undeploy or deactivate an event policy
- Delete an event policy
- Export an event policy

NOTE
For information about managing external policies, which were created or refined outside of the user interface, see Manage Manual Event Policies.

View Event Policy Summary

You can view an event policy summary to see the important policy information, such as the scope, search pattern, and action.

Follow these steps:

1. Right-click a policy in any section of the Events tab, and select Summary.
   The Summary dialog opens with the following information:
   – Deployed data sources
   – Time scope
   – Search patterns for policy criteria
   – Event action type and action details
2. (Optional) Click Yes to print the policy.
   The Print dialog opens.
3. Select a printer and click OK.
   The policy information prints.

Edit an Event Policy

You can edit a saved or deployed policy to change any of the policy properties. Properties include connection parameters, enrichment properties, and so on. If you edit a deployed policy, redeploy the policy to apply the changes.

Follow these steps:

1. Select a policy in the Saved Policies section of the Events tab.
   The specified event patterns for the policy appear in the Event Search tab. You can click Search to view current search results.
2. Click Edit Policy or Edit Map Events on the Event Search tab.
   The Create Event Policy wizard opens and displays the New Policy page.
3. Make the necessary changes to action properties.
   If you change enrichment connection information in a deployed policy, restart the CA SAM Integration Services service on the connector system. A restart ensures that the change takes effect.
4. Perform one of the following actions:
   – If the policy is saved and you want it to remain saved, select Save policy only on the Select Data Sources page. Then complete the policy wizard.
   – If the policy deployed or you want to deploy a previously saved policy, deploy the policy from the Select Data Sources page. Then complete the policy wizard.

Deploy or Activate a Saved, Deactivated, or Imported Event Policy

In most cases, you deploy event policy during the policy creation process. However, the following situations require you to deploy a policy after its creation:

- You saved a policy without deploying it
- You imported a policy from another SA Manager
- You undeployed a deployed event policy to deactivate it, and you want to reactive the policy
- You made a change to a previously deployed policy

Follow these steps:

1. Right-click the policy in the Events tab and select Deploy Policy.
   The Deploy Policy dialog opens.
2. Select the connectors on which to deploy the policy, move them to the Selected Data Sources pane, and click OK.
   The saved, imported, or deactivated policy is deployed.

Undeploy or Deactivate an Event Policy

To stop the policy from evaluating events and running actions, undeploy the event policy. You can use the undeploy feature as a deactivation and retain the policy for activation later, or you can delete the undeployed policy.

> **NOTE**
> This functionality does not work on policies that are created or refined manually outside of the user interface.

Follow these steps:

1. Right-click the policy under Deployed Policies in the Events tab and select Deploy Policy.
   The Deploy Policy dialog opens.
2. Perform one of the following actions:
   – To undeploy from all data sources, move all data sources from the Selected Data Sources pane to the Available Data Sources pane. Click OK.
   – To undeploy from selected data sources, move specific data sources from the Selected Data Sources pane to the Available Data Sources pane. Click OK.

   > **NOTE**
   > You can also add data sources to a deployment using this method.

   The policy is undeployed from all or specific data sources. If you undeployed from all sources, it moves to the Saved Policies section in the Events tab. You can keep the policy as a saved policy for later activation or you can delete it.

Delete an Event Policy

You can delete a saved event policy to remove it permanently. To delete a deployed policy, undeploy it first.

Follow these steps:

1. If the policy is deployed, undeploy the policy.
   The policy appears under the Saved Policies section in the Events tab.
2. Right-click a policy under Saved Policies in the Events tab and select Delete Policy.
   A confirmation dialog opens.
3. Click Yes.
   The policy is deleted and disappears from the Events tab.

Export an Event Policy

You can export an event policy for use on another SA Manager.

Follow these steps:

1. Access the Event Policy dialog on the SA Manager that contains the policy to export.
2. Right-click an event policy and select Export.
   The Export Policy dialog opens.
3. Enter the destination SA Manager server name, and click OK.
   If the export succeeds, a 'Policy exported successfully' message appears. If the export fails, an 'Export failed' message appears with details about the reason for failure.
4. Access the Event Policy dialog from the Operations Console that manages data from the destination SA Manager.
   The imported event policy appears under Saved Policies in the Events tab.

5. Deploy the policy.

How to Manually Refine Event Policy

Contents

When you, as an administrator, require a functionality for an event policy that the Event Policy dialog does not support, you can manually refine event policy files. Examples of situations that require manual event policy customization are as follows:

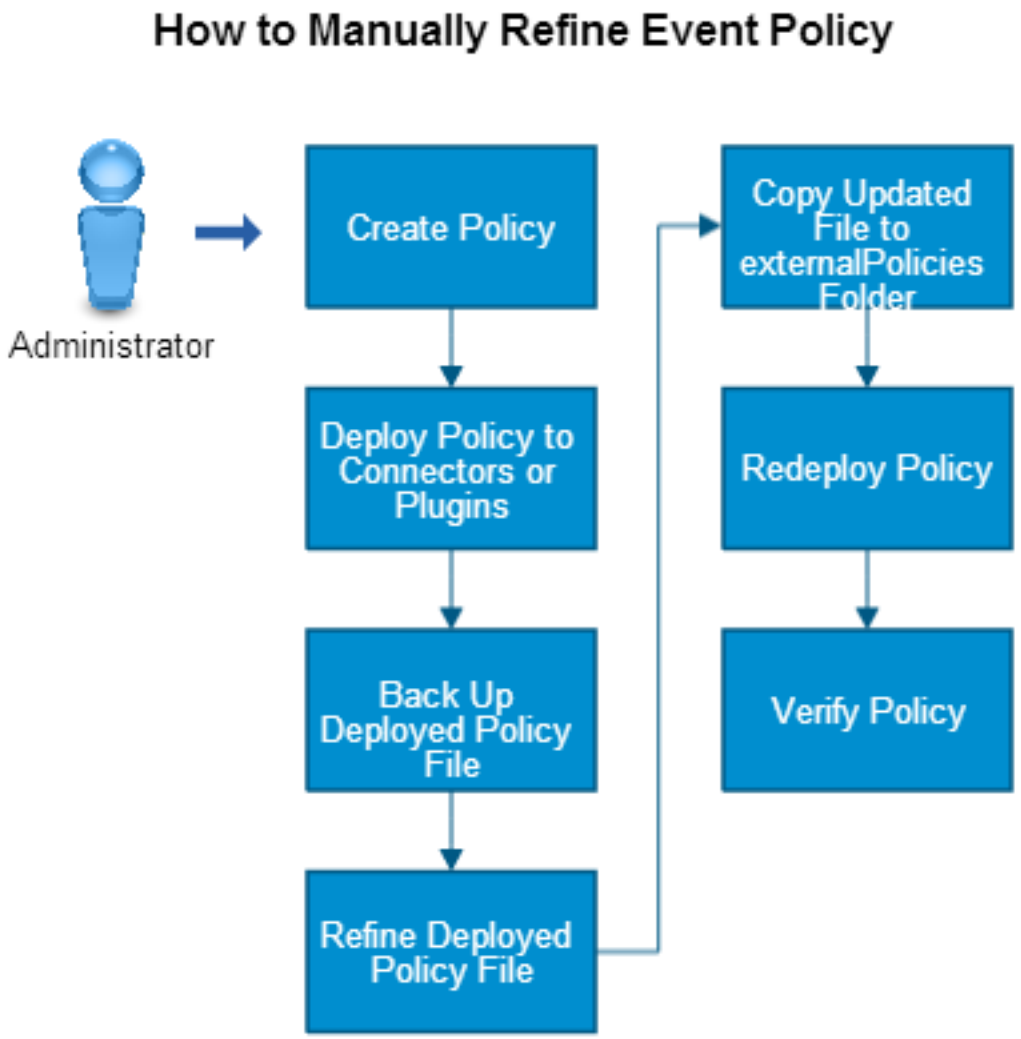
- Policies that require a combination of exclude and include filters
- Policies with an enrichment action that requires complex querying and enrichment assignments, involving operations such as SQL joins
- Policies that require more than three distinct search patterns
- Enrichment policies that are conditional based on the presence of certain event values

Unless one of these or another unsupported use case occurs, always use the functionality provided in the Event Policy dialog to define and deploy event policies.

To manually refine event policy, you must edit or create the appropriate event policy files using the <Evaluate> policy operation and embedded Drools rules.

Use this scenario to guide you through the process:

Figure 39: how to refine event policy



1. Create the basic framework for the event policy in the Event Policy dialog, omitting the part of the policy not supported by the user interface. For example, if you want to create a policy with multiple filters, create the policy with the correct search patterns and the initial filter in the Event Policy dialog so that the basic framework is in place for you to refine.

   NOTE
   You can create a custom event policy without first using the user interface, but the effort required and margin for error increases significantly.
2. Deploy the policy from the user interface to the appropriate connectors. You must deploy the policy for the event policy file to be created.
3. Make a backup copy of the deployed policy file stored in the <SOI_HOME>\resources\Core\Catalogpolicy\extensions directory (if the connector runs in the IFW) or <CONTAINER_HOME\container\Data\Core\Catalogpolicy\Extensions (if the connector runs on a CA Catalyst Container) on the connector system, and copy the backup to a separate directory on your system.

The deployed policy file is named according to the policy file name of the deployed connector and the event policy name (for example, sampleconnector_policy.service crash.xml).

4. Open the active version of the deployed policy file and make the necessary refinements.
   Rule evaluations and actions must adhere to the supported syntax for the Evaluate policy operation and the Drools language.

   NOTE
   Do not edit the similar file located at <SOI_HOME>\resources\EventManagement\Policies on the SA Manager. This file records the user interface policy selections. After you edit the deployed policy file, the <SOI_HOME>\resources\EventManagement\Policies file is out of date and obsolete.

5. Copy the edited policy file to the <SOI_HOME>\resources\EventManagement\externalPolicies directory on the SA Manager system, change the file extension from '.xml' to '.policy', and change the file name so that it matches the name of the corresponding file in the <SOI_HOME>\resources\EventManagement\Policies directory.
   Policies either created or refined manually must exist in this directory, so that the Event Policies dialog recognizes them as manually created or refined policies. The policy now appears under External Policies in the Events tab.
   Any time you edit a policy file and move the corresponding SA Manager record of that policy to the externalPolicies directory, it appears as an external policy in the user interface.
6. Right click the policy, select Deploy Policy, select the connectors on which to deploy, and click OK.
   The updated policy redeploys.
7. Verify or test that the policy is working correctly.

Evaluate Operation

The evaluate operation is the connector policy that enables the correlations and actions that are required for the event policy. The operation evaluates streams of events against defined rules (event search patterns) and runs workflow actions (event actions) when the rules are met.

Evaluate operations rely on the Drools language, which adheres to a different format than traditional connector policy and must be inserted in the evaluate operation. For more information about writing Drools event-based rules and workflow actions, see the following page for Drools documentation (version 5): http://www.jboss.org/drools/documentation.html.

Evaluate Property--Evaluate Events Based on Rules and Workflow Actions

Evaluate operations begin with an <Evaluate> property, which has the following basic syntax:

```
<Evaluate>
   <Field input="rule name" output="DRL">
      <!CDATA[
      <Drools rule>
   </Field>
   <Field input="action name" output="DRF">
      <!CDATA[
      <Drools action>
   </Field>
</Evaluate>
```

- **input**
  Defines the name of the event rule in the rule section and the name of the corresponding action in the action section.
- **output**
  Defines the type of Drools language to output. Use DRL for rules and DRF for actions.
- **Drools rule**
  Defines the event rule criteria in the Drools language.
- **Drools action**

Defines the event workflow action to run if the rule criteria are met. If the rule itself cannot perform the appropriate action, a workflow action is not required for every rule.

See the Example Event Policy File for an illustration of how to construct embedded Drools rules.

Example Event Policy File

The following example event policy file detects when a Windows service shuts down within 30 seconds after starting. These operations are tracked in separate events, so an event rule is required to correlate the events and trigger an appropriate action. The event policy creates an event to replace the other events with a message and severity that reflect the more serious nature of the situation. This evaluate operation contains a rule and does not require a separate action.

> **NOTE**
> This is a simple example that is easily configurable using the Event Policies dialog in the Operations Console. Always use the Event Policies dialog to create event policies, unless the interface does not support the operation. For information about creating more complex Drools rules, see the Drools documentation. For other syntax examples (for example, if you want to create a complex enrichment evaluate operation and need a frame of reference), create and deploy event policies from the Event Policies dialog and see the resultant syntax at <SOI_HOME>\resources\Core\Catalogpolicy\extensions or <CONTAINER_HOME\container\Data\Core\Catalogpolicy\Extensions.

The deployed event policy file for this example is as follows:

```
<Catalog version='1.0' globalextends='GLOBAL!'>
<EventClass name='Alert'>
    <Evaluate>
      <Field input='Service Crash' output='DRL'>
<![CDATA[
package com.ca.eventplus.catalog;
import com.ca.eventplus.catalog.util.EPEvent;
import java.util.HashMap;
declare EPEvent
  @role(event)
end

rule "Service Crash
no-loop true
when
pattrn1 : EPEvent((alertedMdrElementID=="?" && message matches ".*entered the running state.*") && reEvaluate!
="Service Crash")
pattrn2 : EPEvent((alertedMdrElementID=="?" && message matches ".*entered the stopped state.*") && reEvaluate!
="Service Crash", this after[0s,30s] pattrn1)
then
    pattrn1.createEvent("Service Crash",true,false,pattrn1,pattrn2);
end
]] >
      </Field>
    </Evaluate>
</EventClass>
<EventClass name='Service Crash' extends='Alert'>
   <FormatPostN>
      <Field  output='AlertType' format='Quality' input='' />
      <Field conditional='pattern1.AlertedMdrProduct'
      output='AlertedMdrProduct' format='{0}'
      input='pattern1.AlertedMdrProduct' />
```

```
      <Field conditional='pattern1.AlertedMdrProdInstance'
      output='AlertedMdrProdInstance' format='{0}'
      input='pattern1.AlertedMdrProdInstance' />
      <Field conditional='pattern1.AlertedMdrElementID'
      output='AlertedMdrElementID' format='{0}'
      input='pattern1.AlertedMdrElementID' />
      <Field conditional='pattern2.OccurrenceTimestamp'
      output='OccurrenceTimestamp' format='{0}'
      input='pattern2.OccurrenceTimestamp' />
      <Field  output='Severity' format='Major' input='' />
      <Field  output='Summary' format='Service Crash' input='' />
      <Field  output='MdrProduct' format='{0}' input='pattern1.MdrProduct' />
      <Field  output='MdrProdInstance' format='{0}'
      input='pattern1.MdrProdInstance' />
      <Field conditional='pattern1.MdrElementID' output='MdrElementID'
      format='{0}' input='pattern1.MdrElementID' />
   </FormatPostN>
 </EventClass>
 </Catalog>
```

NOTE
Some of the field attributes from the event policy are omitted from the Format syntax.

When the event policy deployment occurs, this file is generated at <SOI_HOME>\resources\Core\Catalogpolicy\extensions or <CONTAINER_HOME\container\Data\Core\Catalogpolicy\Extensions and named according to the deployed connector and policy name. In a typical manual refinement scenario, you deploy a simple policy then add the elements unsupported by the user interface in the deployed policy file. Using this method, you only have to work with and refine an existing policy file; you do not create one.

The input and output properties define the rule name and output. The Drools rule is embedded in the '![CDATA[' property. The Drools rule contains the following sections:

- **import**
  Defines Java methods to import for use in the rule. This declaration must include the EPEvent method, which describes the event properties that the Drools engine can use.
- **declare EPEvent**
  Declares EPEvent as an event role, enabling correlation between events.
- **rule "Service Crash"**
  Starts the event rule that contains the event search patterns.
- **when**
  Defines the rule criteria. The *when* clause in this example looks for the following events occurring within 30 seconds of one another:
  – An event with a message that contains the text 'entered the running state'
  – An event with the same alertedMdrElementID value as the 'pattrn1' event and a message that contains the text 'entered the stopped state'

  Note the format of the clause, specifically how it uses the EPEvent method to retrieve and evaluate the properties. Also note the syntax of the clause that defines the time interval between events.
- **then**
  Defines the action that runs when the criteria in the *when* clause are met. The *then* clause in this example creates an event that is based on the properties of the correlated events.
- **<FormatPostN>**
  Sets the properties for the new event. This syntax uses the Format operation to establish the new event properties, and the event class matches the name of the event policy. The AlertType, Summary, and Severity properties have new values that reflect the new event condition. The other properties use the values from the first or second event.

NOTE
Several properties have been omitted from this example.

For more examples and information about the syntax and requirements of the Drools language version 5, see the following page: http://www.jboss.org/drools/documentation.html.

Manual Policy Verification

To ensure that the manual changes compiled correctly and did not break the policy, verify your manual policy refinement after completion. To verify the manual policy, perform any of the following actions:

- Generate events that match the policy search criteria through the Universal connector to see whether the policy detects the events and performs the expected action.
- View the eventManagement.log file at <SOI_HOME>\jsw\logs for event policy errors.
- Enable detailed transformation logging to view potential transformation errors that the policy caused.

Manage Manual Event Policies

As long as you copy the deployed policy file for manually refined event policies to the <SOI_HOME>\resources\EventManagement\ExternalPolicies directory, they appear in the External Policies folder on the Events tab. Policies in this folder do not support management operations in the Event Policies dialog. You manually manage refined policies directly in the file system.

To undeploy a manual policy, remove the policy extension from the <SOI_HOME>\resources\Core\Catalogpolicy\extensions or <CONTAINER_HOME\container\Data\Core\Catalogpolicy\Extensions directory, retain the policy in a separate directory for future redeployment, and restart the affected connector. To redeploy the same policy, copy the file back into the extensions directory and restart the affected connector or plug-in service.

To delete a manual policy, delete the policy file from the <SOI_HOME>\resources\Core\Catalogpolicy\extensions or <CONTAINER_HOME\container\Data\Core\Catalogpolicy\Extensions directory.

To change the connectors on which the policy is deployed, perform one of the following actions:

- Change the *connectorname*_policy prefix in the deployed policy file name to match the name of the policy file for the new target connector. Restart both affected connectors (the previous and new deployment).
- Clone the event policy file. Change its file name prefix to match the name of the policy file for the new target connector. Restart the connector service.

  NOTE
  These instructions assume that the new target connector exists on the same system as the current target connector. If the new connector is on a different system, perform these actions across systems.

Manual Policy Scenario Sequencing Exclude and Include Filter Combinations

The filter event policy action is most valuable when you use a combination of exclude and include filters to gain more granular control over which events make it to the Operations Console as alerts. The Create Event Policy wizard supports a default sequencing rule that automatically evaluates include filters before exclude filters when multiple filters are deployed on the same connector. This default rule accommodates the most common filter combination use cases, where you exclude a large set of events and include a small important subset of those events. Manual policy refinement is required to configure a filter evaluation sequence other than the default.

This example combines three filters that do the following:

- Exclude response time alarms that include the word Minor followed by the word App3
- Include response time alarms that include the word Minor in their message
- Exclude all response time alarms that do not match the other filters

The listed order is correct, but the default filter sequencing rules would evaluate the include filter first. Therefore, all Minor alarms for App3 would be included, because they would be explicitly included and never evaluated by the specific exclude filter. This scenario shows how you manually refine the policies to configure the correct filter evaluation order.

Follow these steps:

1. Access the Event Policy dialog.
2. Enter the following search pattern in the Event Pattern 1 field, select ANY event occurs in the Additional Criterion pane, and click Search:
   ```
   matches ( Message, 'ResponseTime:Minor:App3' )
   ```
   This search pattern matches events with 'ResponseTime:Minor:App3' in the Message property.
3. Click Create Policy, name the policy ExcludeFilter1, select Filter Events and Exclude, and click Next.
   The Select Data Sources page opens.
4. Select Save and Deploy policy, move the appropriate connector to the Selected Data Sources pane, click Next, and click Finish on the Confirm page.
   The policy is deployed on the specified connector. The policy excludes any events with 'ResponseTime:Minor:App3' in the message property.
5. Return to the Event Policy dialog, enter the following search pattern in the Event Pattern 1 field, select ANY event occurs in the Additional Criterion pane, and click Search:
   ```
   matches ( Message, 'ResponseTime:Minor:.*' )
   ```
   This search pattern matches events with 'ResponseTime:Minor:' in the message property followed by any text after the last colon. Events matching this search pattern would also match the pattern entered in Step 2.
6. Click Create Policy, name the policy IncludeFilter, select Filter Events and Include, and click Next.
   The Select Data Sources page opens.
7. Select Save and Deploy policy, move the same connector as Step 4 to the Selected Data Sources pane, click Next, and click Finish on the Confirm page.
   The policy is deployed on the specified connector. The policy explicitly includes events with 'ResponseTime:Minor:' followed by any extra text in the message property.
8. Return to the Event Policy dialog, enter the following search pattern in the Event Pattern 1 field, select ANY event occurs in the Additional Criterion pane, and click Search:
   ```
   matches ( Message, 'ResponseTime:.*:.*' )
   ```
   This search pattern matches events with 'ResponseTime:' in the message property followed by any text after the colon. Events matching this search pattern would also match the patterns entered in Steps 2 and 5.
9. Click Create Policy, name the policy ExcludeFilter2, select Filter Events and Exclude, and click Next.
   The Select Data Sources page opens.
10. Select Save and Deploy policy, move the same connector as Step 4 and 7 to the Selected Data Sources pane, click Next, and click Finish on the Confirm page.
    The policy is deployed on the specified connector. The policy excludes all events with 'ResponseTime:' followed by any extra text in the message property.
    By default, the combined policy evaluates the include filter first, followed by the exclude filters in random order. You must refine the policy to ensure that the policy evaluates the filters in the order in which you deployed them so that no events are erroneously included or excluded.
11. Access the connector system, and back up the deployed event policy files at SOI_HOME\resources\Core\Catalogpolicy\extensions or <CONTAINER_HOME\container\Data\Core\Catalogpolicy\Extensions.
12. Open the *connectorname*.excludefilter1.xml policy file, and add a sequence number to the filter operation as follows:
    ```
    <FilterPostN>
        <Field input='internal_suppresseventExclude Filter 1' pattern='^true$' type='exclude' seqnumber='1' />
    </FilterPostN>
    ```
    Filters with a seqnumber property take precedence over all other filters without a seqnumber property defined. Assign this filter a seqnumber of 1 to ensure that the event policy always evaluates it first.

13. Repeat Step 12 in the files for the IncludeFilter and ExcludeFilter2, adding seqnumber=2 to the Include Filter and seqnumber=3 to the Exclude Filter 2.
    With these changes, the policy evaluates the filters in the correct order as follows:
    – Excludes Minor App3 response time events
    – Includes Minor response time events (except for those excluded by the previous filter)
    – Excludes response time events (except for Minor ones included by the previous filter)
    Events matching the first filter are excluded and therefore not evaluated by the subsequent include filter. Events that do not match the first filter and do match the second filter are included and therefore not evaluated by the subsequent exclude filter. Events that do not match the first two filters and do match the last filter are correctly excluded.
14. Copy the refined files to the <SOI_HOME>\resources\EventManagement\ExternalPolicies directory on the SA Manager system so that the user interface can identify them as manually refined policies, change the file extensions from .xml to .policy, and change the file names so that they match the name of the corresponding files in the <SOI_HOME>\resources\EventManagement\Policies directory.
    The policy now appears under External Policies in the Events tab. Any time you edit a policy file and move the corresponding SA Manager record of that policy to the ExternalPolicies directory, it appears as an external policy in the user interface.
15. Right click the policy, select Deploy Policy, select the connectors on which to deploy, and click OK.
    The updated policy redeploys and evaluates the filters in the correct order using the defined sequence numbers.

Manual Policy Scenario Making an Enrichment Conditional

This scenario is a simple map enrichment that adds text in the User Attribute 1 property to function as a key for alert queue criteria. The policy requires manual refinement to add syntax that makes the enrichment conditional. If the enriched property already has a manually defined alert queue key, the enrichment does not occur to avoid overwriting the current value.

Follow these steps:

1. Access the Event Policy dialog.
2. Enter the following search pattern in the Event Pattern 1 field:
   ```
   matches(Summary,'MANAGEMENT AGENT LOST')
   ```
3. Select 'ANY events occurs' and click Search to obtain results for previewing the enriched event that is based on existing events.
4. Click Create Policy, name the policy AlertQueueEnrichment, select Enrich Events, and click Next.
5. On the Enrichment Configuration page, select Map only and click Next.
6. On the Enrichment Policy page, enter ACTION_QUEUE in the Assigned Value cell for User Attribute 1.
   This policy enriches an event with a summary that contains MANAGEMENT AGENT LOST with a value of ACTION_QUEUE in User Attribute 1.
   The Select Data Sources page opens.
7. Select Save and deploy policy, move the applicable connector to the Selected Data Sources pane, click Next, and click Finish on the Confirm page.
   The policy is deployed on the selected connector. The policy always performs the User Attribute 1 enrichment when the search pattern matches. However, you want to perform the enrichment only if a queue value does not exist in User Attribute 1.
8. Access the deployed connector system, and back up the deployed event policy file at <SOI_HOME>\resources\Core\Catalogpolicy\extensions or <CONTAINER_HOME\container\Data\Core\Catalogpolicy\Extensions.
9. Open the active policy file and replace the <EventClass> content with the following elements:
   ```
   <ParsePostE>
       <Field output='temp_parse_userAttribute1' pattern='(.*QUEUE)' input='userAttribute1' />
   </ParsePostE>
   <FormatPostE>
   ```

```
    <Field conditional='!temp_parse_userAttribute1' output='userAttribute1' format='ACTION_QUEUE'
 input='' />
</FormatPostE>
```

This manual edit makes the enrichment conditional. The enrichment only occurs if the User Attribute 1 property does not already contain a manually assigned queue name.

10. Copy the refined file to the <SOI_HOME>\resources\EventManagement\ExternalPolicies directory on the SA Manager system. The user interface can then identify the policy as a manually refined policy, change the file extension from .xml to .policy, change the file name so that it matches the name of the corresponding file in the <SOI_HOME>\resources \EventManagement\Policies directory.
    The policy now appears under External Policies in the Events tab. When you edit a policy file and you move the corresponding SA Manager record of that policy to the externalPolicies directory, the policy appears as an external policy in the user interface.
11. Right-click the policy, select Deploy Policy, select the connectors on which to deploy, and click OK.
    The updated policy redeploys, and the manual edits take effect.
12. Create an alert queue with the criteria of User Attribute 1=ACTION_QUEUE.
    Events that a policy enriches appears in the created alert queue.

How to Add External Extensions to CI Types

You can customize connector policy by adding external extensions to CI Types. The sections you extend correspond with a section in the base policy; the base policy process first and then the extensions process. For example, if the policy maps the product domain property *ipv4binary* to the *primaryIPv4address*, but you want the product domain property *ipv4string*, you would override that in your extensions file. In this case, it would first set the *primaryIPv4Address* to the value of *ip4binary*, then since the extension was merged to the end of the section, it would set *primaryIPv4Address* to the value of *ipv4string*. For this example, if you wanted to extend a section that is in the delivered policy *<EventClass name="ComputerSystem" extends="Item">* , you would need a section called *<EventClass name="ComputerSystem" extends="Item">.*

Add an External Extension to CI Types

To customize your connector policy, add an external extension to CI types.

Follow these steps:

1. Create a file in the extensions directory under the following path:

```
IFW = <SOI_HOME>\resources\Core\Catalogpolicy\extensions

Container = <CATALYST_CONTAINER>\container\Data\Core\Catalogpolicy\Extensions
```

> **NOTE**
>
> Ensure that **\container\Data\Core\Catalogpolicy\Extensions** directory must be created after the installation of the container.

2. Copy the following template for the CI Extension:

```
<Catalog version='1.0'>



      <!--EventClass name="ComputerSystem" extends="Item">
```

```
                <Format>

                        <Field conditional="name" output="ComputerName" format="{0}"
 input="name" />

               </Format>

      </EventClass-->

 </Catalog>
```

3. Manually edit the extension policy. You must add an EventClass section that matches the EventClass section of the delivered connector policy. You then add the policy lines to change, overwrite, and add additional attributes.
   The following example shows delivered policy where it assigns the value of the domain manager property *name* to the USM property *ComputerName*:

```
 <EventClass name="ComputerSystem" extends="Item">

      <Format>

                <Field conditional="name" output="ComputerName" format="{0}"
 input="name" />

     </Format>

 </EventClass>
```

   In this case, the intention is to customize the policy to assign the USM property *ComputerName* with the value of of the domain manager property *dns_name.* Additionally, to assign the value of the domain manager property *name* to the USM property *Description,* you would have an extension that extends this delivered policy section with this:

```
 <EventClass name="ComputerSystem" extends="Item">

       <Format>

                 <Field conditional=" dns_name" output="ComputerName" format="{0}"
 input="dns_name" />

                 <Field conditional="name" output="Description" format="{0}"
 input="name" />

      </Format>

 </EventClass>
```

4. Restart the Connector for the policy to take effect.

Event Management Example Scenarios

This section provides end-to-end example scenarios for common event management use cases.

Event Management Example 1: Filter Duplicate Events from Integrated Domain Managers

This scenario illustrates how you can filter duplicate events received from connectors with integrated domain managers so that one consolidated alert appears for each reported condition.

Several domain managers for which you may have connectors could already be integrated with one another. Examples of common domain manager integrations include the following:

- CA Spectrum and CA eHealth

For example, CA Spectrum might already be feeding its alarms into CA eHealth when the two products are integrated. If you have CA Spectrum and CA eHealth connectors installed, you could receive an alert for the original CA Spectrum alarm and an alert for the CA eHealth alert representing the same CA Spectrum alarm. Duplicate alerts in CA SOI caused by cross-domain integrations require extra time to clear, could cause confusion for operators, and could provide an inaccurate report of CI severity.

This scenario assumes that you have integrated CA eHealth and CA Spectrum, so that CA eHealth alarms are sent to CA Spectrum. It does the following:

- Creates a new event that duplicates the CA Spectrum event (which represents the integrated CA eHealth event) and updates the event message to reflect the consolidation
- Discards the original duplicate CA eHealth and CA Spectrum events in the same policy

Follow these steps:

1. Enter the following in the Event Pattern fields in the Event Search tab:
   ```
   MdrProduct='CA:00005' and Message=?
   MdrProduct='CA:00002' and Message=?
   ```
   This search criteria returns events from CA eHealth and CA Spectrum that have identical message text.

   NOTE
   The scenario assumes that the event message is the same for events from CA eHealth and integrated CA eHealth events from CA Spectrum. If the messages differ slightly, a more fine-grained search is required.
2. Select ALL events occur within 120 seconds in the Additional Criterion pane.
   This selection specifies that the events must occur within two minutes of each other.
3. Click Search.
   The search results appear.
4. Click Create Policy.
   The Create Event Policy wizard opens and displays the New Policy page.
5. Enter CreateConsolidatedEvent in the Policy Name field, select Create New Event, and click Next.
   The Create New Event page opens.
6. Do the following:
   – Edit the Message event property as follows and click Next:
     ```
     ${pattern1.Message} - consolidated
     ```
     This change appends the Message property with a notice that the event is a consolidated version of multiple events.
   – Set the mdrElementID to fx:uniqueidentifier().
     This changes helps ensure that a new event is created with a unique mdrElementID value.

   The Select Data Sources page opens.
7. Select Save and Deploy policy, move the Mid-tier connector to the Selected Data Sources pane, and click Next.

NOTE
Assignment to the Mid-tier connector is required, because the search requires event correlation across connectors. Assigning to the CA eHealth and CA Spectrum connector would prevent the events from correlating across domain managers. However, the MdrProduct values in the search patterns prevent the search from occurring on connectors other than CA Spectrum and CA eHealth.

The Confirm page opens.

8. Confirm the policy information and click Finish.
   The policy is deployed. This policy creates a new event to represent events duplicated in CA eHealth and CA Spectrum instances that are integrated each other. The event uses properties from the source CA Spectrum event and appends the message with a notification that the event is consolidating duplicates.
9. Select the deployed policy in the Events tab, and click Edit Policy.
   The Create Event Policy wizard opens and displays the New Policy page.
10. Enter FilterIntegratedEvents in the Policy Name field, select Filter Events and then Exclude, and click Next.
    The Select Data Sources page opens.
11. Select Save and Deploy policy, retain the Mid-tier connector in the Selected Data Sources pane, and click Finish.
    The filter event policy is deployed. This policy discards the original CA eHealth and CA Spectrum events, so that only the created event becomes an alert in the Operations Console. The created event is not discarded, because the addition to the Message property causes its Message value to be different from the original events.

Event Management Example 2: Enrich Events with Related CI Information from the Persistence Store

This scenario illustrates how you can enrich events with information extracted from the Persistence Store database. CIs stored in CA SOI in the Persistence Store database contain a varied set of USM properties depending on the CI type. These properties contain useful information that related alerts do not contain, such as the following:

- Label
- Memory
- Processor type and speed
- CI description
- Operating system type

Key CI properties such as vendor, IP address, DNS name, and others are available to use as criteria in alert queues and escalation policy and in event searches and policies when you enable the default Persistence Store enrichment.After you enable the Persistence Store enrichment, all CI properties are available for use in enrichments without configuring a connection to the Persistence Store database. Therefore, a map enrichment can enrich events with any associated CI property when Persistence Store enrichment is enabled.

NOTE
Persistence Store enrichment is available for CA Catalyst connectors only.

This scenario assumes that you are monitoring computer system CIs in your enterprise with the following operating systems:

- Windows
- Windows Vista
- Linux
- Linux Fedora
- Android (mobile operating system)

A different technician is responsible for resolving problems with each operating system type, and you want to organize alerts on those computer system CIs by their operating system type. The scenario does the following:

- Enriches all events associated with a computer system CI with the operating system type stored in the Persistence Store
- Creates alert queues for each operating system type, so that the assigned technicians can quickly find and diagnose problems with their computer systems

> **NOTE**
> You can use this basic scenario as a frame of reference to enrich events with any property in the Persistence Store.

To implement Scenario Two: Enrich events with related CI information from the Persistence Store

1. Access the CA Catalyst Registry from a web browser, and navigate to the topology\physical*nodename*\modules\configuration*connectorname*.conf file.
2. Click Edit as Text.
   The file contents appear.
3. Add the following property to the <Config> section of the file, and click Save Content:
   ```
   EnrichFrmPS='ON'
   ```
4. Restart the CA Catalyst Container CatalystConnector service on the connector system.
   The Persistence Store enrichment occurs for all ensuing events that are collected from that connector.
5. (Optional) Repeat Steps 1-4 to enable Persistence Store enrichment for other connectors.
6. Do the following and click Search:
   – Select the Mid-tier connector entry in the Data Source list, or specific connectors if you did not enable Persistence Store enrichment on all connectors.
   – Leave the Event Patterns blank on the Event Search tab.
   – Select 'ANY event occurs' in the Additional Criterion pane.

   Search results appear. Blank search patterns perform the specified actions on all events in the scope. Scoping to the Mid-tier connector makes all events available to the search.
7. Click Create Policy.
   The Create Event Policy wizard opens and displays the New Policy page.
8. Enter PSEnrichment in the Policy Name field, select Enrich Event, and click Next.
   The Enrichment Configuration page opens.
9. Select Map only and click Next.
   The Enrichment Policy page opens.
10. Add ${pattern1.AssociatedCI_PrimaryOSType} in the Assigned Value cell corresponding to the User Attribute 1 Event Property in the Enrichment Property Assignment table, and click Next.
    This configuration enriches events with the value of the PrimaryOSType associated CI property when an associated computer system CI is detected.

    > **NOTE**
    > You can change the name of the User Attribute 1 property if you want it to accurately represent the enrichment property. However, this property appears under its original name in the Event Policy dialog, even if you renamed it. Assigning a value to the original name properly displays the value under the renamed property in the Operations Console.

    The Select Data Sources page opens.
11. Select Save and Deploy policy, move the Mid-tier connector to the Selected Data Sources pane, and click Next.
    The Confirm page opens.
12. Confirm the policy information and click Finish.
    The policy is deployed. Any event generated for a computer system CI is enriched with the CI Primary OSType property in the User Attribute 1 event property.
13. Select the Alert Queues tab and click Add.
    The New Alert Queue dialog opens.
14. Do the following and click Next:

- Enter Windows OS Alerts in the Queue Name field.
- Select User Attribute (1) in the Attribute drop-down list.
- Select Starts With in the Comparison Type drop-down list.
- Enter Windows in the Attribute Value field.
- Click Add.

The queue criteria adds alerts with a User Attribute 1 property value that starts with Windows.

15. Complete the alert queue creation process and click Finish on the Confirm page.

    NOTE
    You can assign escalation policies and user group access to the queue.

    The alert queue is created. The alerts for computer systems CIs with the Primary OSType values of Windows and Windows Vista appear in this queue.
16. Repeat Steps 12-14 to create queues for Linux and Android operating system alerts. During criteria definition, give each queue a unique name, and enter Linux for the Linux queue and MobileOS for the Android queue in the Attribute Value field.
    All computer system alerts are grouped into operating system-specific queues using the enrichment value.

Event Management Example 3: Create a New Event to Indicate a Crashing Service

This scenario illustrates how you can correlate events that occur together to indicate a different or more severe condition than when the events occur separately. You create an event to indicate the correlated condition. Several conditions are detectable only with the correlation of separate event occurrences or the same event. The follow events are such situations:

- A persistent CPU or memory deficiency (which can be more severe than an occasional spike)
- Servers in a cluster going down at the same time
- Any situation where the root cause of a condition may not be evident through service or CI hierarchy and relationships

Correlating events lets you represent the true condition in a new event that you can use to trigger escalation policy to resolve the problem.

This scenario assumes that you have connectors monitoring running services, and you have had problems in the past with services that shut down immediately after they are started. It does the following:

- Detects when service startup and shutdown occur within a short amount of time from one another
- Creates a new event that increases the severity and modifies the event summary to indicate the problem
- Discards the original events
- Creates escalation policy that triggers based on the new event summary

Follow these steps:

1. Select the Mid-tier connector in the Data Source list and enter the following in the Event Pattern fields in the Event Search tab:

   ```
   AlertedMdrElementID=? and matches (Summary,'service has started')
   AlertedMdrElementID=? and matches (Summary,'service has stopped')
   ```

   This search criteria returns events from the same connector and CI, where the first event summary contains the text 'service has started', and the second event summary contains the text 'service has stopped'.
2. Select ALL events occur within 45 seconds in the Additional Criterion pane, and select the Sequence enforced check box.
   This selection specifies that the events must occur within 45 seconds of each other and that the 'service has started' event must occur before the 'service has stopped' event.
3. Click Search.
   The search results appear.

4. Click Create Policy.
   The Create Event Policy wizard opens and displays the New Policy page.
5. Enter ServiceCrash in the Policy Name field, select Create New Event, and click Next.
   The Create New Event page opens.
6. Edit the properties of the new event as follows and click Next:
   – Set the Severity to Critical.
   – Set the Summary to 'Service crashing immediately after startup'.
   – Retain the defaults for other properties, which inherits the properties of the event from the first pattern.
   – Set the mdrElementID to fx:uniqueidentifier().

   This change increases the severity to critical and changes the summary to a specific indication of the correlated problem.
   The Select Data Sources page opens.
7. Select Save and Deploy policy, move the Mid-tier connector to the Selected Data Sources pane, and click Next.

   NOTE
   If only certain connectors are monitoring services, you can assign to specific connectors instead.

   The Confirm page opens.
8. Confirm the policy information and click Finish.
   The policy is deployed.
9. Select the deployed policy in the Events tab, and click Edit Policy.
   The Create Event Policy wizard opens and displays the New Policy page.
10. Enter FilterCorrelatedEvents in the Policy Name field, select Filter Events and then Exclude, and click Next.
    The Select Data Sources page opens.
11. Select Save and Deploy policy, retain the Mid-tier connector in the Selected Data Sources pane, and click Finish.
    The filter event policy is deployed. This policy discards the original service startup and shutdown events, so that only the created event becomes an alert in the Operations Console.
12. Select Tools, Escalation Policies and Actions.
    The Escalation Policies and Actions dialog opens.
13. Click Add.
    The Alert Escalation Policy Editor dialog opens.
14. Enter Service Crash Policy in the Name field and click the Attributes tab.
    A pane opens for specifying alert attribute-specific criteria.
15. Select Summary in the Attribute drop-down list, Equal To in the Comparison Type drop-down list, and enter 'Service crashing immediately after startup' in the Attribute Value field. Click Add.
    The policy triggers when an alert occurs with the summary you specified for the new event.
16. Select the Policy Actions tab and click New.
    The Escalation Action Editor dialog opens.
17. Enter Create Service Crash Ticket in the Action Name field and select Create Ticket in the Action Type drop-down list.
    Tabs appear for specifying ticket properties.
18. Select Summary in the Property Name drop-down list, enter 'Service is crashing immediately after startup' in the Property Value field, and click Add.
    The ticket summary matches the alert summary.
19. Click OK.
    The action is saved.
20. Click OK on the Alert Escalation Policy Editor dialog.
    CA SOI saves the escalation policy. When the deployed event policy detects the correlated event condition, the following actions occur:

  - A new Critical event is created with a descriptive summary
  - The original events are discarded
  - When the event appears in the Operations Console as an alert, it triggers an escalation policy that creates a help desk ticket

Event Management Example 4: Combine a Create Event Action with an Enrichment Using Reevaluation

This scenario illustrates how you can reevaluate an event on which an action has already occurred when multiple actions are required to optimize the resultant alert. The Reevaluate option on the Create Event Policy dialog lets you send an event that has been created or enriched by an event policy back through the policy engine for evaluation by other event policies.

This scenario assumes that you have connectors monitoring vital ComputerSystem CIs, and that the default alerts that are generated are not of the quality required for a prompt diagnosis and resolution. The scenario does the following:

- Detects when a pattern of events occurs that indicates a ComputerSystem CI is down
- Creates a new event that increases the severity and modifies the event summary to indicate the problem
- Discards the original events
- Reevaluates the created event and enriches it with contact information for the problematic CI from a database table

Follow these steps:

1. Enter the following in the Event Pattern fields in the Event Search tab:
   ```
   AlertedMdrElementID=? and Summary='Management agent lost contact'
   AlertedMdrElementID=? and Summary='Device response exceeds threshold'
   ```
   This search criteria returns events from the same connector and CI, where the first event summary is 'Management agent lost contact', and the second event summary is 'Device response exceeds threshold'.
2. Select ALL events occur within 30 seconds in the Additional Criterion pane.
   This selection specifies that the events must occur within 30 seconds of each other. When occurring together, these events are strong indications that the associated CI is down.
3. Click Search.
   The search results appear.
4. Click Create Policy.
   The Create Event Policy wizard opens and displays the New Policy page.
5. Enter CreateEventDeviceUnresponsive in the Policy Name field, select Create New Event, and click Next.
   The Create New Event page opens.
6. Select the Reevaluate check box.
   This selection specifies to reevaluate the created event against other event policies.
7. Edit the properties of the new event as follows and click Next:
   - Set the Severity to Fatal.
   - Set the Summary to DEVICE UNRESPONSIVE.
   - Set the Message to 'Device fn:Parse(${pattern1.AlertedMdrElementID}) is unresponsive'.

     NOTE
     This value uses the Parse function to include the name of the CI in the message using the AlertedMdrElementID value returned by the first event pattern. For example, if the AlertedMdrElementID value in the first event is Server5, the output value of the Message property would be 'Device Server5 is unresponsive'.
   - Set the mdrElementID to fx:uniqueidentifier().
   - Retain the defaults for other properties, which inherit the properties of the event from the first pattern.

This change increases the severity to fatal and changes the summary and message to a more specific indication of the problem.
The Select Data Sources page opens.

8. Select Save and Deploy policy, move the Mid-tier connector to the Selected Data Sources pane, and click Next.
   The Confirm page opens.
9. Confirm the policy information and click Finish.
   The policy is deployed.
10. Select the deployed policy in the Events tab, and click Edit Policy.
    The Create Event Policy wizard opens and displays the New Policy page.
11. Enter FilterOriginalEvents in the Policy Name field, select Filter Events and then Exclude, and click Next.
    The Select Data Sources page opens.
12. Select Save and Deploy policy, retain the Mid-tier connector in the Selected Data Sources pane, and click Finish.
    The filter event policy is deployed. This policy discards the original event pattern, so that only the created event becomes an alert in the Operations Console.
13. Return to the main Event Policy dialog, and enter the following search pattern in the Event Pattern 1 field:
    ```
    Summary='DEVICE UNRESPONSIVE'
    ```
    This search pattern returns the event created by the create event policy, on which you enabled reevaluation.
14. Click Create Policy.
    The Create Event Policy wizard opens and displays the New Policy page.
15. Enter EnrichEventDeviceUnresponsive in the Policy Name field, select Enrich Event, and click Next.
    The Enrichment Configuration page opens.
16. Select JDBC in the Type drop-down list, enter connection settings for the database in the fields, and click Next. This example assumes the following:
    - The database with the information required for the enrichment is a Microsoft SQL Server database, and the connection information follows the conventions described in JDBC Connection Examples and accessible from the MS SQL entry in the Templates drop-down list.
    - The database server name is dbserver1.
    - The database name is Contacts.
    - The database table with the required contact information is ContactTable.

    The Enrichment Policy page opens.
17. Do the following in the Parameter Configuration table:
    - Use the right-click menu to add DeviceName in the first cell of the Input Parameter column.
    - Use the right-click menu to add ${pattern1.AlertedMdrElementID} in the corresponding cell in the Assigned Value column.

    This configuration queries the ContactTable table from the Contacts database for instances where the AlertedMdrElementID property in the created event matches the DeviceName database column value, which matches the created event to its associated CI in the database. If a match is not found, the enrichment does not occur.
18. Do the following in the Enrichment Property Assignment table and click Next:
    - Use the right-click menu to add ${ContactEmail} in the Assigned Value cell corresponding to the User Attribute 1 property.
    - Use the right-click menu to add ${ContactName} in the Assigned Value cell corresponding to the User Attribute 2 property.

    This configuration enriches the User Attribute 1 and 2 properties of the created event with the values of the ContactEmail and ContactName database columns when the event's device name is matched in the database.

    NOTE
    You can change the name of the User Attribute properties if you want them to accurately represent the enrichment properties. However, these properties appear under their original names in the Event Policy dialog, even if you renamed them. Assigning values to these original names properly displays the values under the renamed properties in the Operations Console.

The Select Data Sources page opens.

19. Select Save and Deploy policy, move the Mid-tier connector to the Selected Data Sources pane, and click Next.

    NOTE
    You must use the Mid-tier connector for this scenario, because CA Catalyst connectors do not support database enrichments.

    The Confirm page opens.

20. Confirm the policy information and click Finish.
    The enrichment policy is deployed. This policy enriches the event created by the create event policy with contact information for the CI from a Contact database. When the event displays as an alert in the Operations Console, it contains an elevated severity, a more accurate description of the CI condition, and contact information in the User Attribute 1 and 2 properties for prompt assignment and resolution.
    You could use alert management functionality to further facilitate resolution of this high quality alert as follows:
    – Create an escalation policy based on the alert message that generates a help desk ticket or sends an email. For either action, you could use the enriched contact information to help ensure that the help desk ticket or email reaches the appropriate technician.
    – Create an alert queue that uses the modified or enriched property values to include the alert in its appropriate group, such as a queue for fatal alerts, a queue for a specific data source or service that manages the CI, or a queue for the assigned technician.

Event Management Example 5: Normalize Monitoring Traps

This scenario illustrates how you can normalize events from raw event sources to the USM alert format. The SNMP connector collects traps from all trap sources that send their traps to the configured trap destination. However, because traps from different sources have different formats, detailed policy does not exist to convert those traps to the USM alert format.

The trap source normalized in this scenario is CA Systems Performance for Infrastructure Managers (powered by the CA SystemEDGE agent). The CA SystemEDGE agent monitors objects and processes and sends traps when configured thresholds are breached. This scenario normalizes the aggregate state traps that are sent when a monitor entry configured for stateful monitoring detects a threshold breach. You can manage the state of important resources and processes in the Operations Console. The examples and subsequent alert queue creation focus on process monitoring traps.

NOTE
This procedure normalizes aggregate state traps, which are sent when a monitor entry is configured for stateful monitoring. Monitor and process monitor traps are sent when monitors are not configured for stateful monitoring. These traps are not covered in this scenario. For more information, see the CA SystemEDGE documentation.

Follow these steps:

1. Configure CA SystemEDGE to send traps to the SNMP connector system on the SNMP connector listening port (162 by default).
2. Either generate or confirm that process monitor traps have occurred. A raw event search must return results to create a normalization action for the traps.
3. Run a raw event search for CA SystemEDGE traps as follows:
   – Select Generic SNMP Traps in the Data Source list on the Events tab to scope the search to the SNMP connector.
   – Enter the following pattern in the Event Pattern 1 field in the Event Search tab:
   ```
   snmp_enterprise='1.3.6.1.4.1.546.1.1' and snmp_specificTrap='20'
   ```
   Aggregate state threshold breach traps from the CA SystemEDGE agent appear in the results table.
4. Click Map Events.
   The New Policy page opens.

5. Name the policy SystemEDGEMonitors, select Normalize Event, and click Next.
   The Normalize Event page opens.
6. Establish the following mappings in the Assigned Value cells and click Next:
   - **Mdr Element ID: ${pattern1.varbind-1.3.6.1.4.1.546.17.1.1.2.5}:${pattern1.varbind-1.3.6.1.4.1.546.17.1.1.3.5}: ${pattern1.varbind-1.3.6.1.4.1.546.17.1.1.4.5}**
     Maps the MdrElementID property to the monitor entry's object class, object instance, and object attribute.
     Example: Process://./OUTLOOK:Memory(KB)
   - **Occurrence Timestamp and Report Timestamp: fx:xsdateTime()**
     Maps these properties to the current time. Find this value by right-clicking the cell and selecting Functions, fx:xsdateTime-now.
   - **Alert Type: Risk**
     Maps the AlertType property to a static value of Risk.
   - **Severity: Use Map Function**
     Maps the Severity property to the Current State varbind value. Right-click the cell, select Map, and map the values for varbind-1.3.6.1.4.1.546.17.1.1.6.5 to valid USM Severity values as follows using the Map function:
     Value column: USM Value column
     - 1: Unknown
     - 2: Normal
     - 3|4: Minor
     - 5: Major
     - 6: Critical
     - 7: Fatal

       NOTE
       The Preview cell does not support map values derived through regular expressions. If the map value uses a regular expression, the Preview cell displays a message 'Mapping not found by preview'. However, the mapping itself occurs as expected in actual event policy.
   - **Summary: ${pattern1.varbind-1.3.6.1.4.1.546.17.1.1.3.5} ${pattern1.varbind-1.3.6.1.4.1.546.17.1.1.4.5} threshold breach**
     Maps the Summary property to the following statement: '*objectinstance objectattribute* threshold breach'.
     Example: //./OUTLOOK Memory(KB) threshold breach
   - **Message: ${pattern1.varbind-1.3.6.1.4.1.546.17.1.1.3.5} ${pattern1.varbind-1.3.6.1.4.1.546.17.1.1.4.5} ${pattern1.varbind-1.3.6.1.4.1.546.17.1.1.17.5} on fx:fqdn(${pattern1.snmp_agent})**
     Maps the Message property to the following statement: '*objectinstance objectattribute currentvalue* on *agentserver'.*
     Example: //./OUTLOOK Memory(KB) 150380 on server1.ca.com
   - **Repeat Count: ${pattern1.varbind-1.3.6.1.4.1.546.17.1.1.9.5}**
     Maps the RepeatCount property to the number of traps that have been generated on this object.
   - **User Attribute 1: Threshold: ${pattern1.varbind-1.3.6.1.4.1.546.17.1.1.4.5}**
     Maps the User Attribute 1 property to the object attribute value.
     Example: Memory(KB)

       NOTE
       Instead of prefixing the value with 'Threshold', you can rename the User Attribute 1 value to Threshold.
   - **User Attribute 2: Use Map function**
     Maps the User Attribute 2 property to the monitor threshold. Map the 1.3.6.1.4.1.546.17.1.1.18.5 varbind from integers to operators as follows:

- 1: (No operator)
- 2: >
- 3: <
- 4: >=
- 5: <=
- 6: =
- 7: !=

– **User Attribute 3: ${pattern1.varbind-1.3.6.1.4.1.546.17.1.1.19.5}**
  Maps the User Attribute 3 property to the monitor threshold value.
  Example: 50000
– **User Attribute 4: ${pattern1.varbind-1.3.6.1.4.1.546.17.1.1.2.5}**
  Maps the User Attribute 4 property to the object class.
  Example: Process
– **User Attribute 5: SystemEDGE trap**
  Assigns 'SystemEDGE trap' as the value for User Attribute 5.

Use the Service right-click menu to assign the AlertedMdr properties to a managed service so that the normalized event appears on that service CI.
The Select Data Sources page opens.

7. Perform the following actions and click Next:
   a. Select Save and Deploy policy.
   b. Select Generic SNMP Traps in the Data Source Type drop-down list.
   c. Move the Generic SNMP Traps entry to the Selected Data Sources pane.

   The Confirm page opens.
8. Verify the policy information and click Finish.
   The policy deploys. Any time a CA SystemEDGE monitor entry generates an aggregate state threshold breach trap, Event Management normalizes it according to the deployed policy.
9. Return to the Event Policies dialog and run the following event search:
   ```
   userAttribute2='SystemEDGE trap'
   ```
   This search pattern returns all normalized CA SystemEDGE traps.
10. Click Create Policy, name the policy RefineNormalizedTraps on the New Policy page, select Create New Event, and click Next.
11. Make the following changes in the New Event table:
    – **User Attribute 1: Threshold: ${pattern1.userAttribute1} ${pattern1.userAttribute2} ${pattern1.userAttribute3}**
      Maps the User Attribute 1-3 values in the normalized trap into a single value. The mapping provides a consolidated trap threshold statement, which includes the operator values that you mapped in the normalization action.
      Example: Memory(KB) > 50000
      Note: Instead of prefixing the value with 'Threshold', you can rename the User Attribute 1 value to Threshold.
12. Deploy the policy on the same Generic SNMP Traps connector.
    This policy takes the information in the normalized event and creates an event with a complete threshold statement in the User Attribute 1 value. The separate policy is required because normalization does not support embedded map functions in an Assigned Value cell. Therefore you cannot combine the threshold statement. The create event policy also filters out the original events to avoid duplicate alerts.
13. Create a separate policy with a filter action on the same search pattern that you entered in Step 9.
    This policy discards the original event so that only the created event with the correct threshold mapping appears in the Operations Console.
14. Select the Alert Queues tab and click Add.
    The New Alert Queue dialog opens.
15. Perform the following actions and click Next:

a. Enter SystemEDGE Monitors in the Queue Name field.
b. Select User Attribute (4) in the Attribute drop-down list.
c. Select Equal To in the Comparison Type drop-down list.
d. Enter Process in the Attribute Value field.
e. Click Add.

The queue criteria adds alerts with a User Attribute 4 property value of Process.

16. Complete the alert queue creation process and click Finish on the Confirm page.

    NOTE
    You can assign escalation policies and user group access to the queue.

    The alert queue is created. The alerts for CA SystemEDGE traps with a User Attribute 4 value of Process appear in this queue. You can manage process monitoring traps together.
17. Repeat Steps 13-15 to create queues that are based on other trap properties. For example, because you isolated the key identifier properties in the User Attribute properties, you can create queues to group traps of the same object class or attribute.

How to Schedule Maintenance for Services and Resources

Contents

As an administrator, you schedule service and CI maintenance to help ensure peak performance.

A *maintenance schedule* defines a time in the future (which can recur at defined intervals) on which to put a service or CI in maintenance.

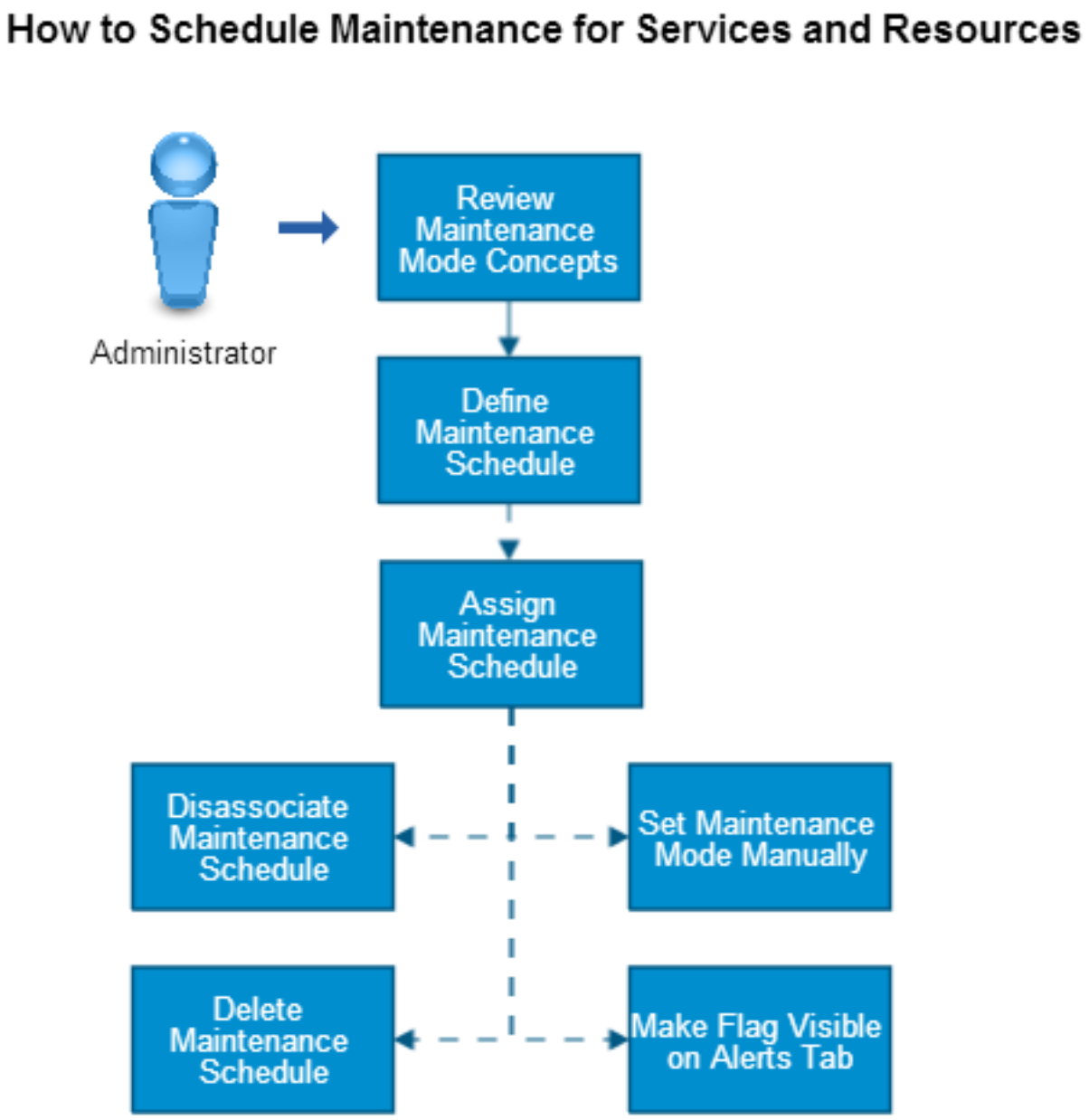
Because CIs can be in multiple services, they can also have multiple schedules.

Before you schedule maintenance, determine the hours when services and resources could be taken offline without affecting the business. If a server is part of a server farm, for example, it could be maintained during business hours without adversely affecting a service because of the redundancy with the other servers.

NOTE
CIs in maintenance mode are excluded from custom policy calculations that are related to average or percentage.

Use this scenario to guide you through the process:

Figure 40: how to schedule maintenance



1. Review maintenance schedule concepts:
   - Maintenance schedule features and behaviors
   - Maintenance mode and impact analysis
   - Maintenance mode and alert escalation
   - Domain manager maintenance propagation
2. Define the maintenance schedule.
3. Assign the maintenance schedule to services and CIs.
4. (Optional) Perform any of the following actions:

- Disassociate the maintenance schedule.
- Delete a maintenance schedule.
- Set the maintenance mode manually.
- Make the maintenance mode flag visible on the Alerts tab.

Maintenance Schedule Concepts

Become familiar with the following maintenance schedule concepts.

Maintenance Schedule Features and Behaviors

The following features and behaviors are related to maintenance for services and CIs:

- **Operational mode**
  Specifies whether services and CIs are in production or in maintenance. The following interfaces shows the operational mode:
  - Operations Console, Component Detail pane, Information tab
    If you must maintain an object off the regular schedule, you can set the mode on this tab manually. Otherwise, the schedule sets the mode automatically.
  - Dashboard, Services table
- **Maintenance schedule**
  Specifies when to perform regular maintenance on a service or CI. The schedules are shown on the Information tab of the Component Detail pane.
- **Icon in Topology view**
  Shows when an object is in maintenance mode because its icon is dimmed in the Topology view. Also, a small decal (similar to a picture of a wrench) appears at the upper left of the icon. No change happens to the icon in the Navigation pane.
- **Maintenance flag**
  Shows which CIs are in maintenance mode. By default, the flag is not visible on the Alerts tab of the Contents pane. You can make it visible by setting preferences. Flags are applied to existing alerts and any new alerts that are generated.
  When a service is in maintenance, the services and CIs under it are not.
- **Impact and escalation**
  Configures the effect that maintenance has on alert impact and escalation. You can control the following behaviors:
  - Whether alerts for CIs in maintenance mode affect the impact of a parent object. By default, the impact of CIs is propagated to the parent service based on propagation and propagation policy.
    For more information about how maintenance mode affects impact, see Maintenance Mode and Impact Propagation.
  - Whether alerts are escalated when CIs are in maintenance. By default, alerts are escalated when the CIs return to production mode.
    For more information about how maintenance mode affects escalation, see Maintenance Mode and Alert Escalation.
- **Alert creation**
  Specifies whether alerts are shown for items in maintenance. By default, alerts are shown after the items return to production mode. You can control this setting using a check box on the Alert Filter dialog.
  When the filter is enabled, the Alerts tab in the Contents pane displays "Filtered By Maintenance."

Maintenance Mode and Impact Propagation

A global setting on the Administration tab lets you specify how maintenance mode affects the CI and service impact. You can select whether to propagate the impact to a parent CI or service.

NOTE
CIs in maintenance mode are excluded from custom policy calculations that are related to average or percentage.

CA SOI does not automatically place CIs for a service in maintenance mode because the CIs may belong to another service that is not in maintenance mode. Consequentially, Operators could receive alerts for a service that is in maintenance mode. The Operators then mistakenly open tickets against the alerts. You can configure CA SOI to hide alerts on CIs in which that parent service is in maintenance mode. For more information, see Hide Alerts in Maintenance Mode.

- **Impact propagation on**
  Specifies that a parent service receives impact values from child services or CIs no matter whether it is in maintenance or production mode.
- **Impact propagation off**
  (Default) Specifies that a parent service in maintenance mode is not affected by the impact of child services or CIs. If a CI is propagating impact, either from a child CI or from alerts on the CI itself, impact propagation stops when the service enters maintenance.
  When the parent enters maintenance, any propagated impact the service received during production mode is set to zero and the status is returned to normal. The only exception is for impact from an alert on the service itself, as opposed to from child objects.

For information about how to configure impact propagation for maintenance mode, see Configure Global Settings.

Maintenance Mode and Alert Escalation

Alert escalation behavior of items in maintenance mode is specific to each escalation policy.

When you define escalation policy, you can specify whether alerts are escalated for CIs that are in maintenance mode. For CIs that are in multiple services, escalation occurs if only one parent is in maintenance.

If the parent service is in maintenance, you can also define whether to escalate any type of infrastructure alert. If you disable this escalation, the policy is only applied after the service is out of maintenance mode. If you enable this escalation, policy is assigned to multiple services, and one of the services is not in maintenance mode, the policy is still triggered.

For more information about how to configure alert escalation for maintenance mode Hide Alerts in Maintenance Mode.

Domain Manager Maintenance Propagation

You can configure a global setting to propagate the maintenance mode setting from domain managers that are integrated through connectors for managed CIs and services in CA SOI.

If you enable this feature, a change to the maintenance mode in any domain manager that is integrated through a connector causes the CA SOI maintenance mode for that resource to change accordingly.

Multiple domain managers can have different maintenance settings for the same CI. Reconciliation formulas in the CA Catalyst Logic Server determine which maintenance setting is propagated to CA SOI. For more information about reconciliation, see How to Perform CA Catalyst Reconciliation.

NOTE
By default, the last update to the maintenance mode is always propagated to CA SOI.

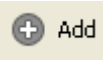
You can also configure maintenance settings in CA SOI to propagate to the source domain managers. For more information about configuring CI synchronization, see Synchronization.

Define a Maintenance Schedule

CA SOI lets you schedule maintenance to occur on specific days and times.

Follow these steps:

1. Open the Operations Console and select Tools, Schedule Editor.
2. Click

   .
3. Select or enter the following information:
   - **Schedule Type**
     Specifies the type of schedule to create. Select Maintenance Schedule from the drop-down list. The other two schedule types (SLA Period and Business Hours) are used for SLAs and escalation policy.
   - **Start Date**
     Specifies the day of the week and the date when the schedule begins. The date cannot be more than one year in the future.
   - **Time**
     Specifies the start time, end time, and duration (in hours) of the maintenance. The duration is calculated automatically based on the start and end times.
   - **Recurrence**
     Specifies how often to apply maintenance. Select from the following options:
     - **None**
       Specifies that the maintenance schedule occurs only once.
     - **Daily**
       Specifies that the schedule recurs every week day, every weekend, or after a specified number of days. For example, you can create a schedule that renews every three days. A daily schedule starts for the first time on the specified start date.
     - **Weekly**
       Specifies that the schedule recurs after a specified number of weeks and begins on a specific day. For example, you can create a schedule that starts on the first Monday after the specified start date and renews every two weeks.
     - **Monthly**
       Specifies that the schedule recurs after a specified number of months and begins on a specific day of the month. For example, you can create a schedule that starts on the fifth day of the month after the specified start date and renews every three months.
     - **Yearly**
       Specifies that the schedule recurs after a specified number of years and begins on a specific day of a specific month. For example, you can create a schedule that starts on January 5 (the first occurrence of this day after the start date) and renews every two years.

     For all recurrence options, select either No Expiration for the schedule to recur indefinitely or enter a Schedule End Date when the schedule stops.
   - **Description**
     Specifies a name for the schedule. Spaces and punctuation are allowed.
     Limits: 256 characters
4. Click OK.
   The Schedule Editor reopens, with the new schedule listed on the Schedules tab.

Assign a Maintenance Schedule to Services and CIs

After you define a maintenance schedule, you assign it to services and CIs. Assigned schedules are listed on the Information tab of the Component Detail pane. You can assign maintenance schedules to a single CI, multiple selected CIs at once, and a CI and all of its children.

When maintenance for an object occurs based on the schedule, the following changes take place:

- The Operational Mode is changed automatically to Maintenance. The modes are shown on the Information tab in the Component Detail pane. When maintenance is finished, the mode is changed back to Production.
- The icon is dimmed in the Topology view. Also, a small decal (similar to a picture of a wrench) appears at the upper left of the icon. No change happens to the icon in the Navigation pane. An 'M' appears in the Information column next to the CI.

Follow these steps:

1. Select a single or multiple CIs and perform one of the following actions:
   – Right-click the select CIs and select Assign Maintenance Schedules, or Advanced Maintenance, Assign Maintenance Schedules with Children.
   – Click a service or CI on the Services tab, and select Tools, Assign Maintenance Schedules or Tools, Advanced Maintenance, Assign Maintenance Schedules with Children.
2. Click a schedule in the Available Schedules pane on the left side of the dialog, and then click the right arrow button.

   NOTE
   If you do not see a schedule that you want to use, click the Create button to define a maintenance schedule.
3. Click OK.

Disassociate a Maintenance Schedule

If you want to remove a service or CI schedule, you can disassociate the schedule. The assigned schedules are listed on the Information tab of the Component Detail pane.

NOTE
Before you delete an unwanted schedule, disassociate all the associated items from the schedule.

Follow these steps:

1. Select a single or multiple CIs and perform one of the following actions:
   – Right-click the select CIs and select Assign Maintenance Schedules, or Advanced Maintenance, Assign Maintenance Schedules with Children.
   – Click a service or CI on the Services tab, and select Tools, Assign Maintenance Schedules or Tools, Advanced Maintenance, Assign Maintenance Schedules with Children.
2. Click a schedule in the Current Schedules pane and click the left arrow button.
3. Click OK.

Delete an Unused Schedule

You can delete a schedule (that is, a maintenance schedule, SLA period, or business hours) that no service or CI uses.

Follow these steps:

1. Open the Operations Console and select Tools, Schedule Editor.
2. Select the schedule to delete in the Schedules tab.
3. (Optional) Disassociate the schedule from any items in the Associated Items tab them before deleting.
4. Click Delete then confirm the deletion.

Set Maintenance Mode Manually

When scheduled maintenance occurs, CA SOI automatically changes the operational mode of services and resources to Maintenance. When the scheduled maintenance is finished, the mode automatically changes back to Production.

If you must perform unscheduled maintenance, you can also change the maintenance mode manually. You can change the maintenance mode for a single CI, multiple selected CIs, or a CI and its children.

Follow these steps:

1. Right-click one or many services or CIs in the Navigation or Contents pane and select Set Maintenance Mode or Advanced Maintenance, Set to Maintenance with Children.

   NOTE

   Set to Maintenance with Children is only available in the Navigation pane.
2. Perform one of the following actions:
   – Click Set in the New Mode column for the appropriate resource, select the maintenance mode to apply, and click OK.
     The maintenance mode changes for the selected CI or service.
   – Click Maintenance or Production next to "Reset 'New Mode' column value for all selected items" and click OK.
     The maintenance mode changes to the appropriate setting for all selected services or CIs.

You can also manually set the operational mode from the Information tab for a CI or service in the Component Detail pane. Click set next to Operational Mode to change the mode.

Make the Maintenance Flag Visible on the Alerts Tab

An alert maintenance flag indicates that CIs are in maintenance mode. The flag appears for existing alerts and any new alerts that are generated. By default, the flag is not visible on the Alerts tab in the Contents pane. You can make it visible, however, by setting preferences.

Follow these steps:

1. Open the Operations Console and select View, Preferences.
2. Expand Alerts Tab and Alerts Table and click Columns.
3. Click Maintenance, and click OK.
4. Restart the Operations Console.

Customization and Maintenance

This page describes how administrators can customize product functions and maintain installed components when updates such as password changes are required.

Product Customization

This section describes how you can customize the product UIs.

PC Dashboard Customization

Contents

As an administrator, you can customize various areas of the Dashboard to display pertinent data and optimize the appearance of the interface.

The Dashboard is typically accessible to all departments of an organization, not only administrators and operators. Depending on the structure of your organization, you can allow high-level business executives or even customers to access the Dashboard.

Change the Icon Shown on the Dashboard

You can replace the icon that is shown on the CA SOI Dashboard with a custom graphic. You replace the icons for selected user groups by setting a preference in the Operations Console. If necessary, you can assign different logos for each user group to support multi-tenancy or different groups within one organization.

Follow these steps:

1. Open the Operations Console, and click the Users tab in the left pane.
2. Click a user group and select View, Preferences.
3. Expand the following preferences from the list in the left pane: Web, Logo Icon File Name.
4. Copy your icon file to the following directory:
   <SOI_HOME>\SamUI\webapps\sam\ui\images
5. Enter the file name in the Logo Icon File Name field and click OK. **Note:** The filename name is case sensitive.
   The change takes effect the next time a user in the group logs in to the Dashboard.

Customize the Services Table

The Dashboard tab contains the Services table. The table displays information about the services CA SOI is managing. You can customize this table to display only the columns that contain the information important to you.

> **NOTE**
> CA SOI applies updated preferences when the Dashboard refreshes, which is every 30 seconds by default.

Follow these steps:

1. Click the Dashboard tab.
   The Services table displays up to five rows of services and up to eight columns (depending whether it is the default view or has already been customized). The Services column is always included in the table.
2. Click Preference.
   The Columns tab of the User Preference dialog opens with the currently included columns listed in the Show Columns field, and the unused columns listed in the Hide Columns field.
3. Click one or more columns you want to move to the opposite pane (use Shift+click or Ctrl+click for multiple selections).
4. Click the arrows to move the selected columns to the opposite pane.

   NOTE

   - The columns listed in the Hide Columns field are removed from the Services table. The corresponding tab is *not* removed from the Details of Selected Service panel when you save your changes.
   - When you move a column to the Show Columns field, it appears at the bottom of the list. The table displays the columns in the order that is determined by this list. The first column (top of the list) is the first (leftmost) column in the table. The last column in the list is the rightmost column in the table. You can select a column in the Show Columns list, then click the up or down arrows to change the column order.
5. For the Current SLA, Health, Quality, and Risk tabs, move any of the SLA states to the Hide (if value equals) column to prevent services in that state from appearing in the Services table.
6. Click the Other tab, and perform one of the following actions:
   a. Clear the check box next to any priority state to prevent services in that state from appearing in the Services table.
   b. Clear the check box next to any operation mode to prevent services in that mode from appearing in the Services table.
   c. Set the availability minimum and maximum (expressed as a percentage) to prevent services that do not fall within the specified range from appearing in the Services table.
7. Click the Tabs tab and click the arrows to show or hide specific details tabs.
8. Click Save.

   NOTE
   Your changes are saved immediately, but are not reflected until the Dashboard refreshes, which is approximately every minute by default.

Add Custom Tabs to the Dashboard

You can configure CA SOI to display up to ten custom tabs on the Dashboard. Each custom tab displays a website that can be important for you to monitor. The following websites are examples:

- A website that an associated service produces and hosts.
- A website that provides you with information related to service management. For example, an Intranet site that informs you when the associated service is updated with new components.

The custom tabs appear at the top of the Dashboard to the right of the Administration tab.

Follow these steps:

1. Click the Dashboard tab and click Preference.
2. Click the Custom Links tab, enter the appropriate information in the following fields, and click Save:
   - **Show**
     Specifies whether the corresponding custom tab is shown or hidden. You can clear the check box to hide the custom tab after it is created.
   - **Tab Title**
     Specifies the text that appears on the custom tab.
   - **Web Address**
     Specifies the URL of the website that is displayed on the custom tab. The transfer protocol (for example, http or https) is required. You can optionally enter {servicename} to include the service as part of the query to a third-party tool. For example, to pass the service name to www.anyurl.com, enter the following URL query:
     ```
     www.anyurl.com/?query={servicename}
     ```

   The custom tab appears next to the Administration tab and displays the corresponding website when clicked.
3. (Optional) Repeat Step 3 to add additional custom tabs.

Note: Only a regular CA EEM user can save new custom tabs. The 'samuser' can create new tabs but they are not saved in the preference setting.

Add Custom Links to the Dashboard

A custom link can launch any URL that provides more information about a service, such as an internet search on the service name or a URL for the source management application. You can add the custom launch-in-context links for all services to the Action drop-down menu on the Dashboard.

Follow these steps:

1. Open the <SOI_HOME>\SamUI\webapps\sam\WEB-INF\console\config\custom-menu-config.xml file in a text editor on the UI Server.
2. Create and uncomment a full menu item using the conventions described in the custom-menu-config.xml file, and follow the instructions in the file, paying close attention to the following attributes:
   - **item name**
     Specifies the name to display in the Actions drop-down menu.
   - **URL**
     Specifies the launch URL. You can use the value {0} as a substitution value for the service name in the URL.

   The Dashboard requires only these two attributes to enable the link, but they must exist in the context of a full menu item. The other attributes can be empty, but the link appears in the Launch menu of the Operations Console.
   For information about creating custom Operations Console menus and links, see Operations Console Menu Customization.
3. Save and close the file.
4. Restart the CA SAM User Interface service.
   The custom link appears in the Dashboard for all services when you click the Actions menu.

Add Custom Metrics to the Dashboard

You can add custom metrics to the dashboard that appear as table columns on the Services pane or as line charts on the Details pane. CA SOI can retrieve custom metrics from Microsoft SQL Server and Sybase databases to add important service-related data that CA SOI does not monitor by default to the Dashboard.

> **NOTE**
> To retrieve metrics from another database type, use your own driver that works with those databases. CA Technologies supports only Microsoft SQL Server and Sybase database metrics.

Follow these steps: for custom table columns

1. Open the <SOI_HOME>\SamUI\webapps\sam\thinuiconf\custom_metric_definition.xml file in a text editor on the UI Server.
   This file contains detailed instructions and examples for adding custom dashboard metrics.
2. Create a custom metric table using the provided example, or find the METRIC_TABLE tag and uncomment the example entry in the section labeled '<!-- Put custom metric here'. For either method, populate the following attributes in the entry:
   - **MAPTODASHBOARDTABLENAME**
     Retain the default MyServicesStatusTable value. This attribute must have this value for the metric to display.
   - **NAME**
     Specifies the custom metric name. This name follows the Java class name convention, such as no spaces or special characters.
   - **COLUMNLABEL**
     Specifies the column labels to display in the Services pane. The number of values in this attribute equal the number of values in the DATANAME attribute minus one.
   - **COLUMNALIGN**
     Specifies the column alignment of each label. The alignments are not used. The number of values in this attribute equal the number of values in the COLUMNLABEL attribute.
   - **COLUMNDATATYPE**
     Specifies the data type for each column. The number of values in this attribute equal the number of values in the COLUMNLABEL attribute. Use 'string' as the data type for normal text display, or select from one of the following icon styles: image_sla, image_health, image_quality, image_risk. See the dashboard for examples of these icon styles.
   - **COLUMNSORTABLE**
     Specifies whether each column is sortable. Enter 'true' or 'false'.
   - **CONNECTION_URL**
     Specifies the connection URL for the database that contains the metric information. The only supported driver is JTDS, which supports most major database vendors.
   - **USERNAME**
     Specifies the user name for connecting to the database.
   - **PASSWORD**
     Specifies the password for connecting to the database. Use the <SOI_HOME>\tomcat\bin\WSSamEncryptCmd.bat utility to generate an encrypted password for this file.
   - **QUERY**
     Specifies the query for obtaining the metric information from the database. Note the following items:
     - For best results, use a simple query. If a complex query is required, create a database view to mask the complex query and expose only the relevant metric data.
     - Ensure that the service name is the first field in the query, because it is used as the key to map to data in the dashboard table.
     - CA SOI does not verify the external connection, so ensure that the database is always available to avoid errors.
   - **DATANAME**

Specifies the column names in the SELECT query. These values map directly to the SELECT statement. If the column names in the query are mixed-case, use exact characters.
  - **DATATYPE**
    Specifies the data type of the columns defined in the query.
  - **REFRESH_RATE**
    Specifies the refresh rate to retrieve custom metric data in minutes. The refresh rate must be greater than 60 seconds. A high frequency rate affects UI Server performance.
3. Save and close the file.
4. Restart the CA SAM User Interface service.
   The metric appears as a table column in the Services pane. If you defined multiple metrics, you cannot control the relative order of the metrics.

Follow these steps: for custom line charts

1. Open the <SOI_HOME>\SamUI\webapps\sam\thinuiconf\custom_metric_definition.xml file in a text editor on the UI Server.
   This file contains detailed instructions and examples for adding custom dashboard metrics.
2. Create a custom metric chart using the provided example, or find the METRIC_CHART tag and uncomment the example entry in the section labeled '<!-- Put custom metric here'. For either method, populate the following attributes in the entry:
  - **NAME**
    Specifies the name of the custom metric chart. The name must be unique, so prefix the chart name with 'CUSTOMMETRIC_'.
  - **TITLE**
    Specifies the chart title that displays on the tab in the dashboard Details pane.
  - **XLABEL**
    Specifies the label of the X-axis data on the chart.
  - **XDATATYPE**
    Specifies the data type of the X-axis data on the chart. The current supported types are 'string' and specific time unit data types (hour, date, month, and year). For more information about these types, see the text in the configuration file.
  - **XDATATYPELOCALE**
    Specifies the locale value of the datetime data that is retrieved from the database. This parameter only applies when the XDATATYPE value is a time-unit data type. The value should conform to the appropriate standard local type, such as EN_US. If this value is empty, the chart uses the locale of the UI Server system as the default.
  - **YLABEL**
    Specifies the label of the Y-axis data on the chart. You can have up to six Y-axis labels.
  - **YDATATYPE**
    Specifies the data type of the Y-axis data on the chart. Enter the same number of data types in the YLABEL attribute.
  - **YDISPLAYUNIT**
    Specifies the display unit in the Y-axis chart.
  - **CHART_REFRESH_RATE**
    Specifies how often the chart sends requests to update data in minutes.
  - **CONNECTION_URL**
    Specifies the connection URL for the database that contains the metric information. The only supported driver is JTDS, which supports most major database vendors.
  - **USERNAME**
    Specifies the user name for connecting to the database.
  - **PASSWORD**

Specifies the password for connecting to the database. Use the <SOI_HOME>\tomcat\bin\WSSamEncryptCmd.bat utility to generate an encrypted password for this file.

– **QUERY**
  Specifies the query for obtaining the metric information from the database. Note the following items:
  - For best results, use a simple query. If a complex query is required, create a database view to mask the complex query and expose only the relevant metric data.
  - Ensure that the number of values in XLABEL plus the number of values in YLABEL equals the number of columns in the query.
  - CA SOI does not check the external connection, so ensure that the database is always available to avoid errors.

– **DATANAME**
  Specifies the column names in the SELECT query. These values should map directly to the SELECT statement. If the column names in the query are mixed-case, use exact characters.

– **DATATYPE**
  Specifies the data type of the columns defined in the query.

– **REFRESH_RATE**
  Specifies the refresh rate to retrieve custom metric data in minutes. The refresh rate must be greater than 60 seconds. A high frequency rate affects UI Server performance.

3. Save and close the file.
4. Restart the CA SAM User Interface service.
   The metric appears as a separate tab in the Details pane that displays a line chart.

Customize the Details Pane

The Dashboard tab contains the Details pane, which provides detailed information about the service selected in the Services table. You can customize the pane to display only the tabs that contain the information important to you.

Follow these steps:

1. Click the Dashboard tab and click Preference.
2. Click the Tabs tab.
3. Click one or more tab names (use Shift+click or Ctrl+click for multiple selections) to move to the opposite column.
   The selected tab names are highlighted.
4. Click the arrows to move the selected tabs to the opposite field.
5. Click Save.

> **NOTE**
> When you move a tab to the Show Tab field, it appears at the bottom of the list. However, unlike customizing the Services table, you cannot modify the order of the tabs.

Configure the Dashboard Refresh Rate

Users with administrator rights can configure the dashboard refresh rate.

Follow these steps:

1. Navigate to the <SOI_HOME>\SamUI\webapps\sam\ui\ directory, and open the refresh.properties file in a text editor.
2. Locate the following line, and change the refresh rate:
   ```
   dashboard.refresh=30000
   ```
   The default refresh rate is 30000 milliseconds (30 seconds). Increasing the refresh rate may improve the user experience when using the dashboard.
3. Save and close the file.
   This new refresh rate is used the next time that a new browser is opened.

Configure the Level of Services the Dashboard Displays

Users with administrator rights can configure the number of service levels that the Dashboard displays. This is a system-wide configuration setting where all users see the same number of service levels.

Follow these steps:

1. Navigate to the <SOI_HOME>\SamUI\conf\thinuiconf directory and open the tables_definition.xml file in a text editor.
2. Locate the following line, and change the SERVICE_LEVEL=ALL to SERVICE_LEVEL=*numeric_value*".

   ```
   <TABLE NAME="MyServicesStatusTable" TITLE="ServicesStatus" CONSOLELINKTEMPLATE="/sam/oneclick.jnlp?
   explorer=" SSOREDIRECT="/sam/sso/redirect?reqURL=" SERVICE_LEVEL="ALL">
   ```

   For example, SERVICE_LEVEL=1 displays only top-level services, SERVICE_LEVEL=2 displays the first two levels of a service, and so on.
3. Save and close the file, and then restart the CA SOI User Interface service.
   The Dashboard displays the level of services that you configured.

Display CA SOI Dashboard in SharePoint

As an administrator or an operator, you can configure Microsoft SharePoint to display the CA SOI Dashboard. This procedure assumes SharePoint is installed and you have working knowledge of adding content to SharePoint. For more information about installation and working with SharePoint, see the SharePoint documentation.

Obtain the CA SOI URL to complete this procedure.

Follow these steps:

1. Open your browser to the SharePoint server.
2. Log in to SharePoint as an administrator.
3. Select the tab (site) where you want to add the Dashboard.
4. Click Site Actions near the top right of the page, then click Edit Page.
5. Click Add a Web Part.
6. Select the Page Viewer Web Part check box under All Web Parts, Miscellaneous.
7. Click Add.
   The Page Viewer Web Part appears at the top of the page.
8. Click Edit, then click Modify Shared Web Part.
   A dialog expands and provides fields for the new website; in this case, the CA SOI Dashboard.
9. Enter the URL link to CA SOI.
10. (Optional) If necessary, adjust the height and width to display the CA SOI Dashboard correctly.
11. Click OK, then click Exit Edit Mode.
    The CA SOI Dashboard displays in SharePoint.

Mobile Dashboard Customization

Contents

As administrator, you can customize the appearance of the Mobile Dashboard, including the metric icons, display, or deploy on a standalone server.

CA SOI provides a version of the Dashboard that is designed for mobile devices. On the Mobile Dashboard, you can view and perform a subset of actions available on the PC version of the Dashboard and the Operations Console.

> **NOTE**
> For more information about configuring the Mobile Dashboard, see Configure Mobile Dashboard Integration.

Change the Metric Icons on the Mobile Dashboard

You can change the metric icons that show the status of services and SLAs on the Mobile Dashboard. For example, you can change the icons if you prefer a vertical graph representation or an icon that is accompanied by text.

Follow these steps:

1. Stop the CA SOI User Interface service on the UI Server.
2. Delete the <SOI_HOME>\SamUI\webapps\mobile directory.
3. Rename the <SOI_HOME>\SamUI\webapps\mobile.war file to mobile.zip.
4. Unzip the mobile.zip file to any empty temporary directory.
   The compressed files are extracted.
5. Replace the icons in the \images\gauge directory with a new version.
   The new icons must have the same names as the old ones and must be PNG files. The following sets of files exist:
   – X-out-of-5.png files show the Quality, Risk, Availability, and Health values.
   – X-out-of-3.png files show the SLA status.
   You cannot add different icons for each metric (for example, separate icons for Health and Quality).
6. Add the modified contents of the temporary directory back to mobile.zip.

   WARNING
   Verify that you compress the contents of the temporary directory only and not the directory itself; otherwise, the Mobile Dashboard will not work.
7. Rename the file to mobile.war.
8. Replace the old <SOI_HOME>\SamUI\webapps\mobile.war file with the new one.
9. Start the CA SAM User Interface service.
   The new icons appear on the Mobile Dashboard.

Change the Icon on the Mobile Dashboard

You can replace the icon that is shown on the Mobile Dashboard with a custom graphic, which applies for all user groups.

To change the icon on the Mobile Dashboard, place a PNG file named mobile_logo.png into the <SOI_HOME>\SamUI \webapps directory.

The new file appears on the Mobile Dashboard in place of the CA Technologies logo.

NOTE
You can change the display and positioning of the icon if necessary by customizing the Mobile Dashboard display.

Customize Mobile Dashboard Display

If certain objects or pages do not appear correctly on new or unsupported devices, you can customize the Mobile Dashboard display.

Follow these steps:

1. Create a file named mobile_extra_style.css in the <SOI_HOME>\SamUI\webapps\mobile\styles directory on the UI Server.
2. Write the modifications to the default style sheet and save the file.
   The customizations display on the Mobile Dashboard. You can tweak the style sheet as many times as necessary to achieve the optimal display settings.

For example, if you change the Mobile Dashboard icon, you can modify the width and height as follows in mobile_extra_style.css:

```
.logo {
```

```
  width: 40px;
  height: 40px;
  left: 2px;
  top: 2px;
}
```

This example sets the icon to be a square of 40x40 pixels and moves it five pixels near the top of the page.

Deploy Mobile Dashboard on a Standalone Server

By default, the Mobile Dashboard installs with and runs on the UI Server. An enterprise with strict firewall and port requirements may need to deploy the Mobile Dashboard manually on a standalone server within a DMZ. In this dual-firewall deployment, the UI Server remains in the protected Intranet, while the Mobile Dashboard is accessible to clients from the Internet by opening ports 7070 and 7403 on the firewall that separates the DMZ from general user access.

To accomplish this configuration, manually move the Mobile Dashboard installation on the UI Server to the server in the DMZ.

> **NOTE**
> This procedure is only necessary in the special case described.

Follow these steps:

1. Open ports 7070, 7403, 7090, and 7493 on the firewall that separates the UI Server and SA Manager from the DMZ. These ports are the default access ports for the SA Manager and UI Server. To change the ports for accessing these components behind the firewall, see Step 8.
2. Stop the CA SAM User Interface service on the UI Server.
3. Create a root path for the CA SOI files on the Mobile Dashboard server as C:\Program Files\CA\SOI.
4. Copy the following files and directories from the UI Server to the Mobile Dashboard server in the DMZ.

   > **NOTE**
   > The listed directories are all relative to the SOI_HOME installation path on the UI Server, which you created on the Mobile Dashboard server in Step 3. Create the following intermediate folders on the Mobile Dashboard server, if necessary:

   - jsw\bin
   - jsw\lib
   - jsw\conf\soi-user-interface.conf
   - jsw\conf\soi-user-interface.properties
   - jsw\conf\wrapper-jvm-32.conf
   - jsw\conf\wrapper-jvm-64.conf
   - jre-32
   - jre-64
   - SamUI\bin
   - SamUI\conf
   - SamUI\custom
   - SamUI\lib
   - SamUI\registry
5. Create the following empty directories on the Mobile Dashboard server in the created SOI_HOME location:
   - jsw\logs
   - SamUI\logs
   - SamUI\webapps
   - SamUI\work
6. Copy the following files from the UI Server to the same location on the Mobile Dashboard server:

- SOI_HOME\SamUI\webapps\ssamobile.war
- SOI_HOME\SamUI\webapps\mobile.war
- SOI_HOME\SamUI\webapps\help.war

7. Delete the following files from the SOI_HOME\SamUI\conf\Catalina\localhost directory on the Mobile Dashboard server:
   - host-manager.xml
   - manager.xml
   - sam.xml
   - solr.xml
8. Do one of the following based on whether you want the Mobile Dashboard to connect to the CA REST Service using HTTP or HTTPS:
   - HTTPS: Set the following properties in the SOI_HOME\SamUI\conf\soi_conf\mobile.properties file on the Mobile Dashboard server, and save the file:
     - headlessURL: https://<UI Server>:<HTTPS Port>/rest

       NOTE

       The default UI Server HTTPS port is 7403.
     - applicationSecurityScheme: https
     - headlessKeyStoreLocation: set to the location of the self-signed certificate that the CA SOI REST service uses. The default keystore is SOI_HOME\SamUI\conf\ssa.jks, which you would specify in the properties file using forward slashes.
       Example: headlessKeystoreLocation=C:/Program Files (x86)/CA/SOI/SamUI/conf/ssa.jks
     - headlessKeystorePassword: keystore password, which by default is catalyst
   - HTTP: Set the following properties in the SOI_HOME\SamUI\conf\soi_conf\mobile.properties file on the Mobile Dashboard server, and save the file:
     - headlessURL: http://<UI Server>:<HTTP Port>/rest

       NOTE

       The default UI Server HTTP port is 7070.
     - applicationSecurityScheme: http
9. (HTTP only) Enable the REST Service to run on HTTP as follows if you want the Mobile Dashboard to connect to the CA REST Service using HTTP:

   NOTE

   If you are connecting using HTTPS, skip to Step 10. The REST Service runs on HTTPS by default.

   - Open the SOI_HOME\SamUI\webapps\rest\WEB-INF\web.xml file on the UI Server.
   - Find the <param-name>allowPlainCredentials</param-name> line, and change the associated <param-value> value to true.
10. Open a command prompt on the Mobile Dashboard server, navigate to SOI_HOME\jsw\bin, and run the following commands:

    ```
    SAM_Services.cmd install ui
    SAM_Services.cmd start ui
    ```

    The necessary CA SOI services are installed and started on the Mobile Dashboard server.
11. Start the CA SAM User Interface service on the UI Server.
    The Mobile Dashboard is deployed on a standalone server and configured to communicate with the UI Server and SA Manager behind a firewall.
12. Open ports 7070 and 7403 (or the defined nondefault ports for accessing the UI Server) on the firewall that separates the DMZ with the Mobile Dashboard server from general internet access.

NOTE
Depending on your network configuration, you may also need to configure Network Address Translation (NAT) to forward requests from the Internet to the server inside a DMZ.

External clients can now access the Mobile Dashboard from the Internet using the following URL:

```
http://<MobileDashboardServer>:<port>/mobile
```

NOTE
If you configured NAT to forward requests to the Mobile Dashboard server, instead use the server name and port configured in the NAT configuration in the URL.

Operations Console Customization

Contents

As an administrator or an operator , you can customize the Operations Console in the following ways:

- Specify which columns appear, resize columns, and sort column data
- Dock and undock panes
- Clone (copy) panes
- Set display preferences, such as such things as which columns to display on the Alerts tab and the default filter to use for viewing all alerts.
- Export or import display preferences
- Navigate, collapse, or expand the Topology view.

Set Preferences

As an administrator or an operator, you set preferences that affect how the Operations Console displays information. You can specify such things as which columns to display on the Alerts tab, the default filter to use for viewing all alerts, and whether to add subcomponents to a service you are creating. Administrators can set preferences for either the logged in user only or for all users in a specified user group. An administrator can also lock a user group from changing any preference.

You can set the following preferences. The set preferences dialog provides more details about the preferences available for each tab.

- **Alerts Tab**
  Lets you set a global filter for all displayed alerts; specify whether a popup opens and a beep sound occurs for new alerts; control column order, the columns displayed, sort order of data in columns, and font; indicate whether a confirmation dialog opens when alerts are cleared (closed); and indicate whether the available ticket actions dialog displays when submitting a ticket.
- **Auditor**
  Lets you set the maximum number of audit entries the Auditor retrieves.
- **General**
  Lets you specify the default font for Information panes and tables; indicate the region used to format dates, times, and numbers; select an overall look other than the system default; specify the amount of scrollbar adjustment after a click of a scrollbar arrow; indicate whether the time format is 12-hour or 24-hour; select Coordinated Universal Time (UCT) instead of the default local system time zone; and specify the number of seconds that the cursor hovers over a button, field, or other component before a tooltip appears.
- **List Tab**
  Lets you specify the columns displayed, sort order of data in columns, and font.
- **Locator**

  Lets you set the maximum number of results returned and if only monitored objects are searched in the Locator dialog.

For more information, see Search for Objects Using the Locator.

- **Modeler**
  Lets you set values for the Service Modeler window, which is where you create and edit services. You can specify the confirmation and other dialogs to display, the default display and layout style, the default values for new items added, whether to retain the previous settings when performing various actions, whether automatic policy maintenance is active, and whether to add sub-components when adding a parent object to a service.
- **Service Discovery**
  Lets you set display options for Service Discovery confirmation dialogs including warning dialogs.
- **Services Tab**
  Lets you specify a maximum number of elements to display and whether a warning opens if the limit is exceeded; control the columns displayed, sort order of data in columns, and font; control whether drag-and-drop of items in the Services tab is allowed and if a confirmation dialog displays; and specify how items are displayed when the Operations Console opens.
- **Topology**

  Lets you set values for the Topology tab in the Contents pane of the Operations Console. Some preferences are the same as for the Service Modeler because they both have a Topology view. You can specify the confirmation dialogs to display, the layout for imported services, and whether to retain the previous interface settings.

  For more information, see Navigate the Topology View.

  - **Web UI**
    Lets you change the logo at the upper left corner of the browser interface, which has the Dashboard and Administration tabs. Changing the logo is useful for customers who want to display their own logo.

Follow these steps:

1. Access the Operations Console.
2. Do one of the following:
   – Select View, Preferences to set preferences for the logged in user.
   – Select the User tab, right-click a user group, and select Set Preferences to set preferences for all users in the selected user group.
3. (Optional) Click the type of preference you want to configure from the list in the left pane.

   NOTE
   An alternative method is to click a plus (+) button to display a list of available preferences in the left pane.
4. Set the preferences you want to change, and click OK.
   Most preferences take effect immediately. The following preferences, however, require a restart of the Operations Console:
   – Alerts Tab, Alerts Table
   – General, Locale
   – General, Look and Feel
   – General, Time Format
   – General, Time Zone
   – Services Tab, Initial View
5. (Optional) Select the Make Changes Permanent check box to keep your changed preferences the next time you log in.
6. Restart the Operations Console if the change did not take effect.
   The preference change takes effect for the logged in user or the selected user group.

Export or Import Preferences

Preferences affect how the Operations Console displays information. You can export preferences to a file so that another user or user group can copy them.

Follow these steps:

1. Open the Operations Console and select View, Preferences.
2. Click the top level to select all preferences, or click a subfolder containing the type of preferences you want to import or export.
3. Click the Export or Import button.
4. (Optional) Click one or more check boxes to remove the checkmark.
5. Click OK.
   The Select Users/Groups dialog opens.
6. Click a user or group, and click OK.

Customize Columns

You can change the way the Operations Console panes display table columns.

To specify the columns to display

1. Right-click a column heading.
2. Select the columns to display and click OK.

To specify column order

1. Select View, Preferences.
2. Expand Alerts Tab and Alerts Table, and click Column Order.
   Buttons for the available columns appear in a horizontal line.
3. Drag the buttons to the position you want, and click OK.

To resize columns

- Mouseover between columns so that the double-arrow icon opens. Click and drag left or right.
- Double-click the right side of a column header boundary to fit the column to the longest text it contains.

To sort columns

You can click a column heading to sort by that one heading or follow these steps to sort by multiple headings:

1. Right-click a column heading.
2. Click the Sort tab.
3. Select up to three columns by which to sort the table contents and whether to sort Ascending or Descending for each property, and click OK.

Dock and Undock Panes

By default, all panes open in the Operations Console; however, you can modify the view when necessary. Docked panes are visible on the main Operations Console page, and you can undock panes to separate them from the main page.

Each pane contains one of the following buttons:

-

Undocks the pane from the Operations Console. The pane opens in its own window and is removed from the main Console view. The button changes to the Dock button, which is a mirror image of the Undock button. Undocking panes can help you to make better use of your screen space.

-

  Docks an undocked pane with the Operations Console. To display closed panes, click the View menu and select the pane to display. You can also use the View menu to dock undocked panes.

Clone Panes

Cloning opens the Contents or Component Detail pane in a separate window that contains another instance of the pane. Cloning is useful for viewing more than one area of the Operations Console simultaneously. If you navigate away from the original source, the cloned window information display is not affected.

To clone panes, click the Clone

icon in the upper right corner of the Contents or Component Detail pane.

> **NOTE**
> If you click the Clone button in the Contents pane while the Component Detail pane is visible, a new window opens that contains instances of both panes.

Customize Operations Console Look and Feel

User preferences let you influence the look and feel of the Operations Console, such as the general appearance, font size, time zone, and locale. These preferences can apply for the logged in user only or for all users in a specified user group.

Follow these steps:

1. Access the Operations Console.
2. Do one of the following:
   - Select View, Preferences to customize the Operations Console look and feel for the logged in user.
   - Select the Users tab, right-click a user group, and select Set Preferences to customize the Operations Console look and feel for all users in the selected user group.

   The Set Preferences dialog opens.
3. Select the General folder.
   The General preferences appear in the right pane.
4. Set preferences for any of the following items and click OK:

   > **NOTE**
   > See the text under each preference for detailed descriptions.

   - Time Zone
   - Scrollbar Increment
   - Default Table Font
   - Locale
   - Time Format
   - Default Field Font
   - Look and Feel
   - Tool Tip Delay

The changes are saved for the logged user or the selected user group. If you select the Make Changes Permanent check box, the changes remain for all Operations Console sessions that the logged user or user group initiates.

5. (Optional) Restart the Operations Console to apply the following preferences:
   - Locale
   - Look and Feel
   - Time Format
   - Time Zone

   The changes appear in the Operations Console.

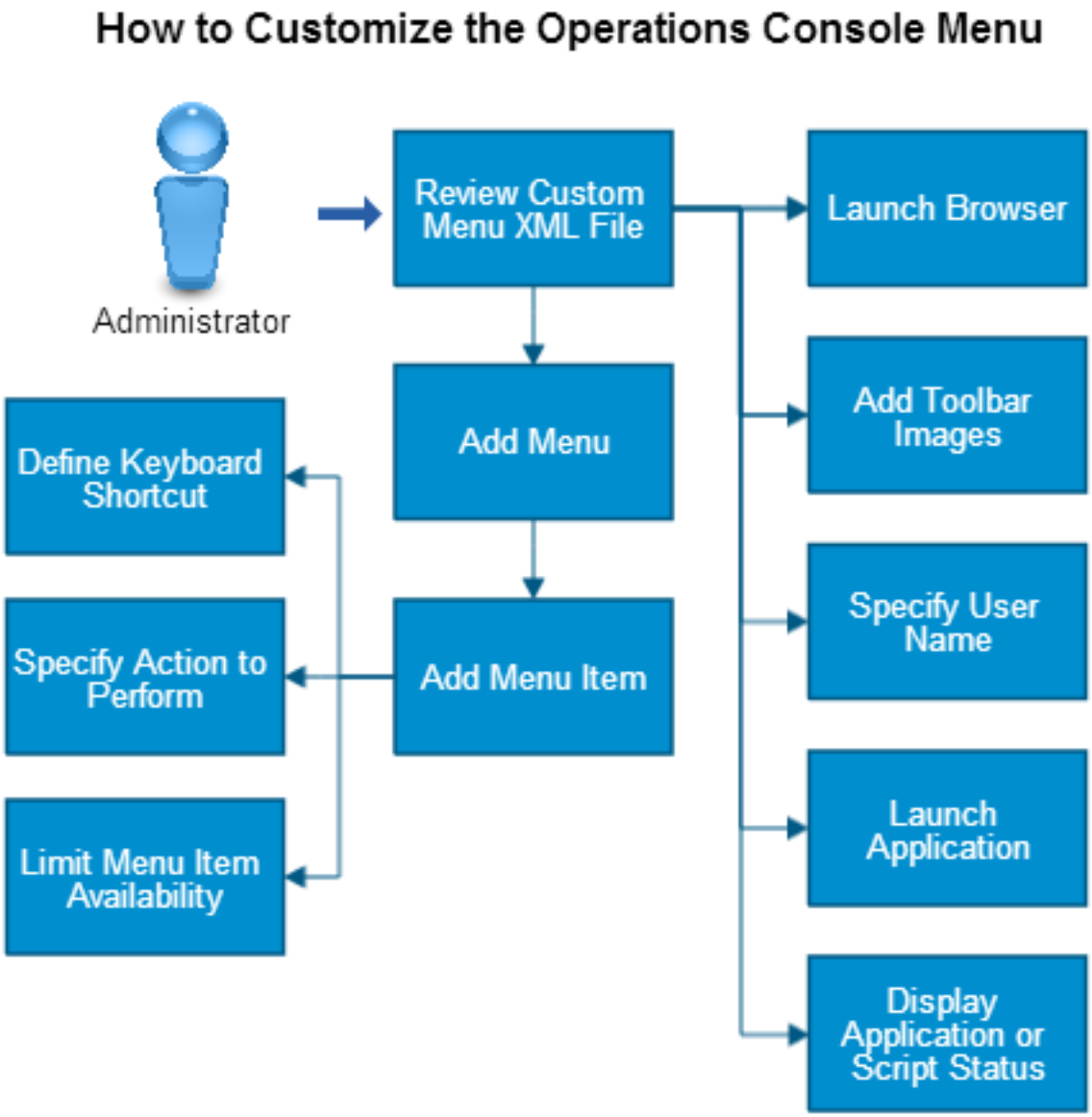
How to Customize the Operations Console Menu

Contents

As an administrator, you can add new menus and new menu items to the Operations Console interface. You can use new menu items to launch URLs, third-party applications, and scripts, and to pass parameters to them.

Use this scenario to guide you through the process:

Figure 41: how to customise operations console menu



1. Review the custom-menu-config.xml file.
2. Add a menu.
3. Add a menu item, then perform any of the following actions:
   - Define a keyboard shortcut.
   - Specify an action to perform.
   - Limit the availability of a menu item.
4. Launch a browser.
5. Add toolbar images.
6. Specify a user name.
7. Launch an application.

8. Display the status of a launched application or script.

custom-menu-config.xml File

The custom-menu-config.xml file in either of the following folders contains examples of how to add custom menus and custom menu items to the Operations Console:

- <SOI_HOME>\tomcat\webapps\sam\WEB-INF\console\config
- <SOI_HOME>\SamUI\webapps\sam\WEB-INF\console\config

Use the <menu> and <item> XML elements to create menus and menu items. The <menu> element can enclose one or more <item> elements that define the commands on the menu. The <item> element can enclose several other elements that define how the menu item appears and behaves. The following table describes each element:

Element	Parent Element	Description
<menu>	<root>	Defines the menu. The name attribute defines the name of the menu.
<separator>	<menu>	Defines a separator line. Use it before an <item> element.
<item>	<menu>	Defines an item on a menu. Use the name attribute to define the item name.
<privilege>	<item>	Associates a privilege to the menu item. A user who does not have this privilege cannot access the menu item.
<toolbar-image>	<item>	Specifies the image to display for the menu item and its associated toolbar button.
<toolbar-image-rollover>	<item>	Specifies the toolbar image displayed when a user places the cursor hovers over the toolbar button.
<toolbar-image-disabled>	<item>	Specifies the toolbar image displayed when the functionality is disabled (not available to the user). A typical representation for this state is an image that is 80% dimmed.
<accelerator>	<item>	Defines a keyboard sequence that completes the action that the menu item defines.
<action>	<item>	Defines the action that the menu item performs.
<hot-key>	<item>	Underlines the first instance of the indicated letter as a keyboard shortcut, and performs the action when that letter is pressed.

Add a Menu

The <menu> element is used to create an Operations Console menu.

Follow these steps:

1. Open the custom-menu-config.xml file.

   NOTE
   The <root> element is the first element for this file. You must define all new menus inside the <root> element.

2. Add a new menu using the <menu> element. This element has one attribute, name, which defines the menu name.

NOTE
Some examples in the custom-menu-config.xml file show a fully qualified menu name that references a Java class. For example, com.aprisma.spectrum.app.swing.window.menu.Tool is the name attribute in the <menu> element for the Tools menu. You do not have to use a fully qualified name; simply use the exact text that you would like for the menu name.

3. Add items to the new menu using the <item> element and its child elements.

NOTE
For more information, see Add a Menu Item. If you do not specify menu items, the menu is not visible in the Operations Console.

4. Save the changes you have made to custom-menu-config.xml.
5. Restart the Operations Console to view and test the new menus.

Example: Add a New Menu

This example shows a new Connections menu.

```
<menu name="Connections">
   <item name="Ping Local">
    .
    .
    .
   </item>
   <item name="Launch Diagnostics">
    .
    .
    .
   </item>
</menu>
```

Add a Menu Item

To add an item to an existing menu, create an <item> element inside a <menu> element.

NOTE
The item is also added to the context menu.

Follow these steps:

1. Open the custom-menu-config.xml file.
2. Find the <menu> element to which you want to add items.
3. Create new menu items using the <item> element. This element has one attribute, name, which defines the name of the menu item.
4. Add toolbar images.
5. Specify an action to perform
6. (Optional) Define a keyboard shortcut.

NOTE
The <item> element has several child elements that define how the item behaves. For more information, see custom-menu-config.xml File and subsequent procedures.

7. Save the changes you have made to custom-menu-config.xml.
8. Restart the Operations Console to view and test the new menu items.

Example: Add a New Menu Item

This example adds a menu item named Ping Local to a menu named Connections.

```
<menu name="Connections">
   <item name="Ping Local">
     <accelerator modifiers="2">VK_I</accelerator>
     <action>
       <filter>
          <has-attribute>AttributeID.NETWORK_ADDRESS</has-attribute>
       </filter>
       <context>com.aprisma.spectrum.app.topo.client.render.ModelContext
       </context>
       <context>com.aprisma.spectrum.app.alarm.client.group.AlarmContext
       </context>
       <launch-application>
          <platform>
             <os-name>Windows 9x</os-name>
             <command>command.com /c start "Local ping {0}" cmd.exe /c
             "ping.exe {0} &#38;&#38; pause"</command>
          </platform>
          <platform>
             <os-name>Windows</os-name>
             <command>cmd.exe /c start "Local ping {0}" cmd.exe /c "ping.exe
             {0} &#38;&#38; pause"</command>
          </platform>
          <platform>
             <command>/usr/dt/bin/dtterm -e ping -s {0}</command>
          </platform>
          <param>
             <attribute>AttributeID.NETWORK_ADDRESS</attribute>
          </param>
       </launch-application>
     </action>
   </item>
</menu>
```

Define a Keyboard Shortcut

The optional child attribute <accelerator> of a menu item lets you specify a keyboard shortcut that completes a menu action. It consists of the following items:

- The *modifiers* attribute is an integer that indicates the key or keys to use with a text character:
  1 = Shift
  2 = Ctrl
  3 = Ctrl+Shift
  8 = Alt
  9 = Alt+Shift
  10 = Ctrl+Alt
- The code *VK_X* where *VK_* is fixed code and *X* is the capital letter of the text character.

Example: Define a Keyboard Shortcut

This example specifies that the menu action is performed by pressing Ctrl+L.

```
<accelerator modifiers="2">VK_L</accelerator>
```

Specify an Action to Perform

The <action> element specifies the action that the menu item performs. You can use the child elements shown in the following table to specify a particular action.

The <context> element specifies when the menu item is active so that the action can be run. This applies to both the standard and the shortcut menus. You can specify a ModelContext, which indicates that the action is available when you select a CI. You can also specify an AlarmContext, which indicates that the action is available when you select an alert.

<context>com.aprisma.spectrum.app.topo.client.render.ModelContext

</context>

<context>com.aprisma.spectrum.app.alarm.client.group.AlarmContext

</context>

You can specify one or both contexts. If no specified context matches the current window context, the menu item is disabled. If no contexts are specified, the menu item is displayed in all contexts.

The following table describes the elements used to implement an action.

Element	Parent Element	Description
<context>	<action>	Specifies when the menu item is active so that you can perform the action.
<filter>	<action>	Limits the display of menu items.
<has-attribute>	<filter>	Specifies the attribute on which to filter.
<and>, <or>, <value>, <equals>	<filter>	Creates an expression for use with a filter.
<launch-browser>	<action>	Opens a browser. See Launch a Browser for more information.
<launch-sso-browser>	<action>	Opens a browser and includes in the URL a single sign-on token associated with the current session. You can use this token to reauthenticate the session across integrated web applications instead of prompting repeatedly for a user name and password. See Launch a Browser for more information.
<url>	<launch-browser>	Specifies the URL to launch in the browser.
<launch-application>	<action>	Opens an application. See Launch an Application for more information.
<launch-web-server-script>	<action>	Launches a script available on the web server. See Launch a Web Server Script for more information.
<display-output>	<launch-application>, <launch-web-server-script>	Displays the output from the script you ran. For more information, see Display the Status of a Launched Application or Script.
<display-exit-status>	<launch-application>, <launch-web-server-script>	Displays the exit status of a launched script.
<command>	<launch-application>, <launch-web-server-script>, <platform>	Specifies the application or script that the menu item launches.
<platform>	<launch-application>	Used with <os-name> to specify the application to launch based on the client operating system.

<os-name>	<platform>	Used with <platform> to specify the application to launch for the client operating system.
<param>	<url>, <command>	Specifies a parameter that is passed to a browser, program, or script.
<attribute>	<param>	Specifies an attribute used as a parameter.

Limit the Availability of Menu Items

The <filter> element specifies a filter that further restricts the enabled state of the menu item. You can filter on any attribute of the selected context. For example, the following code shows that the action needs the IP address of the alerted CI. Therefore, it is enabled only if the CI has the IP address attribute populated.

```
<filter>
   <has-attribute>AttributeID.NETWORK_ADDRESS</has-attribute>
</filter>
```

You can specify complex attribute filters with any combination of nested, and, and or filters.

For more information, see Attribute Filter Syntax.

Example: Nested Filters

The following example enables the item if the selected model has the Network_Address attribute and the Condition (ID 0x1000a) attribute is RED.

```
<filter>
   <and>
      <has-attribute>AttributeID.NETWORK_ADDRESS</has-attribute>
    <equals>
      <attribute id="AttributeID.CONDITION">
         <value>43</value> <!--red-->
      </attribute>
    </equals>
   </and>
</filter>
```

Attribute Filter Syntax

The file <SOI_HOME>\SamUI\webapps\sam\WEB-INF\common\schema\attribute-filter.xsd contains the complete syntax for attribute filters.

NOTE

- For examples of how to filter menu items, see Limit the Availability of Menu Items.
- If you use an attribute other than the attributes listed in the following tables, specify the attribute using its hexadecimal attribute ID.

The following table defines commonly used attributes where an attribute ID is expected.

Constant	Attribute
AttributeID.NETWORK_ADDRESS	Network Address
AttributeID.MTYPE_NAME	CI Class name
AttributeID.MODEL_OBJECT	CI Handle
AttributeID.MODEL_NAME	CI Name

AttributeID.MODEL_CLASS	CI Class
AttributeID.CONDITION	Condition
AttributeID.DOMAIN_ID	SA Manager ID
AttributeID.DOMAIN_NAME	SA Manager name

You can use the constants in the following table for the indicated alert attributes:

Constant	Attribute
AlarmAttrID.ACKNOWLEDGED	Acknowledged
AlarmAttrID.ALARM_ID	Alert ID
AlarmAttrID.ALERTDETAIL	Alert Detail
AlarmAttrID.ASSIGNED	Assignment
AlarmAttrID.CONNECTOR_NAME	Connector Name
AlarmAttrID.DESCRIPTION	Description
AlarmAttrID.CREATION_DATE	Creation Date
AlarmAttrID.EVENT_SOURCE	Source Name
AlarmAttrID.EVENT_SOURCE_ID	Source Alarm ID
AlarmAttrID.EVENT_OCCURRED	Source Creation Date
AlarmAttrID.PRIORITY	Service Impact
AlarmAttrID.SEVERITY	Severity
AlarmAttrID.SITUATION_TYPE	Category
AlarmAttrID.TICKET_ID	Ticket ID
AlarmAttrID.USER_CLEARABLE	User Clearable

Launch a Browser

The <launch-browser> element lets you launch a specified URL in a browser and pass parameters to the URL. These parameters can be hard-coded values or values from model attributes.

> **NOTE**
> For more information about parameters, see the definition of <param-type> in the file <SOI_HOME>\SamUI\webapps\sam\WEB-INF\common\schema\basic-config.xsd.

Example: <launch-browser> Code

This example launches the default browser on the client computer. The <url> element specifies the URL pattern. You can specify parameters to substitute in the URL pattern by enclosing the parameter number (starting at 0) in curly braces {}. You then specify <param> elements for each parameter.

```
<launch-browser>
<url>http://{0}</url>
<param>
<attribute>AttributeID.NETWORK_ADDRESS</attribute>
</param>
</launch-browser>
```

CA SOI processes the <param> elements in order so that the first one corresponds to the 0th parameter in the URL pattern. A <param> element has a specific syntax. The most common element is <attribute>, which substitutes the value

of the specified attribute for the selected context. In this example, the value of the Network Address attribute is substituted in the URL pattern.

> **WARNING**
> See Characters in URLs for information about the characters to use and not to use in URLs.

Characters in URLs

This section discusses the characters to use or avoid in URLs.

Standard Characters

The URL formatting must adhere to the standards published in the Internet Engineering Task Force (IETF) RFC 1738. Use of non-standard characters in URLs results in unreliable browser performance including the browser not locating the specified web page.

Spaces and Commas

If you use spaces or commas, convert them to their ASCII equivalent. URL encoding consists of a percent (%) symbol followed by the two-digit hexadecimal representation (case-insensitive) of the ISO-Latin code point for the character.

- For spaces, use %20
- For commas, use %2C

> **NOTE**
> Some browsers encounter problems processing URLs even with this encoding.

Ampersands

If you use ampersands, convert them to &amp.

CDATA

You can put URLs inside a CDATA section so that they are not parsed, which avoids possible problems with URLs and the XML parser.

Comply with the CDATA requirements, including the following:

- A CDATA section cannot contain the string "]] >"; therefore, nested CDATA sections are not allowed.
- Use no spaces or line breaks inside the "]] >" string.

Unsafe Characters

The following table lists characters that are easily misinterpreted in URLs. Always substitute these characters with % followed by the hexadecimal code points listed in the table. For example, use %20 to represent a space in a URL.

Character	Code Points (Hex)
Space	20
Quotation marks (")	22
Less Than symbol (<)	3C
Greater Than symbol (>)	3E
Pound character (#)	23
Percent symbol (%)	25
Left Curly Brace ({)	7B
Right Curly Brace (})	7D
Vertical Bar/Pipe (|)	7C
Backslash (\)	5C

Caret (^)	5E
Tilde (~)	7E
Left Square Bracket ([)	5B
Right Square Bracket (])	5D
Grave Accent (`)	60

Reserved Characters

The following table lists characters that have special uses in URLs. Substitute such characters with % followed by the hexadecimal code points listed in the table when they are used as regular text and not in their special role. For example, use %24 to represent a plain-text dollar sign ($) in a URL.

Character	Code Points (Hex)
Dollar ($)	24
Ampersand (&)	26 (or "&amp" as explained earlier)
Plus (+)	2B
Comma (,)	2C
Forward slash/Virgule (/)	2F
Colon (:)	3A
Semi-colon (;)	3B
Equals (=)	3D
Question mark (?)	3F
At symbol (@)	40

Add Toolbar Images

Toolbar images have the following states, which you specify in your menu item definition. The elements for toolbar states are as follows:

- <toolbar-image>
- <toolbar-image-rollover>
- <toolbar-image-disabled>

For more information, see custom-menu-config.xml File.

You can use the following formats for toolbar images: .png, .gif, .jpg, and .jpeg. We recommend the size 24 x 24 pixels.

Create the images directory at the following location: <SOI_HOME>\ui\tomcat\custom\images directory. Store your custom images at this location. When you reference an image in this directory, specify the path from the images directory, for example, images\myimage.png.

Example: Specify a Toolbar Image

This example points to the hints.gif file:

```
<toolbar-image>images/hints.gif</toolbar-image>
```

Specify a User Name

You can pass a user name to an application, web browser, or script. Use the following expression to specify the name of the logged-in user:

```
<param>
```

```
    <expression>
       com.aprisma.spectrum.app.util.context.DefaultApplicationContext.
       getGlobalParameter(com.aprisma.spectrum.app.util.context.ApplicationContext.
       USER_PARAMETER_NAME)
    </expression>
 </param>
```

Example: Pass User Name to Browser

This example opens a browser and passes a user name to it.

```
 <launch-browser>
    <url> http://acme.com?user={0}</url>
    <param>
    <expression>
       com.aprisma.spectrum.app.util.context.DefaultApplicationContext.
       getGlobalParameter(com.aprisma.spectrum.app.util.context.ApplicationContext.
       USER_PARAMETER_NAME)
    </expression>
    </param>
 </launch-browser>
```

Launch an Application

The <launch-application> element lets you start a command or program.

<command> Element

The <command> element specifies the command or program to run. You can provide the path to the command or program in one of the following ways:

Environment variable

In a Solaris environment use the PATH variable. To create an environment variable on Windows, right-click My Computer, select Properties and Advanced, and click the Environment Variables button.

- **Absolute path**
  The path must be the same on each CA SOI client. Path statements on Windows should have a double backslash instead of a single backslash, for example:
  ```
  C:\\Windows\\system32\\cmd.exe
  ```

The following syntax rules apply to the <command> element:

- Spaces delimit command arguments.
- Spaces in an argument are surrounded by quotation marks or preceded by the backslash escape character (\).
- Quotation marks in an argument are preceded by the backslash escape character (\).
- CA SOI automatically surrounds command arguments that contain commas with quotation marks, which is important to know if you parse a numeric argument that contains commas.
- CA SOI replaces arguments that return null or have a string length of zero with empty quotation marks ( ).
  <validate> Element

The <validate> element verifies that the command or program exists on the client and has run permissions. If either of these conditions is not met during startup, the associated menu item is not added to the menu.

If the <validate> element is not used, the menu item is always added to the menu, but its state is determined by the value of other elements.

The <validate> element requires an absolute path in the <command> element, as shown in the following example:

```
<launch-application>
   command>c:\\windows\\system32\\notepad.exe</command>
   <validate/>
</launch-application>
```

Examples: Launch an Application

The following are two examples for launching an application:

- This example launches an application called myapp on the client machine and passes the IP address of the selected model. As with the <launch-browser> action, you can substitute any number of parameters.

```
<launch-application>
   <command>myapp {0}</command>
   <param>
      <attribute>AttributeID.NETWORK_ADDRESS</attribute>
   </param>
</launch-application>
```

- This example uses the <platform> element to specify commands for different platforms. The <os-name> element specifies the operating system name and the <command> element specifies the command to run on that operating system. The <os-name> element is optional. If you do not specify <os-name>, the associated command is the default such that if no other platforms match, the default command runs.

```
<launch-application>
   <platform>
      <os-name>Windows</os-name>
      <command>cmd.exe /c start "ping {0}" cmd /c "ping.exe {0}
      &#38;&#38;pause"</command>
   </platform>
   <platform>
      <os-name>SunOS</os-name>
      <command>>/usr/dt/bin/dtterm -e ping {0}</command>
   </platform>
   <param>
      <attribute>AttributeID.NETWORK_ADDRESS</attribute>
   </param>
</launch-application>
```

At runtime, the specified OS names are compared to the OS name returned by the *os.name* Java property. A best-match algorithm lets you specify only a prefix of the OS name. The following are valid OS names:

- SunOS for the Solaris platform
- Windows for all Windows platforms
- Windows 9x for Windows 95/98
- Windows 2000 for Windows 2000
- Windows XP for Windows XP
- Windows Vista for Windows Vista and Windows Server 2008
- Linux for the Linux platform
- Mac for the Macintosh platform

If no specified platforms match, the associated menu item is disabled.

Display the Status of a Launched Application or Script

Use the <display-exit-status> and <display-output> elements with <launch-web-server-script> and <launch-application> to display the exit status and the output from the script or application.

By default <display-exit-status> displays Success if the exit code is 0 and Failed with error code # otherwise. You can change the default behavior by specifying <status> child tags that map an exit code to a custom message to display.

Example: <display-exit-status> Code

This example maps status codes 1, 2, and 3 to specific message strings. The last status code specifies default=true, mapping all other error codes except 0, which by default maps to Success. If exit code 0 does not indicate success, you can override it with a <status> tag. The {0} in the message string substitutes the exit code.

```
<display-exit-status>
   <status code="1">Could not open file</status>
   <status code="2">Bad parameter</status>
   <status code="3">Could not connect to the server</status>
   <status default="true">Unknown error code {0}</status>
</display-exit-status>
```

By default, <display-output> displays both the standard output and standard error output from the process. You can display only the standard output by specifying:

```
<display-output stdout="t"/>
```

or only the standard error output by specifying:

```
<display-output stderr="t"/>
```

> **NOTE**
> The <display-exit-status> and <display-output> elements can be used only for command line applications or scripts and not GUI applications. The interface waits for the script to finish before being available to the user again.

Launch Web Server Scripts

As an administrator, you enable launching of web server scripts. This functionality was previously disabled due to a security issue.

Follow these steps:

1. Locate and open the following file on the UI Server:
   <SOI_HOME>\jsw\conf\soi-user-interface.conf
2. Add the following entry:
   `wrapper.java.additional.22=-DALLOW_WEB_SERVER_SCRIPT=1`
3. (Optional) Add a parameter to allow the script execution in a specified folder only:
   `wrapper.java.additional.23=-DWEB_SERVER_SCRIPT_PATH="Path"`
   – *Path*
     Specifies the full path of a folder on your UI Server.
     Example: "C:/SOIUI/serverscripts"

     > **NOTE**
     > The additional parameters numbers, 11 and 12, assume that, before this change, the last additional parameter was 10. In general, you increase each new parameter number by one.
4. Restart the CA SAM User Interface Server service to activate the change.

NOTE
If WEB_SERVER_SCRIPT_PATH is configured, the script must be in the specified folder (or one of its subfolders).

Include TenantID in Correlation

CA SOI correlates items coming from different sources based on the Correlation Data Priority list defined in the USM Schema for each item class. By default, the TenantID property is not part of correlation. Therefore, items with the same correlation properties coming from different tenants are correlated into one notebook.

You can customize CA SOI to include TenantID in correlation to partition the tenant data.

Follow these steps:

1. Locate and edit the following file on the SA Manager:

   ```
   SOI_HOME\jsw\conf\soi-manager.conf
   ```

2. Locate the "Java Additional Parameters" section, and add the following line at the end of the section (where the last sequence number in this example is 22):
   wrapper.java.additional.23=-DCORRELATE_WITH_TENANTID=1
   Consider the following items:
   – The sequence numbers must be both consecutive and unique. Any lines starting with the number sign (#) are commented out and are not part of the sequence.
   – For installations on clusters, the CA SOI HA Kit adds a parameter at the bottom of the file which needs to be taken into account.
   – Setting CORRELATE_WITH_TENANTID to any value other than "1" disables this option.
3. (Optional) Add the following parameter to have all item labels prefixed with “<TenantID>-“:
   wrapper.java.additional.24=-DLABEL_WITH_TENANTID=1
   The sequence number should be the last number incremented by one, and any value other than "1" disables this option.
4. Save the file.
5. If possible, remove existing connectors publishing data with TenantID. Remove connectors from the Administration tab on the Dashboard after stopping the connector service so that the connector is not disabled.
   Note: The SA manager attempts to de-correlate existing items once the TenantID is included in correlation in case existing data from connectors cannot be removed temporarily. However, this is not as clean as removing old data and republishing it with TenantID correlation turned on.
6. Stop the CA SAM Application Server service.
7. Remove the SOI_HOME\tomcat\work\soi.correlation.cache file if it exists. This file only exists when the CA SAM Application Server service is down. With the cache file removed, the SA Manager will recorrelate all existing items on restart.
8. Start the CA SAM Application Server service. Be aware that startup can take much longer than normal because the SA Manager has to recorrelate all existing items and rebuild the correlation cache.
9. Start the connector services that you stopped when removing connectors in Step 5.
10. Update the connector policies to populate the TenantID field of the CI with the unique value.

Map CA Spectrum Global Collections to Alert Queues

You can customize the CA Spectrum connector to send Global Collection information for each CI using CI user attributes. You can then use the CI user attributes in alert queue criteria, which lets you view Global Collection alerts segregated into alert queues.

Follow these steps:

1. Review the *CA Spectrum Connector Guide* section ‘Add Additional Model Attributes ‘ and configure your connector to send the Global Collection attribute named ‘ CollectionsModelNameString’. Review the section ‘Map the new attributes to USM data’ on how to map the attribute to CI user attributes in CA SOI.
2. Configure CA SOI to perform delta processing on the CI user attributes so that any time you restart the connector, any changes when the connector was offline or any changes to the Global Collections are sent to CA SOI.
   On connector startup, the CIs are sent with the CIUserAttribute, which you can use to create alert queues for managed and unmanaged CIs.

   NOTE

   The CA Spectrum Connector 2.0.0.195 does not send update events when a CI changes from one Global Collection to another. If CIs change Global Collections, restarting the connector (ensuring that you applied the configuration from Step 2) allows the CIs in CA SOI to be updated with the correct Global Collection information.

Installation Maintenance

This section describes how to make maintenance updates to your installation, such as password changes, moving components, and so on.

Password Maintenance

Contents

As an administrator, use the following password encryption utilities help you with the maintenance of passwords:

- **EncryptSAMCreds**
  Encrypts any password in CA SOI components and also lets you update CA SOI with new passwords. This utility is available at <SOI_HOME>\tools.
- **Encrypter**
  Encrypts any password in CA Catalyst components. This utility is available at <SOI_HOME>\Tools\CatalystEncrypt.

You can use these utilities to change key passwords after installation if necessary, such as the administrator password and the SA Store connection password. Any changes to passwords using the CA SOI utility do not update records of these passwords in CA Catalyst components. For example, if you change the CA SOI administrator password using EncryptSAMCreds.bat, you also change this password using encrypter.bat for the change to occur in the CA Catalyst Registry.

Encrypt Passwords in CA SOI Using the EncryptSAMCreds Utility

As an administrator, use the EncryptSAMCreds utility to perform the following actions:

- Manually encrypt any password that a CA SOI component uses. This ability is useful if you want to encrypt an existing password and then insert the encrypted password in a configuration file. Manual encryption is also useful if you want to change any password manually and then encrypt the new password for insertion in the appropriate configuration files.
- Automatically encrypt a new password for the CA SOI administrator user or SA Store connection and insert the new password into the appropriate CA SOI component files.

  NOTE
  Passwords for CA SOI users are maintained in CA EEM. Change those passwords directly in CA EEM or through the Dashboard. Only the administrator user credentials also require you to update them in configuration files.

Follow these steps:

1. Shut down all CA SOI services on the affected systems (for example, SA Manager, UI Server, and connector systems).
2. Open a command prompt, navigate to SOI_HOME\tools, and run the following command:
   ```
   EncryptSAMCreds <password>
   ```
   An encrypted password is generated.
3. Paste the encrypted password into the appropriate configuration files on all affected systems.

   NOTE
   Ensure that you replace the password in all necessary configuration files according to the ensuing procedures for the administrator user or database user. You can change most other required passwords in the Administration UI without using this utility.
4. Start the CA SOI services.
   The password change takes effect.

Encrypt Passwords in CA Catalyst Components Using the Encrypter Utility

As an administrator, use the CA Catalyst encryption utility (encrypter.bat) to encrypt any password in CA Catalyst components. You can change any password manually and then insert the new encrypted password into the appropriate configuration files.

Follow these steps:

1. Stop the CA SOI services on the affected systems.
2. Open a command prompt, navigate to <SOI_HOME>\CatalystEncrypt, and run the following command:
   ```
   encrypter <password>
   ```
   An encrypted password is generated.
3. Paste the encrypted password into the required configuration files.

   NOTE
   Ensure that you replace the password in all necessary configuration files according to the ensuing procedures for the administrator user, database user, or CA EEM.
4. Start the CA SOI services.
   The password change takes effect.

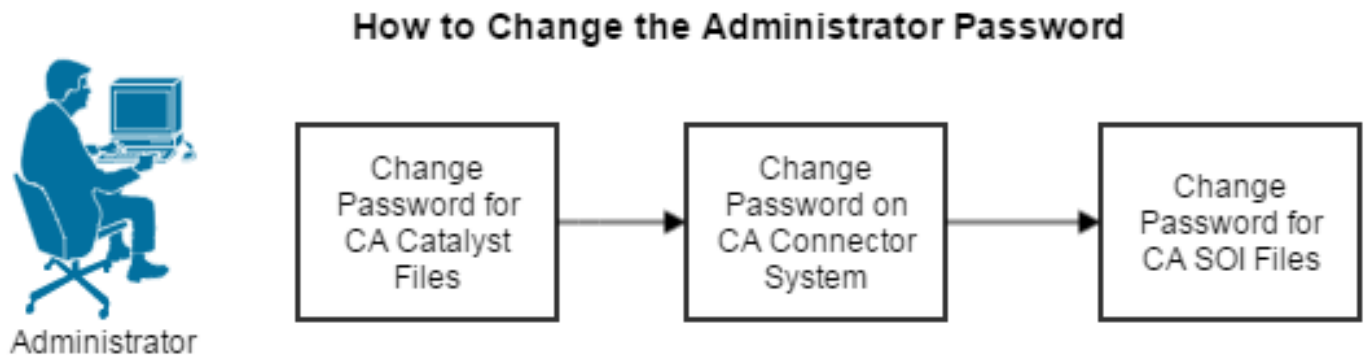
How to Change the Administrator Password

Contents

As an administrator, you can change the administrator password after installation for security.

Use this scenario to guide you through the process:

Figure 42: Change the Administrator Password



1. Change the administrator password in all CA Catalyst files.
2. Change the administrator password on the CA Catalyst connector system.
3. Change the administrator password in all CA SOI files.

Change the password in all locations for the product to work. Keep in mind the password character restrictions to avoid changing to an invalid password.

Change the Administrator Password in all CA Catalyst Files

Change the administrator password in all CA Catalyst files using the encrypter.bat utility for the change to occur in all files stored in and used by the CA Catalyst Registry.

Follow these steps:

1. Stop the CA SAM Application Server service on the SA Manager.
2. Open a command prompt, navigate to <SOI_HOME>\Tools\CatalystEncrypt on the SA Manager system, and run the following command:

   ```
   encrypter new_password
   ```

   An encrypted password appears.
3. Open the following files located at <SOI_HOME>\tomcat\registry\topology\physical\node0\sor, paste the new encrypted password in the listed sections, and save the files:
   - restserver.xml

     ```
     <tns:jmsImplementation inj:injID="jms">
           <tns:brokerURL>tcp://server:61616</tns:brokerURL>
           <tns:userName>samuser</tns:userName>
           <tns:password>password</tns:password>
     <tns:trace inj:injID="trace">
     . . .
           <tns:userName>samuser</tns:userName>
           <tns:password>password</tns:password>
     <tns:trace inj:injID="trace">
     . . .
           <tns:userName>samuser</tns:userName>
           <tns:password>password</tns:password>
     wsman.password=password
     ```
   - sorapp.xml
   - ssaserver.xml

- wsman.properties

4. Start the CA SAM Application Server service.
5. Run registryloader.bat located at SOI_HOME\tomcat\registry. Ignore any "log4j:WARN" messages.
   The Registry loads an updated record of all files in which you changed the password.
6. Restart the CA SAM Application Server service.
   The Registry reconnects to all product components using the new password.
7. Access the CA Catalyst Registry and log in using the old administrator password.
8. Click Users and Roles on the left menu of the Registry home page.
9. Click Change My Password.
10. Enter the old password and new password in the provided fields and click Change.
    The Registry access now uses the new password.
11. Close the Registry, open the <SOI_HOME>\wso2registry\repository\conf file, paste the new password into the following section, and save the file:
    - user-mgt.xml

      ```
      <AdminUser>
         <UserName>samuser</UserName>
         <Password>password</Password>
      </AdminUser>
      ```

    - Cipher-text.properties

      ```
      UserManager.AdminUser.Password==encrypted password
      ```

    The password is changed in all CA Catalyst files. If the UI Server is installed with the SA Manager, the password change is complete. If you installed the UI Server on a separate system, continue with Step 12.
12. (Standalone UI Server only) Repeat Step 11 in the same file on the UI Server, pasting the encrypted password from the SA Manager.

Change the Administrator Password on all CA Catalyst Connector Systems

If you change the CA SOI administrator password, you also change the password in the IFW Proxy on the CA Catalyst Container. Keep this password synchronized in CA SOI and the IFW Proxy to avoid connection issues.

Follow these steps:

1. Open a command prompt on the connector system, navigate to <CATALYST_HOME>\<containerName>\ifw, and run the following command:

   ```
   EncryptSAMCreds newpassword
   ```

   - **newpassword**
     Defines the new password for the administrator user.

   The command generates an encrypted password.
2. Copy the password.
3. Open the <CATALYST_HOME>\<containerName>\ifw\resources\configurations\SSA_IFW_*servername*.xml file, paste the new encrypted password into the password property, and save and close the file.
4. Open the <CATALYST_HOME>\<containerName>\registry\topology\physical\<connectorserver>\ifw \eventManagementServer.properties file, paste the new encrypted password into the password property, and save and close the file.
5. Restart the CA Catalyst Container *ContainerName* service.
   The password change takes effect.

Change the Administrator Password in all CA SOI Files

You change the administrator password in all CA SOI files using the EncryptSAMCreds.bat utility. Change the password on the SA Manager, UI Server, and connector systems.

Follow these steps:

1. Stop CA SOI services on the SA Manager system.
2. Open a command prompt, navigate to <SOI_HOME>\tools, and run the following command:

   ```
   EncryptSAMCreds <old password> <new password> SAMUser
   ```

   The command updates the CA SOI administrator password in all necessary configuration files for CA SOI components on the SA Manager system. The command also updates the configuration files for the UI Server and connectors that exist on the same system.
3. Repeat Steps 1-2 on the UI Server if it is installed on a separate server than the SA Manager, using the following command:

   ```
   EncryptSAMCreds <old password> <new password> UIServer
   ```

   > **WARNING**
   > Do not run this command if the UI Server exists on the same system as the SA Manager. Running this command on the SA Manager may cause a password mismatch with UI Server and SA Manager files. If you ran this command on the SA Manager by mistake, back out the change by running the command again and reversing the old and new passwords (to reestablish the old password).

   Running the command on the UI Server system updates the CA SOI administrator password in all necessary configuration files only on the UI Server system.
4. Repeat Steps 1-2 on all systems with connector installations (besides the SA Manager), substituting the following command:

   ```
   EncryptSAMCreds <old password><new password> Connector
   ```

   This command updates the administrator password in all necessary configuration files on the connector system. Run the command on all systems that contain a connector. If a system contains multiple connectors, you need to run the command once on that system.
5. Start all CA SOI services on the SA Manager, UI Server, and connector systems.

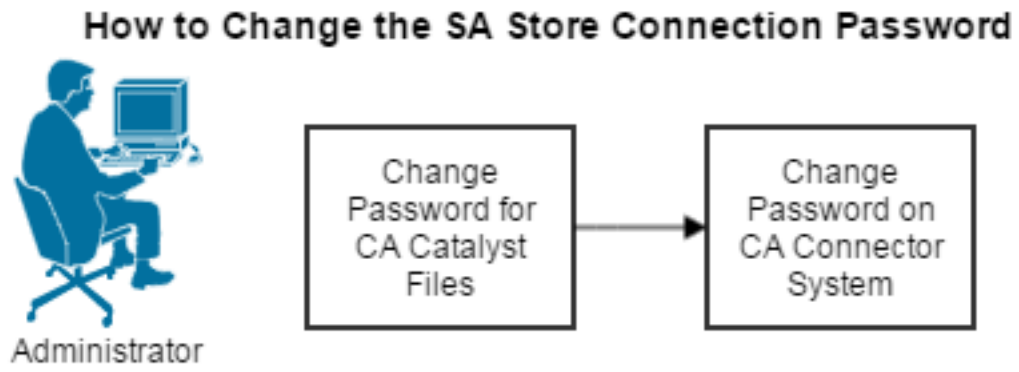
How to Change the SA Store Connection Password

Contents

As an administrator, you can change the connection password for the SA Store database in CA SOI. You change this password on the SA Manager server. Complete the following process to change the SA Store connection password in all necessary areas:

Use this scenario to guide you through the process:

Figure 43: Change the SA Store Connection Password



1. Change the SA Store connection password in all CA Catalyst files.
2. Change the SA Store connection password in all CA SOI files.

Change the SA Store Connection Password in all CA Catalyst Files

You must change the SA Store connection password in all CA Catalyst files using the encrypter.bat utility for the change to occur in the CA Catalyst Registry.

Follow these steps:

1. Stop the CA SAM Application Server service on the SA Manager.
2. Open a command prompt, navigate to <SOI_HOME>\Tools\CatalystEncrypt on the SA Manager system, and run the following command:
   ```
   encrypter <new password>
   ```
   An encrypted password appears.
3. Open the following files located at <SOI_HOME>\tomcat\registry\topology\physical\node0\sor, paste the new encrypted password in the listed sections, and save the files:
   - restserver.xml
   - sorapp.xml
   - ssaserver.xml
   - ssaweb.xml
   ```
   <tns:dataSource inj:injID="dbconfig">
   . . .
         <tns:username>sa</tns:username>
         <tns:password>password</tns:password>
   ```
4. Open the <SOI_HOME>\ServiceDiscovery\connectivityContext.xml file, paste the new encrypted password into the password property value in the '<!-- connectivity to the persistence store database -->' and '<!-- connectivity to the SSA database -->' sections, and save and close the file.
   All CA Catalyst configuration files contain the new encrypted password.
5. Access the following files located at <SOI_HOME>\wso2registry\repository\conf directory, paste the new encrypted password into the listed sections, and save the changes:
   - registry.xml
     ```
     <dbConfig name="wso2registry">
        <userName>sa</userName>
        <password>password</password>
     ```
   - user-mgt.xml

```
<Configuration>
...
   <Property name="userName">sa</Property>
   <Property name="password">password</Property>
```

– Cipher-text.properties

```
UserManager.Configuration.Property.password=encrypted password
...
wso2registry.wso2registry.password=encrypted password
```

The password is changed in all CA Catalyst files.

6. Start the CA SAM Application Server service.
   The Registry reconnects to all product components using the new password.
7. Run registryloader.bat located at <SOI_HOME>\tomcat\registry. Ignore any "log4j:WARN" messages.
   The Registry loads an updated record of all files in which you changed the password.
   The password is changed in all CA Catalyst files. If the UI Server is installed with the SA Manager, the password change is complete. If you installed the UI Server on a separate system, continue with Step 8.
8. (Standalone UI Server only) Repeat Step 5 in the same files on the UI Server, pasting the encrypted password from the SA Manager. Now restart the UI Server Service.

Change the SA Store Connection Password in all CA SOI Files

Change the SA Store connection password in all CA SOI files using the EncryptSAMCreds.bat utility.

Follow these steps:

1. Shut down the CA SAM Application Server and CA SAM User Interface Server services on the SA Manager system.
2. Open a command prompt, navigate to <SOI_HOME>\Tools, and run the following command:
   ```
   EncryptSAMCreds <old password> <new password> Database
   ```

   NOTE

   Use the word 'Database' as written in the command syntax, not the database name.

   This command updates the database password in all necessary configuration files on the SA Manager system.
3. Copy the <SOI_HOME>\tomcat\lib\hibernate.cfg.xml file on the SA Manager to <*tomcat_path*>\webapps\SpectrumSA on the report server, remove the <!DOCTYPE...> element at the top of the file, and save and close the file.

   NOTE
   tomcat_path refers to the default BusinessObjects Tomcat installation directory of C:\Program Files\CA\SC\CommonReporting3\Tomcat55.
4. Select Start, Programs, BusinessObjects XI 3.x, BusinessObjects Enterprise, Central Configuration Manager.
5. Select Apache Tomcat and Server Intelligence Agent, click Stop, and then click Start when the services stop.
   The database password is changed on the report server.
6. Access the Control Panel on the report server and double-click Administrative Tools, then Data Sources (ODBC).
7. Click the System DSN tab, select SAMStore, and click Configure.
8. Click Next, enter the new password in the Password field, and click Next again.
   The connection is verified with the new password.
9. Click Finish.
10. Open the <SOI_HOME>\Reports\samreports.xml file on the report server, enter the new password in the <password> attribute, and save and close the file.
    Future report redeployments will use the new database connection password.
11. Start the CA SAM Application Server and CA SAM User Interface Server services.

The password change takes effect.

Communication Port Maintenance

Contents

As an administrator, you can change the CA SOI communication ports specified during installation. It is important that you change the port number in every file listed, or communication errors can occur.

Change the SA Manager TCP Port Number

You can change the TCP port number for SA Manager communication. The SA Manager TCP port number is 7090 by default.

> **NOTE**
> This procedure is only for changing the SA Manager port number for HTTP communication.

Follow these steps:

1. Stop all CA SOI services on the SA Manager, UI Server, and all plugin systems.
2. Open the following files on the SA Manager system, change the **portnumber** value in the listed sections, and save the changes:
   - <SOI_HOME>\tomcat\conf\server.xml
     ```
     <Connector port="portnumber" maxHttpHeaderSize="16384
     maxthreads="150" minSpareThreads="25" maxSpareThreads="75"
     enablelookups="false" redirectPort="7493" acceptCount="100"
     connectionTimeout="20000" disableUploadTimeout="true"
     compression="on" compressableMimeType="text/html,text/plain,text/
     css,text/javascript" />
     ```
   - <SOI_HOME>\resources\Configurations\SSA_IFW_*servername*.xml
     ```
     <CertGenService URL="http://servername:portnumber/ucf-certgen/CertGenService"
     ```
   - (Optional) <SOI_HOME>\plugin\UniversalConnector\conf\uc.properties
     ```
     uc.port=portnumber
     ```

     > **NOTE**
     > This change is only required if the Universal connector is installed.

   All SA Manager files contain the new TCP port number.
3. Open the following file on the UI Server, change the current TCP port number in the following section, and save the changes:
   - <SOI_HOME>\SamUI\webapps\sam\server-config.xml
     ```
     <root>
           . . .
           <manager>
           < port>portnumber</port>
           </manager>
     </root>
     ```

   All UI Server files contain the new TCP port number.
4. Open the <SOI_HOME>\resources\Configurations\SSA_IFW_*servername*.xml file on all connector systems, change the current TCP port number in the following section, and save the changes:
   ```
   <CertGenService URL="http://servername:portnumber/ucf-certgen/CertGenService"
   ```

   All connector files contain the new TCP port number.
5. Restart the CA SOI Manager service, and start all other services on the SA Manager, UI Server, and all connector systems.

Change the MQ Server TCP Port Number

You can change the TCP port number for MQ Server communication. The MQ Server TCP port number is 61616 by default.

Follow these steps:

1. Stop all CA SOI services on the SA Manager, UI Server, and all connector systems.
2. Open the following files on the SA Manager system, change the **portnumber** value in the listed sections, and save the changes:
   - <SOI_HOME>\tomcat\webapps\activemq-web\WEB-INF\caifwmq.xml
     ```
     <networkConnectors>
           . . .
           <networkConnector name="host1 and host2"
           uri="static://(tcp://host1:portnumber,tcp://host2:portnumber)"/>
           . . .
     </networkConnectors>
     . . .
     <transportConnectors>
           <transportConnector name="openwire"
           uri="tcp://server:portnumber
           ?wireFormat.maxInactivityDuration=-1" />
           . . .
     </transportConnectors>
     ```
   - <SOI_HOME>\tomcat\lib\jmsconnect.properties
     ```
     port=portnumber
     ```
   - <SOI_HOME>\resources\SSA_IFW_*servername*.xml
     ```
     <AMQ name="AMQClient" uuid="">
           <ConnectionInfo failover="TRUE" host="server"
           password="(encrypted pw)" port="portnumber" protocol="tcp"
           reconnectAttempts="240" reconnectTime="30000" user="catalyst"/>
           . . .
     </AMQ>
     ```
   - <SOI_HOME>\resources\mtc_*servername*.xml
     ```
     <ConnectionInfo ...port="portnumber"
     ```
   - <SOI_HOME>\resources\eventManagerServerConfig.xml
     ```
     <property name="port" value="portnumber"/>
     ```
   - <SOI_HOME>\tomcat\lib\eventManagerClientConfig.xml
     ```
     <property name="port" value="portnumber"/>
     ```
   - <SOI_HOME>\wso2registry\repository\conf\axis2.xml
     ```
     <parameter name="myTopicConnectionFactory">
        <parameter name="java.naming.provider.url">
        tcp://localhost:portnumber</parameter>
     ...
     <parameter name="myQueueConnectionFactory">
        <parameter name="java.naming.provider.url">
        tcp://localhost:portnumber</parameter>
     ...
     <parameter name="default">
        <parameter name="java.naming.provider.url">
     ```

```
tcp://localhost:portnumber</parameter>
```

– The following files located at <SOI_HOME>\tomcat\registry\topology\node0\sor:
  a. restserver.xml
  b. sorapp.xml
  c. ssaserver.xml

```
<tns:jmsImplementation inj:injID="jms">
<tns:brokerURL>tcp://server:portnumber</tns:brokerURL>
```

All SA Manager files contain the new MQ Server TCP port number.

3. Open the following files on all connector systems, change the **portnumber** value in the listed sections, and save the changes:
   – <SOI_HOME>\resources\Configurations\SSA_IFW_*servername*.xml

```
<AMQ name="AMQClient" uuid="">
      <ConnectionInfo failover="TRUE" host="server"
      password="(encrypted pw)" port="portnumber" protocol="tcp"
      reconnectAttempts="240" reconnectTime="30000" user="catalyst"/>
      . . .
</AMQ>
```

   – <SOI_HOME>\resources\eventManagerServerConfig.xml

```
<property name="port" value="portnumber"/>
```

   All connector files contain the new MQ Server TCP port number.
4. Start the CA SOI Manager service on the SA Manager system.
5. Run registryloader.bat located at SOI_HOME\tomcat\registry. Ignore any "log4j:WARN" messages.
   The Registry loads an updated record of all CA Catalyst files in which you changed the port number.
6. Restart the CA SOI Manager service, and start all other services on the SA Manager, UI Server, and all connector systems.
   The port number change takes effect.

Change the UI Server TCP Port Number

1. You can change the TCP port number for UI Server communication. The UI Server TCP port number is 7070 by default, and the UI Server uses this port number to provide access to the user interfaces.

   NOTE
   This procedure is only for changing the UI Server port number for HTTP communication.

   Follow these steps:
2. Stop all CA SOI services on the SA Manager and UI Server.
   a. Open the following files on the SA Manager system, change the **portnumber** value in the listed sections, and save the changes:
      • <SOI_HOME>\tomcat\webapps\sor.application\WEB-INF\web.xml

```
<context-param>
   <param-name>restURL</param-name>
   <param-value>http://localhost:portnumber</param-value>
</context-param>
```

      • <SOI_HOME>\tomcat\registry\topology\physical\node0\sor\solr.properties

```
solr.url=http://servername:portnumber/solr
```

3. All files on the SA Manager contain the new UI Server port number.
4. Select Start, Programs, CA, Service Operations Insight, right-click CA Service Operations Insight User Interface, and select Properties.
   The Shortcut Properties dialog opens.
5. Change the port number in the Target dialog and click OK.

The shortcut is redirected to use the new port number.

6. Open the following files on the UI Server, change the current UI Server port number in the following sections, and save the changes:
   - <SOI_Home>CA\SOI\jsw\conf\soi-user-interface.properties
     - Change TOMCAT_PORT_HTTP=<required portnumber>
   - <SOI_Home> CA\SOI\SamUI\conf\soi_conf\mobile.properties
     - Change headlessUrl=http://localhost:<portnumber>/rest

   All files on the UI Server contain the new UI Server port number.
7. Start the CA SOI Manager service on the SA Manager system.
8. Run registryloader.bat located at <SOI_HOME>\tomcat\registry. Ignore any "log4j:WARN" messages.
   The Registry loads an updated record of all CA Catalyst files in which you changed the port number.
9. Restart the CA SOI Manager service, and start all other services on the SA Manager and UI Server.
   The port number change takes effect.

Change the JDBC Port Number

You can change the JDBC port number for communicating with the SA Store database. The JDBC port number is 1433 by default.

Follow these steps:

1. Stop all CA SOI services on the SA Manager and UI Server.
2. Open the following files on the SA Manager system, change the **portnumber** value in the listed sections, and save the changes:
   - <SOI_HOME>\tomcat\lib\hibernate.cfg.xml

     ```
     <session-factory>
           <!-- Database connection settings -->
           . . .
           <property name="connection.url">jdbc:jtds:sqlserver:
           //server:portnumber/SAMStore;instance=;
           </property>
           . . .
     </session-factory>
     ```
   - The following files located at <SOI_HOME>\tomcat\registry\topology\node0\sor:
     a. restserver.xml
     b. sorapp.xml
     c. ssaserver.xml
     d. ssaweb.xml

     ```
     <tns:persistenceManager inj:injID="pm">
     . . .
           <tns:dataSource inj:injID="dbconfig">
           <tns:url>jdbc:jtds:sqlserver://server:portnumber/
           SAMStore;instance=;socketKeepAlive=true;</tns:url>
     ```
   - <SOI_HOME>\ServiceDiscovery\connectivityContext.xml

     ```
     <bean id="dbSettings" class="com.ca.ssa.servicediscovery.loader.DBSettings">
           <property name="driver" value="net.sourceforge.jtds.jdbc.Driver" />
           <property name="url" value="jdbc:jtds:sqlserver://server:portnumber/SAMStore" />
     ```

     NOTE
     Make this change in the '<!-- connectivity to the persistence store database -->' and the '<!-- connectivity to the SSA database -->' sections.
   - (Optional) <SOI_HOME>\tomcat\webapps\sam\thinuiconf\custom_metric_definition.xml

Change the port number in the CONNECTION_URL property for each custom metric. This is only required if you have added custom Dashboard metrics that communicate on the same port as the SA Store.

All SA Manager files contain the new JDBC port number.

3. Copy the <SOI_HOME>\tomcat\lib\hibernate.cfg.xml file on the SA Manager to <*tomcat_path*>\webapps\SpectrumSA on the report server, remove the <!DOCTYPE...> element at the top of the file, and save and close the file.

   NOTE
   tomcat_path refers to the default BusinessObjects Tomcat installation directory of C:\Program Files\CA\SC \CommonReporting3\Tomcat55.

4. Select Start, Programs, BusinessObjects XI 3.1, BusinessObjects Enterprise, Central Configuration Manager.
   The Central Configuration Manager dialog opens.
5. Select Apache Tomcat 5.5.20 and Server Intelligence Agent, click Stop, and then click Start when the services stop.
   The JDBC port number is changed on the report server.
6. (Optional) Open the <SOI_HOME>\SamUI\webapps\sam\thinuiconf\custom_metric_definition.xml file on the UI Server, change the port number in the CONNECTION_URL property for each custom metric, and save the changes.
   This step is only required if you have added custom Dashboard metrics that communicate on the same port as the SA Store.
7. Start the CA SOI Manager service on the SA Manager system.
8. Run registryloader.bat located at <SOI_HOME>\tomcat\registry. Ignore any "log4j:WARN" messages.
   The Registry loads an updated record of all CA Catalyst files in which you changed the port number.
9. Restart the CA SOI Manager service.
10. Change the port number in the listed sections of the following files on the SA Manager:
    - <SOI_HOME>\wso2registry\repository\conf\registry.xml

      ```
      <dbConfig name="wso2registry">
            <url>jdbc:jtds:sqlserver://server:portnumber/SAMStore;instance=
            </url>
      ```

    - <SOI_HOME>\wso2registry\repository\conf\user-mgt.xml

      ```
      <Configuration>
      . . .
            <Property name="url">jdbc:jtds:sqlserver
            ://server:portnumber/SAMStore;instance=</Property>
      ```

    The files used to load the CA Catalyst Registry contain the new port number.
11. Start all CA SOI services on the SA Manager and UI Server.
    The port number change takes effect.

Change CA EEM Connection Information

As an administrator, if the CA EEM administrator password has changed after installation, you can change the record of the eiamadmin user password in CA SOI if necessary. You can also change the CA EEM server name if you have moved your CA EEM installation to another system.

Follow these steps:

1. Access the CA SOI Dashboard, click the Administration tab, expand CA Service Operations Insight Manager Configuration and the SA Manager server, and click EEM Configuration.
2. Enter the new password in the Password field, enter the new CA EEM server name in the EEM Server Host field if necessary, and click Save.
3. Repeat Step 2 on the EEM Configuration page under CA Service Operations Insight UI Server Configuration.

   WARNING
   Change the CA EEM password on both the SA Manager and the UI Server.

4. Restart the CA SAM Application Manager and CA SAM User Interface services.

The changes are applied. If you changed the CA EEM server name, the previous users and groups may not exist on the new CA EEM server. Log in to the CA SOI interfaces using the administrator user to reconfigure users and groups, if necessary.

5. Open a command prompt, navigate to <SOI_HOME>\Tools\CatalystEncrypt, and run the following command:
   ```
   encrypter <new password>
   ```
   An encrypted password appears.
6. Open the SamUI\conf\jaas.config file, paste the encrypted password as the value for the eiam.query.password property, enter the new CA EEM server name as the value for the eiam.backend property if necessary, and save and close the file.
7. Open the <SOI_HOME>\tomcat\registry\topology\physical\node0\sor\eem.properties file, paste the encrypted password as the value for the eiam.query.password property, enter the new CA EEM server name as the value for the eiam.backend property if necessary, and save the changes.

   NOTE
   All CA Catalyst configuration files contain the new encrypted password and the new server name if necessary.
8. Update the password or hostname in the following files:
   - SOI\SamUI\webapps\sam\eem-config.xml
   - SOI\SamUI\webapps\sam\eiam.config
   - SOI\tomcat\webapps\sam\eem-config.xml
   - SOI\tomcat\webapps\sam\eiam.config
   - CA\SOI\tomcat\webapps\sam\WEB-INF\sso\config\sso-eem-config.xml
9. Run registryloader.bat located at <SOI_HOME>\tomcat\registry. Ignore any "log4j:WARN" messages.
   The Registry loads an updated record of all files in which you changed CA EEM connection information.
10. Restart the CA SAM Application Server service and CA SAM User Interface services.

Update Relationship Significance

As an administrator, you use the WSSamRelationshipCmd.bat utility to update the significance value of a relationship. The utility updates the significance value by the property name.

Follow these steps:

1. Open a command prompt, and navigate to <SOI_HOME>\tomcat\bin.
2. Run the following command:
   ```
   WSSamRelationshipCmd.bat -hsamanagerhost:port -uadminuser -ppassword -aUpdate -sServiceID -lLeftNodeID-
   iRightNodeID -npropertyName -vpropertyValue
   ```

   NOTE
   Note the lack of spaces between the parameters and values.

   - **-h*samanagerhost*:*port***
     Defines the host name and Tomcat port number of the SA Manager from which you want to update relationship significance. The default Tomcat port is 7090.
   - **-u*adminuser***
     Defines the user name of the CA SOI administrator user ("samuser" user by default).
   - **-p*password***
     Defines the password of the CA SOI administrator user. Run WSSamEncryptCmd.bat from the same directory to generate an encrypted version of the password for use in this parameter.
   - **-a*Update***
     Defines to update the significance of a relationship. Currently this operation is the only option.
   - **-s*ServiceID***

Defines the instanceID of the service in which the relationship to update exists. Find the instanceID property for any service in the Service Modeler.

– **-l*LeftNodeID***
  Defines the instanceID of the source node of the relationship.
– **-i*RightNodeID***
  Defines the instanceID of the target node of the relationship.
– **-n*propertyName***
  Defines the property name to update. Currently, the only valid value is sam_significance, which updates the significance value for a relationship.
– **-v*propertyValue***
  Defines the new value for the specified property. Enter a numeric in the range of 1 through 10 to specify a valid significance value.

The command runs. A return code of 1 indicates that the command failed. The update progress displays in the command prompt. After the update completes, a success message displays with a return code of 0.

Example: Update a server and application relationship significance

The following example command updates the significance of the relationship between the SA_Server:server1 CI and the SA_Application:payroll application:

```
WSSamRelationshipCmd.bat -hSAServer:7090 -usamuser -psamuserpw -aUpdate -sSA_Service:Payroll -
lSA_Server:server1 -rSA_Application:payroll -nsam_significance -v8
```

Update a CI Property

As an administrator, you use the WSSamCICmd.bat utility to update a subset of the service or CI properties. The utility updates one property at a time.

Follow these steps:

1. Open a command prompt, and navigate to <SOI_HOME>\tomcat\bin.
2. Run the following command:

```
WSSamCICmd.bat -hsamanagerhost:port -uadminuser -ppassword -aUpdate -iInstanceID -npropertyName -
vpropertyValue
```

NOTE
Note the lack of spaces between the parameters and values.

– **-h*samanagerhost*:*port***
  Defines the host name and Tomcat port number of the SA Manager from which you want to update CI properties. The default Tomcat port is 7090.
– **-u*adminuser***
  Defines the user name of the CA SOI administrator user ("samuser" user by default).
– **-p*password***
  Defines the password of the CA SOI administrator user. Run WSSamEncryptCmd.bat from the same directory to generate an encrypted version of the password for use in this parameter.
– **-a*Update***
  Defines to update the property of a CI. Currently this operation is the only option.
– **-i*InstanceID***
  Defines the instanceID of the CI to update. A CI can be a service or any CI that is part of an existing service. Find the instanceID property for any CI in the Service Modeler.
– **-n*propertyName***
  Defines the property name to update. Valid values are as follows:
  - **item_label**

Updates the Label property.

- **item_description**
  Updates the Description property.
- **item_priority**
  Updates the Priority property. Valid values are Low, Medium, High, and Critical.
- **item_status**
  Updates the Operational Mode property. Valid values are Active and In Repair.
- **item_department**
  Updates the Location property.
- **sam_category**
  Updates the Category property.
- **sam_ipaddress**
  Updates the IP Address property.

– ***-vpropertyValue***
  Defines the new value for the specified property. Enter a valid value.

The command runs. A return code of 1 indicates that the command failed. The update progress displays in the command prompt. After the update completes, a success message displays with a return code of 0.

Example: Update a Server CI Location

The following example command updates the SA_Server:server1 CI Location property:

```
WSSamCICmd.bat -hSAServer:7090 -usamuser -psamuserpw -aUpdate -iSA_Server:server1 -nitem_department -vOfficeB
```

WSS Command Usage

The following WSS commands are available in CA SOI. These batch files are located in **<SOI_Home>\tomcat\bin** folder.

> **NOTE**
> WSS commands work with **http** requests only, does not work when SSL is enabled.

WSSamCICmd

The WSSamCICmd.bat utility updates a subset of the service or CI properties. The utility updates one property at a time.

For more information about WSSamCICmd command, see Update a CI Property.

WSSamEncryptCmd

The WSSamEncryptCmd.bat utility allows you to encrypt a password.

Syntax: WSSamEncryptCmd.bat <text to encrypt>

For example, The following example encrypts the password "Sandeep".

C:\Program Files (x86)\CA\SOI\tomcat\bin>WSSamEncryptCmd.bat "sandeep"

Executing WSSamEncryptCmd.....

EEwaiG6S5B3flQhrqP/mDCYodUP4QLRXHdTZXlppPLOY

WSSamRelationshipCmd

The WSSamRelationshipCmd.bat utility updates the significance value of a relationship. The utility updates the significance value by the property name. For more information about WSSamRelationshipCmd command, see Update Relationship Significance.

WssamServiceCmd

The WSSamServiceCmd.bat utility allows you to import and export services, escalation policies, and escalation actions. You can also edit the XML file and export only Escalation Policies.

Syntax:

- -h<wsHostName:wsPort>
- -u<wsUsername>
- -p<encrypted wsPassword>
- -a<Export|Import>
- -s<Service InstanceID|*>
- -f<fileName>

WSSamServiceCmd -h<wsHostName:wsPort> -u<wsUsername> -p<encrypted wsPassword> -a<Export|Import> -s<Service InstanceID|*> -f<filename>

For example,

Exporting all the services: WSSamServiceCmd -hserver1:7090 -usamuser -p"EFbJeXR3zLsi9aPfOQ9FzRVOPPEUwCGxCWUFTNF4kxKD" -aExport -s* -fc:/allServicesExport.xml

Importing all the services: WSSamServiceCmd -hserver1:7090 -usamuser -p"EFbJeXR3zLsi9aPfOQ9FzRVOPPEUwCGxCWUFTNF4kxKD" -aImport -s* -fc:/allServicesImport.xml

WSSAServiceCmd

The WSSSAServiceCmd.abt utility allows you to import and export services from one CA SOI system to another CA SOI system. Before importing or exporting a service, generate an encrypted password by executing WSSamEncryptCmd command.

Syntax:

- -h<wsHostName:wsPort>
- -u<wsUsername>
- -p<encrypted wsPassword>
- -a<Export|Import>
- -s<Service InstanceID|*>
- -f<fileName>

WSSamServiceCmd -h<wsHostName:wsPort> -u<wsUsername> -p<encrypted wsPassword> -a<Export|Import> -s<Service InstanceID|*> -f<filename>

For example,

Exporting a particular service:

WSSSAServiceCmd -hlocalhost:7090 -usamuser -p"EFbJeXR3zLsi9aPfOQ9FzRVOPPEUwCGxCWUFTNF4kxKD" -aImport -s"Service:servicename,WSMan New Service" -fc:/myServiceImport.xml

Importing a particular service:
WSSSAServiceCmd -hlocalhost:7090 -usamuser -p"EFbJeXR3zLsi9aPfOQ9FzRVOPPEUwCGxCWUFTNF4kxKD" -aImport -s"Service:servicename,WSMan New Service" -fc:/myServiceImport.xml

CA Catalyst Operations

This section describes how to configure and customize the data processing operations provided by CA Catalyst: reconciliation and synchronization.

How to Perform CA Catalyst Reconciliation

As an administrator, you can configure CA SOI to perform CI reconciliation.

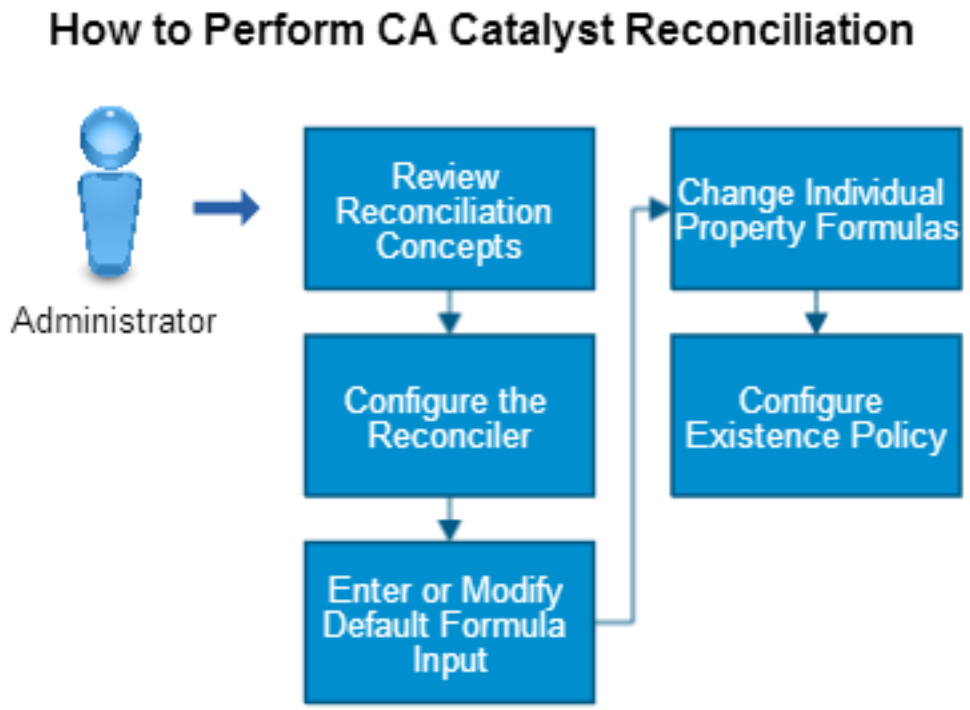
The Correlator and Reconciler components interact as follows to create a reconciled set of CIs:

1. The Correlator receives notification when the correlation keys match for two or more collected CIs.
2. The Correlator gathers the projection sheets from the matching CIs into a notebook and sends an event indicating a new or updated notebook to the Reconciler.
3. The Reconciler retrieves the notebook indicated in the event from the Persistence Service. The Reconciler then creates a reconciled sheet from the notebook. The reconciled sheet is a single set of properties that are derived from the projection sheets using reconciliation formulas.

CA SOI displays the reconciled sheet from each notebook as a CI in the USM Web View. You can view USM properties for each CI to see the reconciled set of properties. You can also view the USM notebook for each CI to see the reconciled sheet compared with the projection sheets listing the properties retrieved from each source domain manager.

Use this scenario to guide you through the process:

Figure 44: how to perform catalyst reconciliation



1. Review reconciliation concepts.
2. Configure the Reconciler.
3. Enter or modify the default formula input.
4. Change individual property formulas.
5. Configure existence policy.

If you experience any problems with reconciliation, see Reconciliation Errors.

Reconciliation Concepts

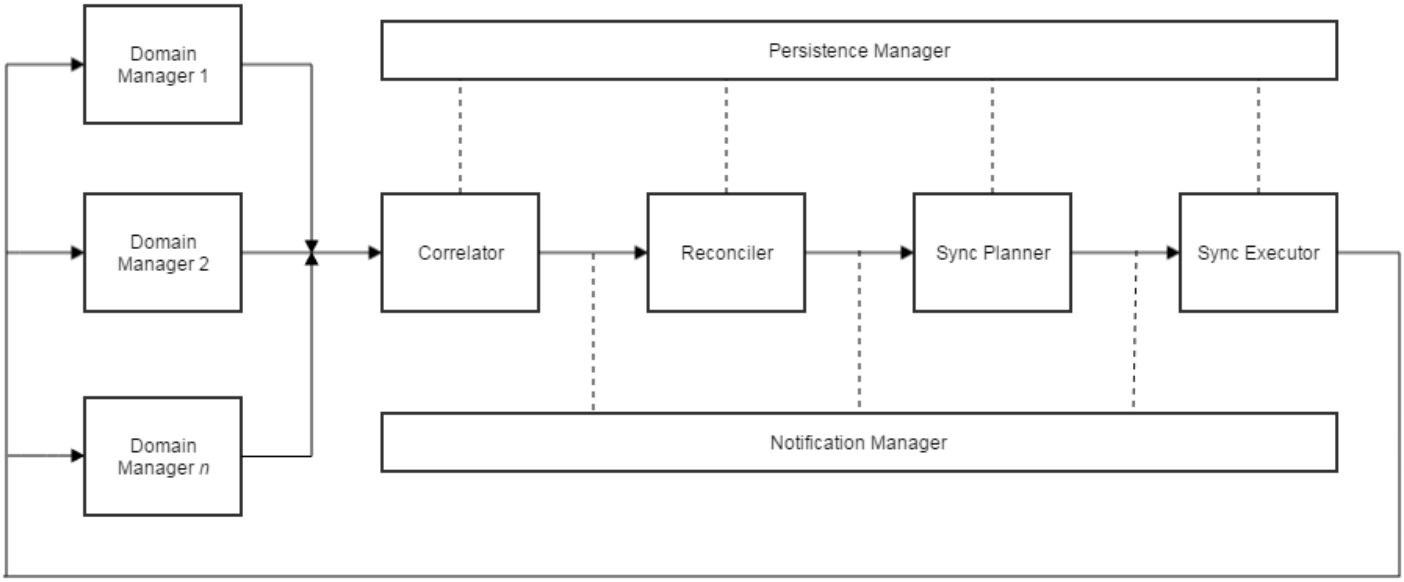
Contents

Become familiar with the following reconciliation concepts and how to access the CA Catalyst Registry.

Logic Server Overview

The Logic Server uses the USM schema and the Persistence Service interface to enact operations that create a unified, Persistent Store of USM data from the domain manager data retrieved by connectors. The following illustration shows the flow of data from connectors through the Logic Server:

Figure 45: Flow of data from connectors



- **Correlator**
  Matches USM entities coming from multiple domain managers that represent the same managed object (for example, a database server managed by CA Spectrum and CA eHealth). This SA Manager component evaluates data against common properties named *correlation keys*. Each CI enters the SA Manager as a *projection sheet*. If correlation keys match, the Correlator creates a *notebook* with all projection sheets that the Correlator determined refers to the same CI. Correlation occurs in the SA Manager and notebooks are transmitted to the Reconciler in the Logic Server.
- **Reconciler**
  Creates a *reconciled sheet* of common properties for correlated projection sheets, so that objects managed by multiple domains appear as one reconciled CI in CA SOI that uses the unified set of properties in the reconciled sheet.

  NOTE
  For more information about how reconciliation works and how to customize reconciliation policies and formulas, see How Reconciliation Works.
- **Persistence Manager**
  Transmits data to and from the Persistence Service for creating, updating, and deleting CI sheets and notebooks and running named queries on objects.
- **Notification Manager**
  Manages the subscription and buffering of events from the Persistence Service. This component notifies the Logic Server modules when USM data requires modification.
- **Sync Planner**

Determines when to synchronize data from the Persistent Store with source domain managers based on synchronization plans.

- **Sync Executor**
  Performs the synchronization operations indicated by the Sync Planner. The Executor pushes synchronization changes to the connector framework, after which the connector carries out the necessary inbound operations on its domain manager to change the domain manager data so that it matches the records in the Persistent Store.

NOTE
For more information about how synchronization works, see Synchronization.

WARNING
The functionality enabled by the Sync Planner and Sync Executor components is only supported for specific synchronization use cases.

Reconciliation Formulas

The Reconciler reconciles notebook CI properties that are based on reconciliation rules and formulas. The Reconciler contains a Formula Processor that determines the appropriate value of each property in a reconciled sheet that is based on the following provided formulas:

- **NullFormula**
  Sets a property value to Null in the reconciled sheet.
- **NoOpFormula**
  Ignores the calculation of any property value in the reconciled sheet. The old property value is preserved if it was set previously.
- **FirstNonNullValueWinsFormula**
  Uses the first not null value found in the projection sheets for the property in the reconciled sheet.
- **MajorityWinsFormula**
  Uses the property value that is reported by the most projection sheets as the value in the reconciled sheet.
- **SingleSourceOfTruthFormula**
  Populates the reconciled sheet based on a CI projection sheet from a specific domain manager as a source of truth. This formula requires an input to define the source of truth.
- **BinaryRelationshipFormula**
  Calculates the source MdrID and target MdrID in BinaryRelationship reconciled CIs. This formula uses only the following properties in the defaultsheet.xml file:
  - SourceMdrProduct
  - SourceMdrProdInstance
  - SourceMdrElementID
  - TargetMdrProduct
  - TargetMdrProdInstance
  - TargetMdrElementID
- **LastUpdatedWinsFormula**
  Uses property values from the last updated projection sheet to calculate the reconciled sheet.

Access the CA Catalyst Registry

You can access the CA Catalyst Registry to configure Logic Server functionality such as reconciliation rules, synchronization policies, and so on.

Follow these steps:

1. Click the Administration tab.
2. Click the plus sign (+) next to CA Service Operations Insight Manager Configuration.
3. Click the plus sign (+) next to the server you want to configure.

4. Click Catalyst Registry.
5. Enter the CA SOI administrator user credentials in the Username and Password fields and click Sign-in.

Configure the Reconciler

The Reconciler configuration is stored in the Registry. You can change the properties of several elements to influence how reconciliation occurs.

Follow these steps:

1. Access the CA Catalyst Registry.
2. Navigate to the following location: /topology/physical/node0/sor/ssaserver.xml
   A page opens for viewing or editing the contents.
3. Click Edit as text.
   The contents of the ssaserver.xml file display. The Reconciler configuration properties are located in the <tns:reconciler> section.
4. Make changes to any of the following key elements and click Save Content:
   - **numberOfCorrelationObservers**
     Defines the number of observers that can read the correlation queue for correlated events and process them. Increasing this number uses more memory and CPU to process events.
     Default: 1
   - **numberOfInstructionObservers**
     Defines the number of observers that can read the instruction queue for new instruction events and process them. Increasing this number uses more memory and CPU to process events.
     Default: 3
   - **mdrName**
     Defines the value of the MDR Name property in the reconciled sheet. The default is CA:00030, which is the registered MdrProduct property value for CA Catalyst.
   - **mdrInstanceName**
     Defines the value of the MDR Instance Name property in the reconciled sheet.
     Default: tenant0
   - **DefaultFormula**
     Defines the default formula used by the Reconciler to reconcile correlated CI projection sheets. You can set any available reconciliation formula as the default. The default formula is the global formula used by the Formula Processor to perform all reconciliations.
     Default: com.ca.ssa.sor.formula.LastUpdatedWinsFormula
     To change the default, set the ini:javaImpl property value next to "Default Formula" to the formula name prefixed by the implementation, which is com.ca.ssa.sor.formula for all provided formulas.
     For example, to set the default formula to NoOpFormula, modify the "Default Formula" line as follows:
     ```
     <tns:formula name="Default Formula"
     ini:javaImpl="com.ca.ssa.sor.formula.NoOpFormula"
     ```
     If the formula that you set as the default requires an input, you provide that input in the defaultsheet.xml file.
   - **defaultSheetURL**
     Defines the location of the file used to calculate the property values in the reconciled sheet.
     Default: /topology/physical/node0/sor/defaultsheet.xml
     You can manipulate the file listed in this location to specify different formulas for calculating the value of each property. Do not change this property unless you have changed the location of the defaultsheet.xml file.
   - **configurationURL**
     Defines the location of the file used to define existence policy, which declares the domain managers that should be the source of truth for determining CI existence in the Persistent Store.
     Default: /topology/physical/node0/sor/sourceoftruthmdr.xml

The Reconciler uses existence policy to detect if a CI managed by multiple domains should be deleted. You can manipulate this file to change the primary, secondary, and non-source of truth domain managers. Do not change this property unless you have changed the location of the sourceoftruthmdr.xml file.

Enter or Modify Default Formula Input

You specify the default reconciliation formula in the Registry.

- **MdrProduct**
  Defines the MdrProduct USM property value for the connector whose domain manager you want to be the single source of truth. All connectors have a registered MdrProduct value that is a five-digit number with a prefix of 'CA:'. The *Connector Guide* contains a table of all registered MdrProduct values.
- **MdrProductInstance**
  Defines the MdrProductInstance USM property value for the domain manager system to use as the single source of truth. This value is typically the server name associated with a specific instance of the domain manager.

  WARNING
  The input string for this property is case sensitive.
- **true|false**
  Specifies how the formula handles a null property value in the single source of truth CI projection sheet. Set this value to true to populate the reconciled sheet property with the null value. Set this value to false to use the FirstNonNullValueWinsFormula to populate the reconciled sheet property.

For the SingleSourceOfTruth formula, which is the only formula requiring an input, you must specify the domain manager in the defaultsheet.xml file and adhere to the following convention:

input="*MdrProduct*::*MdrProductInstance*,true|false"

As no other formulas require an input value, the default entry in the defaultsheet.xml file has the appropriate value of *.

Follow these steps:

1. Access the CA Catalyst Registry.
2. Navigate to the following location: /topology/physical/node0/sor/defaultsheet.xml.
3. Click Edit as text.
4. Change the input property in the line beginning with '<defaultFormula' to the appropriate value and click Save Content.
5. Run **registryloader.bat**. This bat file is located at *<SOI_Home>\tomcat\registry* folder.
6. Restart the **CA SAM Application Server** service.

Example: Set default input for CA Spectrum single source of truth formula

The following example sets a CA Spectrum instance as the single source of truth for the default reconciliation formula (which this example assumes is SingleSourceOfTruthFormula):

```
<defaultFormula name="DefaultFormula" input="CA:00002::cadev2,true" />
```

- **CA:00002**
  Sets data retrieved from the CA Spectrum connector as the single source of truth.
- **cadev2**
  Sets the specific CA Spectrum instance cadev2 as the single source of truth.
- **true**
  Populates any null property value from the single source of truth projection sheet with the null value in the reconciled sheet.

Change Individual Property Formulas

You specify the default reconciliation formula in the ssaserver.xml file. The defaultsheet.xml file lets you set different reconciliation formulas for specific properties. Property-specific formulas override the default for that property.

For example, consider a situation where CA Spectrum is the preferred domain manager for maintenance information. If the default formula is FirstNonNullValueWinsFormula, you may want to set the formula for the IsInMaintenance property to always use the property retrieved from CA Spectrum.

Follow these steps:

1. Access the CA Catalyst Registry.
2. Navigate to the following location: tomcat\registry\topology\physical\node0\sor\defaultsheet.xml.
3. Click Edit as text.
   Notice that several properties already have specific formulas applied, such as MdrProduct and MdrElementID. Do not change the formulas for these properties.
4. Use the following line to define a property-specific formula and click Save Content:
   ```
   <formulaRecord input="" name="" cellName=""/>
   ```
   - **cellName**
     Defines the name of the USM property to which to apply the specific reconciliation formula.
   - **name**
     Defines the name of the reconciliation formula to apply when reconciling the specified property. This attribute requires only the formula name, such as NoOpFormula.
   - **input**
     Defines input parameters for any formula that requires them. Currently, only SingleSourceOfTruthFormula requires input. For all other formulas, enter an asterisk (*) for the input.

Example: Set cell-specific formula for IpAddress

This example sets the reconciliation formula for the IpAddress property to NoOpFormula, so that the Reconciler does not perform any calculations on this property:

```
<formulaRecord input="*" name="NoOpFormula" cellName="IpAddress"/>
```

Configure Existence Policy

Existence policy controls the domain managers that the Reconciler uses as a source of truth to detect whether to delete a CI from the Persistent Store that has been deleted from a source domain manager. By default, CA Catalyst never deletes a reconciled CI from the Persistent Store. You can edit existence policy in the sourceoftruthmdr.xml file to specify a primary and secondary source of truth and a non-source of truth to use for this calculation.

For example, consider a CI that is managed in CA Spectrum and CA CMDB. If CA CMDB is set as the primary source of truth in existence policy and the CI is deleted from CA CMDB, the Reconciler deletes the CI from the Persistent Store, despite its continued existence in CA Spectrum.

Follow these steps:

1. Access the CA Catalyst Registry.
2. Navigate to the following location: /topology/physical/node0/sor/sourceoftruthmdr.xml.
3. Click Edit as text.
4. Enter entries for the following sections as necessary and click Save Content:
   - **<tns:PrimaryST>**
     Defines the primary source of truth for CI existence. If a CI managed by this connector is deleted in the source domain manager, CA Catalyst deletes it from the Persistent Store.
   - **<tns:SecondaryST>**

Defines the secondary source of truth for CI existence. If a CI managed by this connector is deleted in the source domain manager, CA Catalyst deletes it from the Persistent Store unless it still exists in any primary source of truth.

– **<tns:NonST>**
  Defines the non-source of truth for CI existence. If a CI managed by this connector is deleted in the source domain manager, CA Catalyst deletes it from the Persistent Store only if it does not exist in any domain managers listed as primary or secondary sources of truth.

You can define multiple sources of truth in each section, as long as specific instances appear in only one of the sections. When sections contain multiple sources of truth, a connector needs to match only one source of truth to trigger the appropriate action.

Use the following format to define a source of truth:

```
<tns:MdrProduct>MdrProduct</tns:MdrProduct>
<tns:MdrProductInstance>MdrProductInstance</tns:MdrProductInstance>
```

– **MdrProduct**
  Defines the MdrProduct USM property value for the connector whose domain manager to use as a source of truth. All connectors have a registered MdrProduct value that is a five-digit number with a prefix of 'CA:'. For more information, see Connector Identification Numbers.

– **MdrProductInstance**
  Defines the MdrProductInstance USM property value for the domain manager system to use as a source of truth. This value is typically the server name associated with a specific instance of the domain manager defined in the MdrProduct property.

WARNING
The input string for this property is case sensitive.

5. Restart the CA SAM Application Server service.

Example: Multi-domain existence policy

The following example illustrates an existence policy with multiple domains and instances defined in each section:

```
<tns:SourceOfTruthMdr xmlns:tns="http://ns.ca.com/catalyst/2010/02/sourceoftruthmdr"
xmlns:usm-core="http://ns.ca.com/2009/07/usm-core">
        <tns:PrimaryST>
                <tns:MdrID>
                        <tns:MdrProduct>CA:00020</tns:MdrProduct>
                        <tns:MdrProdInstance>cmdbserver</tns:MdrProdInstance>
                </tns:MdrID>
                <tns:MdrID>
                        <tns:MdrProduct>CA:00020</tns:MdrProduct>
                        <tns:MdrProdInstance>cmdbserver2</tns:MdrProdInstance>
                </tns:MdrID>
        </tns:PrimaryST>
        <tns:SecondaryST>
                <tns:MdrID>
                        <tns:MdrProduct>CA:00003</tns:MdrProduct>
                        <tns:MdrProdInstance>nsmserver</tns:MdrProdInstance>
                </tns:MdrID>
                <tns:MdrID>
                        <tns:MdrProduct>CA:00004</tns:MdrProduct>
                        <tns:MdrProdInstance>specserver</tns:MdrProdInstance>
                </tns:MdrID>
        </tns:SecondaryST>
        <tns:NonST>
                <tns:MdrID>
                        <tns:MdrProduct>CA:00031</tns:MdrProduct>
```

```
                <tns:MdrProdInstance>scomserver</tns:MdrProdInstance>
            </tns:MdrID>
            <tns:MdrID>
                <tns:MdrProduct>CA:00002</tns:MdrProduct>
                <tns:MdrProdInstance>5ehserver/tns:MdrProdInstance>
            </tns:MdrID>
        </tns:NonST>
</tns:SourceOfTruthMdr>
```

Based on this example, the Reconciler determines reconciled CI existence in the Persistence Store as follows:

- Deletes the reconciled CI when the projection CI is deleted in the CA CMDB servers cmdbserver and cmdbserver2.
- Deletes the reconciled CI when the projection CI is deleted in the specserver CA Spectrum instance if the CI is not reported from either of the CA CMDB servers listed in the primary source of truth section.
- Deletes the reconciled CI when the projection CI is deleted from the scomserver Microsoft SCOM instance or ehserver CA eHealth instance if the CI is not reported from any of the instances listed as primary and secondary sources of truth.

Synchronization

Contents

> **WARNING**
> Synchronization functionality is not enabled by default for this release and is only supported for specific use cases. This section only includes information about enabling those specific use cases.

Synchronization Use Cases

The CA Catalyst framework and bidirectional connectors provide the functionality to invoke inbound create, update, and delete operations to synchronize the data in integrated domain managers. This functionality is only supported when used as part of a certified use case.

CA SOI supports synchronization for the following use cases:

Alert Synchronization

CA SOI supports synchronization of Cleared and Acknowledged alert properties for CA Spectrum and Microsoft SCOM.

CA SOI Service Synchronization

CA SOI supports synchronization of all CI types and relationships for BMC Atrium or CA CMDB.

Maintenance Mode Synchronization

If you experience any problems with synchronization, see Synchronization Errors.

CA SOI supports synchronization of maintenance mode status with connectors that support inbound to connector update operations on the IsInMaintenance USM property.

> **WARNING**
> Patches to Registry files may require you to re-enable previously enabled synchronization use cases after patch application. See the notes included with each patch for more information.

How to Enable Alert Synchronization

Contents

As an administrator, you can synchronize the Cleared and Acknowledged alert properties from CA SOI to the following products and releases:

- CA Spectrum (Acknowledged and Cleared properties)
- Microsoft SCOM (Cleared and Acknowledged properties)

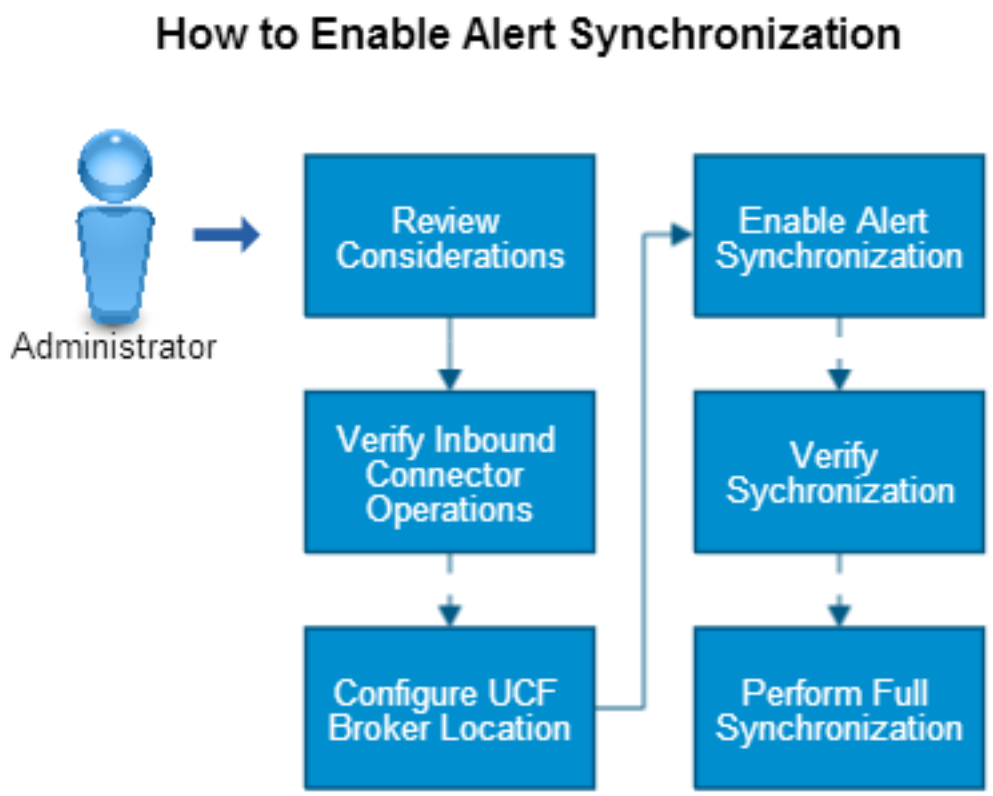
The domain manager sends an alert to CA SOI; once a user clears or acknowledges the alert in CA SOI, the update propagates to the domain manager.

Unlike other synchronizations where CIs are correlated across all domain managers, CA SOI alerts are correlated in pairs with the domain manager that initiated the alert (CA SOI and CA Spectrum or CA SOI and Microsoft SCOM).

All reconciliation formulas are supported, but only SingleSourceOfTruth and LastUpdateWinsFormula (the default formula) are certified.

Use this scenario to guide you through the process:

Figure 46: how to enable alert synchronisation



1. Review the considerations.
2. Verify that inbound to connector operations are enabled on all connectors that you want to participate in the use case.
3. (Optional) Configure the UCF Broker location if it uses a different server or port number than the default.
4. Enable alert synchronization and configure reconciliation formulas.
5. (Optional) Verify the synchronization.
6. (Optional) Perform a full synchronization.
   If you experience any problems with synchronization, see Synchronization Errors.

Considerations

Enable Inbound to Connector Operations in the Connector

By default, inbound to connector operations are enabled in all connectors. Verify that operations are enabled only in the connectors for which you want to make changes in their domain managers.

> **NOTE**
> If you are performing maintenance mode synchronization, verify that each connector supports update operations on the IsInMaintenance property. Then you can verify inbound to connector operations. See the *Connector Guide* for each connector for a list of supported operations.

Follow these steps:

1. Access the CA SOI Dashboard.
2. Click the Administration tab.
3. Expand Connector Configuration and the connector server name and select a connector entry.
   Settings for that connector display.
4. Verify that IsRemotable is selected in the Connector Controls table, select the check box if it is not, and click Save.
5. Repeat Steps 3-4 for all connectors that you want to participate in a supported use case. You can also disable inbound to connector operations for connectors that you do not want to participate in a supported use case.

Configure UCF Broker Location

The Synchronizer uses the UCF Broker to communicate synchronization changes to connectors, so it must have the correct UCF Broker URI. By default, the UCF Broker URI is set to localhost listening on port 8020. If either condition is true, you configure the UCF Broker location:

- You installed the UCF Broker on a separate system from the SA Manager.
- You set the UCF Broker to listen on a nondefault port.

Follow these steps:

1. Access the CA Catalyst Registry.
2. Navigate to the following location: /topology/physical/node0/sor/ucf-broker.properties.
   A page opens for viewing or editing the file contents.
3. Click Edit as text.
4. Change the following line to use the correct UCF Broker system and listener port and click Save
   Content:broker.1.ws.uri=http://localhost:8020/ucf/BrokerService
5. Restart the CA SAM Application Server service.

Enable Alert Synchronization

The CA SOI Dashboard provides an Administration page that lets you enable alert synchronization that the use case supports.

Follow these steps:

1. Click the Administration tab.
2. Click the plus sign (+) next to CA SOI Insight Manager Configuration.
3. Click the plus sign (+) next to the server you want to configure.
4. Click Synchronization Configuration.
5. Select the Enable option for Alerts.
6. Click Save.
   This control enables alert synchronization in all domain managers that the use case supports with connectors that have the IsRemotable control enabled.

7. (Optional) To change the reconciliation formula, see Working with CA Catalyst Reconciliation.
   The default reconciliation formula is LastUpdateWinsFormula. You can change the formula to define a single source of truth or define rules for specific CI properties.
8. Click Submit.
9. Restart the CA SAM Application Server service.

Verify Alert Synchronization

After you have enabled the alert synchronization use case, clear or acknowledge an alert in the Operations Console. Then verify that the change occurs in the reconciled sheet and the supported domain manager that contains the alert.

Perform Full Synchronization

Enabling synchronization use cases only synchronizes changes from the time they are enabled forward. CA SOI provides a synchronization priming utility that fully synchronizes CA SOI to the real-time environment.

This utility is useful in the following situations:

- You just enabled synchronization, and you want to synchronize changes already in place.
- You experienced a product failure (CA SOI, a connector, the Synchronizer, and so on) and you want to synchronize any changes that occurred during the downtime.
- You added a product to your solution and you want to completely synchronize with the products already installed.

Due to the possible increased load on your network that was generated during a full synchronization, we recommend performing the synchronization during non-business hours.

> **WARNING**
> Enable synchronization use cases before you run the priming utility, or full synchronization does not occur. Run the priming utility for use cases you have enabled only. You can enable any or all of the supported use cases simultaneously.

Follow these steps:

1. Locate the sync_primer.properties file at <SOI_HOME>\Tools\PrimingUtility, and open with a text editor.
2. Update the following values as necessary:
   - **sleep_time_secs**
     Defines the sleep time in seconds between sets of CI notebooks that are sent to the reconciler.
     Default: 5
   - **notebook_chunk_size**
     Defines the maximum number of notebooks to send to the Reconciler in a single set.
     Default: 100
     Limit: 1000
   - **Alert_Synch**
     Defines if alert synchronization is enabled (true) or disabled (false).
   - **Alert_Synch_Type**
     Defines the CI types to synchronize.

     > **NOTE**
     > Currently only the Alert type is supported.
   - **InMaintenance**
     Defines if maintenance mode synchronization is enabled (true) or disabled (false).
   - **InMaintenance_Types**
     Defines the CI types to synchronize separated by commas. By default, the full list of CI types is provided, so you can remove CI types you do not want to synchronize if necessary.
   - **SOI_Service_Synchronization**

Defines if the CA SOI service synchronization is enabled (true) or disabled (false).

- **SOI_Service_Synchronization_Types**
  Defines the list of supported CI types separated by commas, or enter All_Types to synchronize all CI types, depending on the policies on the machine.
- start_date=*yyyy-mm-dd hh*:*mm*:*ss*
  Defines the date in the past to begin the CI synchronization. If left blank, all CIs (regardless of date) are retrieved and synchronized. The format is *yyyy-mm-dd hh*:*mm*:*ss* wherehour (*hh*) is based on a 24-hour clock; for example, *hh* for 3PM is 15.
  Example: Enter 2010-11-12 14:21:00 to synchronize all CIs since November 12, 2010 at 2:21:00 PM.
- **SOI_MDR_PRODUCT_ID=CA:00047**
  For future use. Do not change the value of this property.

3. Save the file.
4. Run PrimerUtility.bat in the <SOI_HOME>\Tools\PrimingUtility directory on the server where the SA Manager is installed.
   The utility indicates the number of notebook IDs retrieved. The utility provides its progress by displaying the current number of CIs remaining to be reconciled and when it is sleeping (based on the value you entered for sleep_time_secs).

How to Enable CA SOI Service Synchronization

Contents

As an administrator, you can synchronization all CI types and relationships from CA SOI to the following products and releases:

- CA CMDB
- BMC Atrium

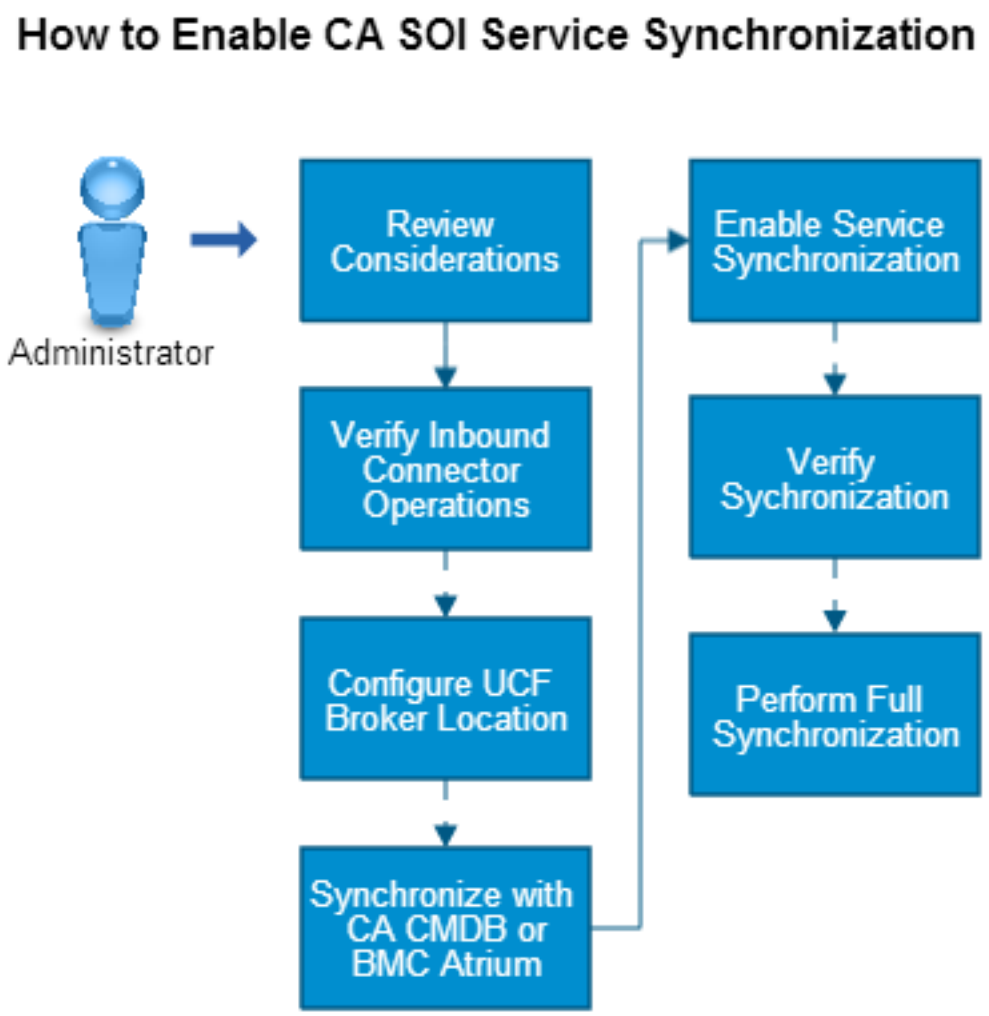
When a change to a service model (including adding or updating a CI) occurs, connectors supporting the use case invoke the change in their domain manager. Therefore, the services are synchronized across your enterprise.

> **NOTE**
> When enabling service synchronization, only relationships can be deleted.

The default reconciliation formula is LastUpdateWinsFormula.

Use this scenario to guide you through the process:

Figure 47: enable service synchronisation



1. Review the considerations.
2. Verify that inbound to connector operations are enabled on all connectors that you want to participate in the use case.
3. (Optional) Configure the UCF Broker location if it uses a different server or port number than the default.
4. (Optional) Synchronize CA SOI with all service models in CA CMDB or BMC Atrium.
5. Enable CA SOI service synchronization and configure reconciliation formulas.
6. (Optional) Verify the synchronization.
7. (Optional) Perform a full synchronization.

If you experience any problems with synchronization, see Synchronization Errors.

Considerations

When using CA SOI service synchronization, consider the following items:

- Due to processing requirements, perform an initial synchronization during non-business hours.
- All CI types and relationships are supported, but CA SOI must manage the CIs and therefore part of the managed service.
- You can synchronize either CA CMDB or BMC Atrium, but not both.
- Only one instance of either product is supported; for example, you cannot have three instances of CA CMDB.
- Relationships are correlated when CA SOI manages them.
- The use case supports all CI types and relationships. However, the CA CMDB and BMC Atrium connectors support inbound to connector operations on only a subset of CI types. For a list of supported types for each connector, see the CA CMDB and BMC Atrium connector documentation. Also, see the InboundToConnectorTypes list on the connector's page in the Administration UI.

  NOTE
  BinaryRelationship is not included in the InboundToConnectorTypes list for BMC Atrium, but the connector does support inbound to connector operations on the BinaryRelationship type.

To correctly synchronize relationship deletions in CA SOI with CA CMDB or BMC Atrium, import all synchronized services from CA CMDB or BMC Atrium back into CA SOI. The import includes services managed in other products that are integrated with CA SOI and created in CA CMDB or BMC Atrium as a result of enabling synchronization. If you want to synchronize all services managed in CA CMDB or BMC Atrium, including those created when synchronization occurs, enable automatic service import from the appropriate connector before enabling the synchronization. However, if you want to synchronize only a subset of managed services in CA CMDB or BMC Atrium, manually import that subset of services back into CA SOI after enabling synchronization.

Enable Inbound to Connector Operations in the Connector

By default, inbound to connector operations are enabled in all connectors. Verify that operations are enabled only in the connectors for which you want to make changes in their domain managers.

NOTE
If you are performing maintenance mode synchronization, verify that each connector supports update operations on the IsInMaintenance property. Then you can verify inbound to connector operations. See the *Connector Guide* for each connector for a list of supported operations.

Follow these steps:

1. Access the CA SOI Dashboard.
2. Click the Administration tab.
3. Expand Connector Configuration and the connector server name and select a connector entry.
   Settings for that connector display.
4. Verify that IsRemotable is selected in the Connector Controls table, select the check box if it is not, and click Save.
5. Repeat Steps 3-4 for all connectors that you want to participate in a supported use case. You can also disable inbound to connector operations for connectors that you do not want to participate in a supported use case.

Configure UCF Broker Location

Synchronize with CA CMDB or BMC Atrium

You can synchronize CA SOI with all service models in CA CMDB or BMC Atrium, including service models that are created during a synchronization.

Follow these steps:

1. Select the appropriate connector on the Configure Data Sources dialog in the Operations Console.
2. Click Auto to enable the automatic service import for that connector. If you select this option, then do *not* perform a full synchronization.

NOTE
If you want to synchronize only a certain subset of services in CA CMDB or BMC Atrium with CA SOI, then manually import those services from CA CMDB or BMC Atrium back into CA SOI. Do not click Auto for the import.

Enable CA SOI Service Synchronization

The CA SOI Dashboard provides an Administration page that lets you enable service synchronization that the use case supports.

Follow these steps:

1. Click the Administration tab.
2. Click the plus sign (+) next to CA SOI Insight Manager Configuration.
3. Click the plus sign (+) next to the server you want to configure.
4. Click Synchronization Configuration.
5. Select the Enable option for Services.
6. Click Save.
   The changes are applied. This control enables service synchronization in all domain managers that the use case supports with connectors that have the isRemotable control enabled.
7. (Optional) To change the reconciliation formula, see the Working with CA Catalyst Reconciliation.
   The default reconciliation formula is LastUpdateWinsFormula. You can change the formula to define a single source of truth or define rules for specific CI properties.
8. Click Submit.
9. Restart the CA SAM Application Server service.

Verify CA SOI Service Synchronization

After you have enabled the service synchronization use case, perform one of the following actions to verify the functionality:

- (LastUpdatedWinsFormula) Import a service into CA SOI from any source (like CA eHealth or CA Spectrum), and verify that the service is created in BMC Atrium or CA CMDB.
- (LastUpdatedWinsFormula) Add CIs to a service in CA CMDB and BMC Atrium and verify that the CIs are created in the CA SOI service.
- (SingleSourceOfTruthFormula) Change a CI property in a synchronized service in the single source of truth domain manager. Also, verify that the change occurs in all participating products.

Perform Full Synchronization

Enabling synchronization use cases only synchronizes changes from the time they are enabled forward. CA SOI provides a synchronization priming utility that fully synchronizes CA SOI to the real-time environment.

This utility is useful in the following situations:

- You just enabled synchronization, and you want to synchronize changes already in place.
- You experienced a product failure (CA SOI, a connector, the Synchronizer, and so on) and you want to synchronize any changes that occurred during the downtime.
- You added a product to your solution and you want to completely synchronize with the products already installed.

Due to the possible increased load on your network that was generated during a full synchronization, we recommend performing the synchronization during non-business hours.

WARNING
Enable synchronization use cases before you run the priming utility, or full synchronization does not occur. Run the priming utility for use cases you have enabled only. You can enable any or all of the supported use cases simultaneously.

Follow these steps:

1. Locate the sync_primer.properties file at <SOI_HOME>\Tools\PrimingUtility, and open with a text editor.
2. Update the following values as necessary:
   - **sleep_time_secs**
     Defines the sleep time in seconds between sets of CI notebooks that are sent to the reconciler.
     Default: 5
   - **notebook_chunk_size**
     Defines the maximum number of notebooks to send to the Reconciler in a single set.
     Default: 100
     Limit: 1000
   - **Alert_Synch**
     Defines if alert synchronization is enabled (true) or disabled (false).
   - **Alert_Synch_Type**
     Defines the CI types to synchronize.

     NOTE
     Currently only the Alert type is supported.
   - **InMaintenance**
     Defines if maintenance mode synchronization is enabled (true) or disabled (false).
   - **InMaintenance_Types**
     Defines the CI types to synchronize separated by commas. By default, the full list of CI types is provided, so you can remove CI types you do not want to synchronize if necessary.
   - **SOI_Service_Synchronization**
     Defines if the CA SOI service synchronization is enabled (true) or disabled (false).
   - **SOI_Service_Synchronization_Types**
     Defines the list of supported CI types separated by commas, or enter All_Types to synchronize all CI types, depending on the policies on the machine.
   - start_date=*yyyy-mm-dd hh*:*mm*:*ss*
     Defines the date in the past to begin the CI synchronization. If left blank, all CIs (regardless of date) are retrieved and synchronized. The format is *yyyy-mm-dd hh*:*mm*:*ss* wherehour (*hh*) is based on a 24-hour clock; for example, *hh* for 3PM is 15.
     Example: Enter 2010-11-12 14:21:00 to synchronize all CIs since November 12, 2010 at 2:21:00 PM.
   - **SOI_MDR_PRODUCT_ID=CA:00047**
     For future use. Do not change the value of this property.
3. Save the file.
4. Run PrimerUtility.bat in the <SOI_HOME>\Tools\PrimingUtility directory on the server where the SA Manager is installed.
   The utility indicates the number of notebook IDs retrieved. The utility provides its progress by displaying the current number of CIs remaining to be reconciled and when it is sleeping (based on the value you entered for sleep_time_secs).

How to Enable Maintenance Mode Synchronization

Contents

As an administrator, you can synchronize the maintenance mode status with connectors that support inbound to connector update operations on the IsInMaintenance USM property. The following connectors support this use case:

- Microsoft SCOM
- CA Spectrum
- CA CMDB
- CA SOI

> **NOTE**
> See the *CA Catalyst Connector Guide* that is provided with each connector to verify other connectors that support the use case.

A maintenance mode update for a CI in a domain manager changes the CA Catalyst CI reconciled sheet according to the reconciliation formula. After this change occurs, connectors supporting the use case invoke the change in their domain managers, so that maintenance status is synchronized across your enterprise.

The use case also affects the maintenance mode in CA SOI. You can configure CA SOI so that IsInMaintenance property updates in the source domain manager change the CA SOI maintenance mode setting for the CI. For more information about maintenance mode in CA SOI, see How to Schedule Maintenance for Services and Resources.

Although most reconciliation formulas work with the maintenance mode synchronization use case, the most common formulas to apply to the use case are as follows:

- **SingleSourceOfTruthFormula**
  Updates the CA SOI maintenance mode status and the IsInMaintenance property in the CA Catalyst reconciled sheet and other domain managers. Updates are based on the setting in one source of truth domain manager. For example, if you define CA CMDB as the single source of truth, and a CI managed in CA CMDB goes into maintenance, the update occurs in CA SOI and all other domain managers with connectors that support the use case. An update to the maintenance status in other domain managers is ignored and reverted to the single source of truth value.
- **LastUpdatedWinsFormula**
  Updates the CA SOI maintenance mode status and the IsInMaintenance property in the reconciled sheet and other domain managers. Updates are based on the last domain manager that updated the property. For example, if you put a CI into maintenance in CA SOI, the update occurs across domains. This formula synchronizes every maintenance mode update.

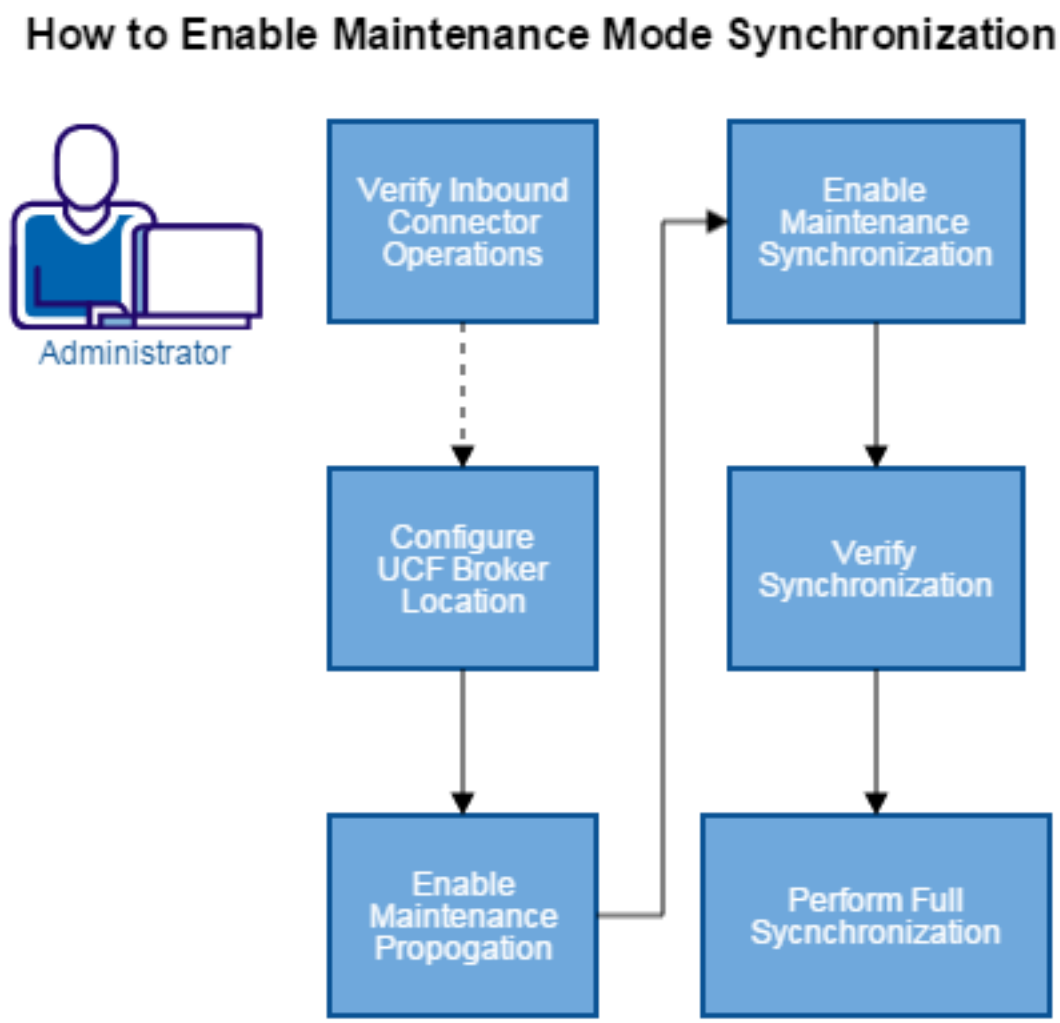
The use case supports only updating the property, not creating or deleting. The specific USM property for which updates are synchronized is IsInMaintenance. The following USM types include this property:

Application	MailServer
ApplicationServer	ManagementAgent
ApplicationSystem	Memory
BackgroundProcess	Network
Cluster	OperatingSystem
ComputerSystem	OrganizationalEntity
Database	Port
DatabaseInstance	Printer
DirectoryServer	Processor
File	Router
GenericIPDevice	Service
Group	Switch
HypervisorManager	Tablespace
InterfaceCard	VirtualSystem

NOTE
The list includes types that at least one participating connector supports. Not all participating connectors support all of these types.

Use this scenario to guide you through the process:

Figure 48: Enable Maintenance Mode Synchronization



WARNING
This release supports only the synchronization customizations described in Step 4. Do not perform additional customization to synchronization settings or the synchronization plan.

1. Verify that inbound to connector operations are enabled on all connectors that you want to participate in the use case.

   NOTE
   See the *CA Catalyst Connector Guide* for each connector to verify whether it supports updating IsInMaintenance in its domain manager.
2. (Optional) Configure the UCF Broker location if it uses a different server or port number than the default.
3. Enable CA SOI maintenance propagation.
4. Enable maintenance synchronization and configure reconciliation formulas.

5. (Optional) Verify the synchronization.
6. (Optional) Perform a full synchronization.

If you experience any problems with synchronization, see Synchronization Errors.

Enable Inbound to Connector Operations in the Connector

By default, inbound to connector operations are enabled in all connectors. Verify that operations are enabled only in the connectors for which you want to make changes in their domain managers.

> **NOTE**
> If you are performing maintenance mode synchronization, verify that each connector supports update operations on the IsInMaintenance property. Then you can verify inbound to connector operations. See the *Connector Guide* for each connector for a list of supported operations.

Follow these steps:

1. Access the CA SOI Dashboard.
2. Click the Administration tab.
3. Expand Connector Configuration and the connector server name and select a connector entry.
   Settings for that connector display.
4. Verify that IsRemotable is selected in the Connector Controls table, select the check box if it is not, and click Save.
5. Repeat Steps 3-4 for all connectors that you want to participate in a supported use case. You can also disable inbound to connector operations for connectors that you do not want to participate in a supported use case.

Configure UCF Broker Location

The Synchronizer uses the UCF Broker to communicate synchronization changes to connectors, so it must have the correct UCF Broker URI. By default, the UCF Broker URI is set to localhost listening on port 8020. If either condition is true, you configure the UCF Broker location:

- You installed the UCF Broker on a separate system from the SA Manager.
- You set the UCF Broker to listen on a nondefault port.

Follow these steps:

1. Access the CA Catalyst Registry.
2. Navigate to the following location: /topology/physical/node0/sor/ucf-broker.properties.
   A page opens for viewing or editing the file contents.
3. Click Edit as text.
4. Change the following line to use the correct UCF Broker system and listener port and click Save
   Content:broker.1.ws.uri=http://localhost:8020/ucf/BrokerService
5. Restart the CA SAM Application Server service.

Enable Maintenance Propagation

You can configure a CA SOI setting so that the maintenance mode status can participate in maintenance synchronization. When enabled, the CA SOI maintenance status can play any role in the synchronization use case. The status can be the single source of truth, or it can implement a maintenance status change from other domains. If you do not enable CA SOI to participate in maintenance synchronization, any maintenance status that you define in CA SOI CIs is unaffected by maintenance synchronization in other connectors and their domain managers.

Follow these steps:

1. Access the CA SOI Dashboard
2. Click the Administration tab.
3. Expand CA Service Operations Insight Manager Configuration and the server that you want to configure.

4. Click Global Settings.
5. Select Yes from the Propagate Domain Manager Maintenance Settings drop-down list and click Save.

Enable Maintenance Synchronization

The CA SOI Dashboard provides an Administration page that lets you enable maintenance mode synchronization that a use case supports.

Follow these steps:

1. Click the Administration tab.
2. Click the plus sign (+) next to CA SOI Insight Manager Configuration.
3. Click the plus sign (+) next to the server you want to configure.
4. Click Synchronization Configuration.
5. Select enabled in the IsInMaintenance Property Synchronization pane.
   This control enables maintenance mode synchronization in all domain managers that a use case supports. The connectors must have the IsRemotable control enabled.
6. Click Add Formula in the IsInMaintenance Property Reconciliation Formula Record pane.
   A drop-down list appears for selecting the reconciliation formula for the IsInMaintenance property. Note that this formula only affects reconciliation on the IsInMaintenance property, not other properties.
7. Select the appropriate reconciliation formula.
   If you select Single Source of Truth, the MDR drop-down list appears.
8. (Single Source of Truth formula only) Do the following:
   – Select the domain manager to use as the single source of truth in the MDR drop-down list. If you do not see an entry that should appear in the MDR drop-down list, verify that the UCF Broker is running and that all connectors are running with the IsRemotable control enabled. You can define multiple sources of truth by adding another formula. The first single source of truth takes precedence over other selections.
   – Select the Accept null value check box to accept any IsInMaintenance value provided by the single source of truth, even a null value, as the reconciled property value to synchronize in all participating domains. If you leave this check box cleared, a null value in the single source of truth causes the Synchronizer to evaluate the next formula for the reconciled IsInMaintenance value.
9. (Optional) Complete Steps 6-8 to add as many reconciliation formulas as necessary.
   The Synchronizer evaluates the formulas in the order they appear on the page, with the first formula taking precedence. You can have multiple sources of truth or multiple formulas to calculate the reconciled value if the first formula does not apply (the source of truth returns a null value, for example).
10. Click Submit.
11. Restart the CA SAM Application Server service.

Verify Maintenance Synchronization

After you have enabled the maintenance synchronization use case, perform one of the following actions to verify the functionality:

(SingleSourceOfTruthFormula) Change the maintenance mode for a CI in the single source of truth domain manager, and verify that the change occurs in the reconciled sheet and other domain managers with connectors that the use case supports.

- (LastUpdatedWinsFormula) Change the maintenance mode for a CI in CA SOI, and verify that the change occurs in all domain managers with connectors that the use case supports.
- (LastUpdatedWinsFormula) Change the maintenance mode for a CI in any domain manager, and verify that the change occurs in CA SOI and other domain managers with connectors that the use case supports.

Perform Full Synchronization

Enabling synchronization use cases only synchronizes changes from the time they are enabled forward. CA SOI provides a synchronization priming utility that fully synchronizes CA SOI to the real-time environment.

This utility is useful in the following situations:

- You just enabled synchronization, and you want to synchronize changes already in place.
- You experienced a product failure (CA SOI, a connector, the Synchronizer, and so on) and you want to synchronize any changes that occurred during the downtime.
- You added a product to your solution and you want to completely synchronize with the products already installed.

Due to the possible increased load on your network that was generated during a full synchronization, we recommend performing the synchronization during non-business hours.

> **WARNING**
> Enable synchronization use cases before you run the priming utility, or full synchronization does not occur. Run the priming utility for use cases you have enabled only. You can enable any or all of the supported use cases simultaneously.

Follow these steps:

1. Locate the sync_primer.properties file at <SOI_HOME>\Tools\PrimingUtility, and open with a text editor.
2. Update the following values as necessary:
   - **sleep_time_secs**
     Defines the sleep time in seconds between sets of CI notebooks that are sent to the reconciler.
     Default: 5
   - **notebook_chunk_size**
     Defines the maximum number of notebooks to send to the Reconciler in a single set.
     Default: 100
     Limit: 1000
   - **Alert_Synch**
     Defines if alert synchronization is enabled (true) or disabled (false).
   - **Alert_Synch_Type**
     Defines the CI types to synchronize.

     > **NOTE**
     > Currently only the Alert type is supported.
   - **InMaintenance**
     Defines if maintenance mode synchronization is enabled (true) or disabled (false).
   - **InMaintenance_Types**
     Defines the CI types to synchronize separated by commas. By default, the full list of CI types is provided, so you can remove CI types you do not want to synchronize if necessary.
   - **SOI_Service_Synchronization**
     Defines if the CA SOI service synchronization is enabled (true) or disabled (false).
   - **SOI_Service_Synchronization_Types**
     Defines the list of supported CI types separated by commas, or enter All_Types to synchronize all CI types, depending on the policies on the machine.
   - start_date=*yyyy-mm-dd hh*:*mm*:*ss*
     Defines the date in the past to begin the CI synchronization. If left blank, all CIs (regardless of date) are retrieved and synchronized. The format is *yyyy-mm-dd hh*:*mm*:*ss* wherehour (*hh*) is based on a 24-hour clock; for example, *hh* for 3PM is 15.
     Example: Enter 2010-11-12 14:21:00 to synchronize all CIs since November 12, 2010 at 2:21:00 PM.
   - **SOI_MDR_PRODUCT_ID=CA:00047**
     For future use. Do not change the value of this property.

3. Save the file.
4. Run PrimerUtility.bat in the <SOI_HOME>\Tools\PrimingUtility directory on the server where the SA Manager is installed.
   The utility indicates the number of notebook IDs retrieved. The utility provides its progress by displaying the current number of CIs remaining to be reconciled and when it is sleeping (based on the value you entered for sleep_time_secs).

Database Maintenance

This section describes how to maintain your SA Store database, including rebuilding, moving, and purging the database.

Rebuild the SA Store Database

As an administrator, you can rebuild the SA Store database to recreate the database while purging all data. The SAMStoreRebuild.bat utility drops the existing database and recreates the tables, indexes, keys, and meta data using the files from <SOI_HOME>\SamStore.

Follow these steps:

1. Stop all CA SOI services.
2. Double-click SAMStoreRebuild.bat located at <SOI_HOME>\SamStore on the SA Manager.
3. Enter the required database server information:
   - If you are using the default instance, the InstanceName field is optional.
   - Enter the sa database credentials in the UserID and Password fields.
   - The default SA Store database name is SAMStore.
4. Restart all CA SOI services.

Move the SA Store to a Remote Database Server

As an administrator, you can move the SA Store database to a different database server as long as the new database server allows the same Microsoft SQL Server user name and password for database access. After moving the database, you edit various configuration files to help ensure that product components communicate with the database at its new location.

Follow these steps:

1. Stop all CA SOI services.
2. Perform a full backup of the SA Store database and restore it to the remote Microsoft SQL Server.
3. Open the <SOI_HOME>\ServiceDiscovery\connectivityContext.xml files on the SA Manager, change the database server name in the ssaDB.url property, and save the file.
4. Change the database server name in the following SA Manager files and save them:
   <SOI_HOME>\tomcat\lib\hibernate.cfg.xml
   <SOI_HOME>\tomcat\registry\topology\physical\node0\sor\restserver.xml
   <SOI_HOME>\tomcat\registry\topology\physical\node0\sor\sorapp.xml
   <SOI_HOME>\tomcat\registry\topology\physical\node0\sor\ssaserver.xml
   <SOI_HOME>\tomcat\registry\topology\physical\node0\sor\ssaweb.xml
   <SOI_HOME>\wso2registry\repository\conf\registry.xml
   <SOI_HOME>\wso2registry\repository\conf\user-mgmt.xml

   NOTE
   For the changes in the regitsry files to take effect, execute the registryLoader.bat file in <SOI_HOME>\tomcat\registry.
5. Start the CA SOI services.

The changes are applied.

6. Locate the <Tomcat_HOME>\webapps\SpectrumSA\hibernate.cfg.xml file on the report server, change the database server name in the connection.url property, and save the file.
7. Change the database server name in the following files and save them:
   <SOI_HOME>\wso2registry\repository\conf\registry.xml
   <SOI_HOME>\wso2registry\repository\conf\user-mgmt.xml
8. Select Start, Programs, BusinessObjects XI 3.x, BusinessObjects Enterprise, Central Configuration Manager.
   The Central Configuration Manager dialog opens.
9. Select Apache Tomcat and Server Intelligence Agent, click Stop, and then click Start when the services stop.
   The database server name is changed on the report server.
10. Open the <SOI_HOME>\Reports\samreports.xml file on the report server, update the database server name, and save the file.
    Future report redeployments will use the new database connection password.
11. Change the database server name in the following files and save them:
    <SOI_HOME>\Reports\samreports.xml
    <SOI_HOME>\Reports\deploy\hibernate.cfg.xml
12. (Optional) Do the following to reenable custom prompt pages:

    NOTE
    This step is only necessary if you have previously enabled custom report prompt pages.

    a. Select Start, Programs, BusinessObjects XI 3.x, BusinessObjects Enterprise, 32-bit Data Source (ODBC).
       The ODBC Data Source Administrator dialog opens.
    b. Select the System DSN tab, select SAMStore, update the database server name, and apply the change.
       The ODBC connection uses the correct database server name.
    c. Select Start, Programs, BusinessObjects XI 3.x, BusinessObjects Enterprise, Central Configuration Manager.
       The Central Configuration Manager dialog opens.
    d. Select Apache Tomcat and Server Intelligence Agent, click Stop, and then click Start when the services stop.
    e. Enter the following URL in a web browser:
       ```
       http://localhost:<BOServer_port>/SpectrumSA/CustomSSA.jsp
       ```
       The Spectrum SA Report Initialization page opens.
    f. Enter the correct information and click Update Reports.
       This sets the CA SOI report type to SI_CA_REPORT_TYPE. A confirmation page opens and custom parameter pages now display when you run CA SOI reports if the operation was successful.

    All product components can now communicate with the moved SA Store database.

CA SOI Toolbox Utility

Contents

The toolbox is a command line utility which allows you to start and stop specific connectors and services, enabling efficient cleanups of your CA SOI instance. Use the CA SOI toolbox application to manage the maintenance of data in your CA SOI SA Store database. The users with Windows administrator and non-administrator privileges can execute the commands.

To run SOIToolBox for Windows non-administrator users, the following permissions are required:

- Read and Execute permissions on the soitoolbox.exe file.
- Full Control permissions for all CA SOI Services.

The toolbox lets you clear unwanted data from your database. Except for the reinitializeDB, setAutoImportFlag and cleanImportedData commands, you can execute all the database cleanup commands with or without stopping the services. However, if the data to be processed is more, you may experience performance issues.

The CA SOI toolbox is on the SA Manager in the SOI_HOME\Tools folder. Navigate to this location on a command prompt and run the following command with the options/parameters described in the ensuing sections:

```
soitoolbox
```

NOTE

For CA SOI distributed system, enter the CA SOI setup details in the **soitoolbox.cfg** file (the .cfg file created by using -x command, if it does not exist) before running any CA SOI toolbox commands

NOTE
The command help also contains information about the functions of the options and commands that are described in this section. For practical examples of using the utility, see How to Clean Up Data with the CA SOI Toolbox.

Configuration Options

This section describes the different commands for the CA SOI toolbox that you can use for CA SOI instance. In all commands include the necessary options to establish a valid connection with the SA Store database. If you do not, the utility prompts you for the necessary information.

- **-m, --machine**
  Specifies the system to connect to which runs the SA Store database.
  Default: localhost
- **-n, --mUsername**
  Specifies the name of the user who has access to the database system.

  NOTE
  This command is not valid and has no effect when using the localhost system.

  Default: Administrator
- **-w, --mPassword**
  Specifies the password of the Windows users in plain text.

  NOTE
  This command is not valid and has no effect when using the localhost system.
- **-d, --dbName**
  Specifies the name of the CA SOI database.
  Default: SAMStore
- **--dbArchiveName**
  Allows you to specify a new name for the archive database.
- **-u, --dbUsername**
  Specifies the user with privileges to access the CA SOI database.
  Default: sa
- **-p, --dbPassword**
  Specifies the password for the database user in plain text.

  NOTE
  If the password is not specified, the utility requests it during runtime.
- **--credentials**
  Specifies the location of the file containing the user names and passwords for all systems that run CA SOI components. This command is required when you operate a distributed CA SOI instance.
- **-x**
  Creates a configuration file template if the file does not exist.

NOTE
This command is the equivalent of *--credentials=soitoolbox.cfg*.

- **-q, --quiet**
  Specifies that the user is not asked to confirm the running of a destructive operation.
- **-t, --timeout**
  Specifies the generic timeout in seconds.
  Default: 30
- **-b, --dbConnectionTimeout**
  Specifies the timeout in seconds for the initial wait time for the opening of the database connection.
  Default: 30
- **-c, --connector**
  Specifies the connectors to be used in action commands.
  Default: *

  NOTE
  *, in this case, means all connectors in a CA SOI instance. The following examples show how the * symbol can be used:

```
--connector=CA:09998_soimachine.ca.com, CA:00005_soimachine.ca.com
--connector=CA:09998_*
--connector=CA:*_soimachine.ca.com
--connector=*
```

- **-s, --beSmart**
  Activates a special method for stopping connectors. In beSmart mode, the utility only stops components (services and connectors) which affect by an action command.

  NOTE
  This option requires that you include *--credentials.*

Service Related Action Commands

You can use the following action commands to stop, start, and restart the services in a CA SOI instance. You can also use the utility to get information about the status of your services.

- **--startConnector**
  Starts the connectors which you specify in the *--connector* parameter.

  NOTE
  Provide the *--credentials* parameter to use this command.

  WARNING
  If the CA SOI UI Server is not running, the command fails.
- **--stopConnector**
  Stops the connectors which you specify in the *--connector* parameter.

  NOTE
  Provide the --credentials parameter to use this command.

  WARNING
  If the CA SOI UI Server is not running, the command fails.
- **--getConnectorStatus**
  Provides the status of all the connectors in a CA SOI instance, including system connectors.
- **--getAvailableConnectors**
  Lists the names of all the connectors present in a CA SOI instance, excluding system connectors.
- **--getStatisticalData**

Displays statistical data from the CA SOI database. This data includes information about the number of CIs, Alerts, and Services.

NOTE

The *--connector* option influences the output of this command. Confirm that the connector name is in the *<MdrProduct>_<MdrProdInstance>@<Hostname>* format.

- **--setAutoImportFlag**Turns off auto import flag for a particular connector.
  Syntax: --setAutoImportFlag=<value> --connector=<name>
  value = 0/1 or on/off; name = connector name
  Examples: --setAutoImportFlag=1 --connector=CA:09998_* --setAutoImportFlag=off --connector=CA:09998_*

  NOTE

  This command can be changed or removed without prior notice.

The following table describes the behavior of commands for Windows administrator users and non-administrator users:

Service Command and Description	Administrator Users	Non-Administrator Users
--stopAllServices Stops all the Windows services in a CA SOI instance	The user can stop the CA SOI service status on both local and remote machine.	The user can stop the CA SOI Services only on a local machine. **Access is denied** error appears for the CA SOI Services on remote machines.
--startAllServices Starts all the Windows services in a CA SOI instance.	The user can start the CA SOI service status on both local and remote machine.	The user can start the CA SOI Services only on a local machine. **Access is denied** error appears for the CA SOI Services on remote machines.
--restartAllServices Restarts all the Windows services in a CA SOI instance by stopping and starting them.	The user can restart the CA SOI service status on both local and remote machine.	The user can restart the CA SOI Services only on a local machine. **Access is denied** error appears for the CA SOI Services on remote machines.
--getServiceStatus Displays the status of all the Windows services in a CA SOI instance.	The user can get the CA SOI service status on both local and remote machine.	The user is prompted for remote machine login password while executing this command.

Database Commands

The database commands include cleaning the data, archiving and restoring the data, re-initializing the database, and rebuilding the indexes in a CA SOI database. Except for reinitializeDB, setAutoImportFlag, and cleanImportedData commands, you can execute all the cleanup commands with or without stopping the services. The database cleanup commands work only if the database is intact. If you notice any error while running the cleanup commands, rebuild the SA Store database.

The list of database commands are as follows:

- **--archiveHistoryData**Moves all historical data older than a specified number of days to an archive database.

  NOTE

  You specify the number of days by adding a number to the end of the command, for example --archiveHistoryData=10.Note: The default name for the archive database isSOIArchiveDB. You can change the name of the archive database with the --dbArchiveName command.
- **--cleanHistoryData** Deletes all the historical data from the CA SOI database that is older than a specified number of days.

NOTE

You specify the number of days by adding a number to the end of the command which specifies the number of days.

- **--cleanImportedData**
  Deletes all the data that came from connectors that you specify in the *--connector* parameter.

  NOTE
  Stop CA SOI services before executing this command. This command does not delete data that users create manually.
- **--cleanSecurityData**
  Deletes all service access rights that are set for specific users and groups within the CA SOI instance.
- **--cleanData**Deletes all history data, imported data, and security data for all the connectors in a CA SOI instance. This command is a single script that cleans all the data. This command is equivalent to the following commands:
  - --cleanHistoryData
  - --purgeClearedAlerts
  - --purgeOutageAlerts
  - --cleanSecurityData
  - --purgeAlertQueueAssignment
  - --cleanAlertsFromRemovedConnectors

  NOTE

  You do not have to stop the CA Services to delete data.
- **--purgeClearedAlerts**
  Deletes cleared alerts from the database that are older than a specified number of days. You can specify two different purge types, :*full*or :s*trict*, by adding *full* or *strict* to the end of the command:
  - **:full**
    Deletes all alerts that are cleared in at least one system.
  - **:strict**
    Deletes only those alerts which are cleared in all systems.
- **--rebuildIndexes**
  Rebuilds database indexes.
- **--reinitializeDB**
  Clears the whole CA SOI database, deleting the data from all the tables.

  NOTE
  Stop CA SOI services before executing this command. You can use this command to start with a clean state similar to that of a new installation. If you notice any error while running the command, rebuild the SA Store database.
- **--restoreHistoryData**
  Restores from the archive database all archived historical data that are not older than the specified number of days.

  NOTE
  You specify the number of days by adding a number to the end of the command, for example --restoreHistoryData=10.
- **--getRowCount**
  Generates .csv file that has all information of the CA SOI database tables.
- **--cleanAlertsFromRemovedConnectors**
  Removes the alerts of the removed connectors.
- --purgeAlertQueueAssignment
  Purges alert assigned to queue which does not exist in CA SOI.
- **--purgeOutageAlerts**

Purges alert that is generated during outages.

- **--findDBInconsistencies**Finds the inconsistencies, such as Orphan CIs, Orphan Candidates, and Catalyst Inconsistencies that are available in the database. Use the -v option to get detailed information about the consistency. Different inconsistency types are as follows:
  - Orphan Candidates
    CIs retained in CA SOI (due to the bug) after they are removed from the connector.
  - Orphan CIs
    CIs that do not have CI staging record from any Connector except Reconciler.
    (this excludes manually, in the Modeler, created services, and groups)
  - Catalyst Inconsistencies
    CIs/Relationships which do not exist in CA SOI database (in theCIstaging table), but are available in the Catalyst database (in the ca_ssa_ci_detail table).
- **--purgeDBInconsistencies** Deletes all orphan CIs and Catalyst inconsistencies that are found by executing --findDBInconsistencies.
- **--getHistoryDataCount** Displays the number of rows in the tables which hold historical data. Information about the main and archive CA SOI databases are displayed.
  Type the number of days to specify the time period as you do in --archiveHistoryData or --restoreHistoryData.

  NOTE

  - The following commands can be changed or removed without prior notice:
  - --purgeDBInconsistencies
  - --getHistoryDataCount
  - --findDBInconsistencies

- **--findDatabaseAnamolies**This command finds database anomalies for Service Discovery policy, Unicode Character, Cyclic relationship and Data from removed connectors.

The following table describes the behavior of commands for Windows administrator users and non-administrator users:

Database Commands	Administrator Users	Non-Administrator Users
• --cleanHistoryData • --cleanSecurityData • --purgeClearedAlerts • --archiveHistoryData • --restoreHistoryData • --purgeOutageAlerts • --purgeAlertQueueAssignment • --purgeDBInconsistencies • --cleanAlertsFromRemovedConnectors	Commands execute successfully.	• The user is prompted to enter the login password of remote machines. • If SOI services are running, the following are two scenarios for the commands to execute. – If a user chooses to stop CA SOI Services, the command execution fails and **Access is Denied** error appears. – If a user chooses not to stop CA SOI Services, the command executes successfully.
• --cleanImportedData • --reinitializeDB • --setAutoImportFlag	Commands execute successfully.	• Stop the CA SOI services manually before executing these commands. **Access is denied** error appears when executing the command without stopping the CA SOI services. • The user is prompted to enter the login password of remote machines given in the soitoolbox.cfg file.

• --rebuildIndexes • --encryptPassword • --findDBInconsistencies • --getHistoryDataCount	Commands execute successfully.	Commands execute successfully.
--getRowCount	No action is required from the user.	To execute this command, the user must have create file permissions on the Tools folder.

How to Clean Up Data with the CA SOI Toolbox

Contents

As an administrator, you maintain the CA SOI database. You ensure that your database runs efficiently and that the data it maintains is the data you need. Using the CA SOI Toolbox command utility, you can clean up the following data types:

- Imported data from a specific connector
- History and security data
- All the data that is maintained in a CA SOI instance

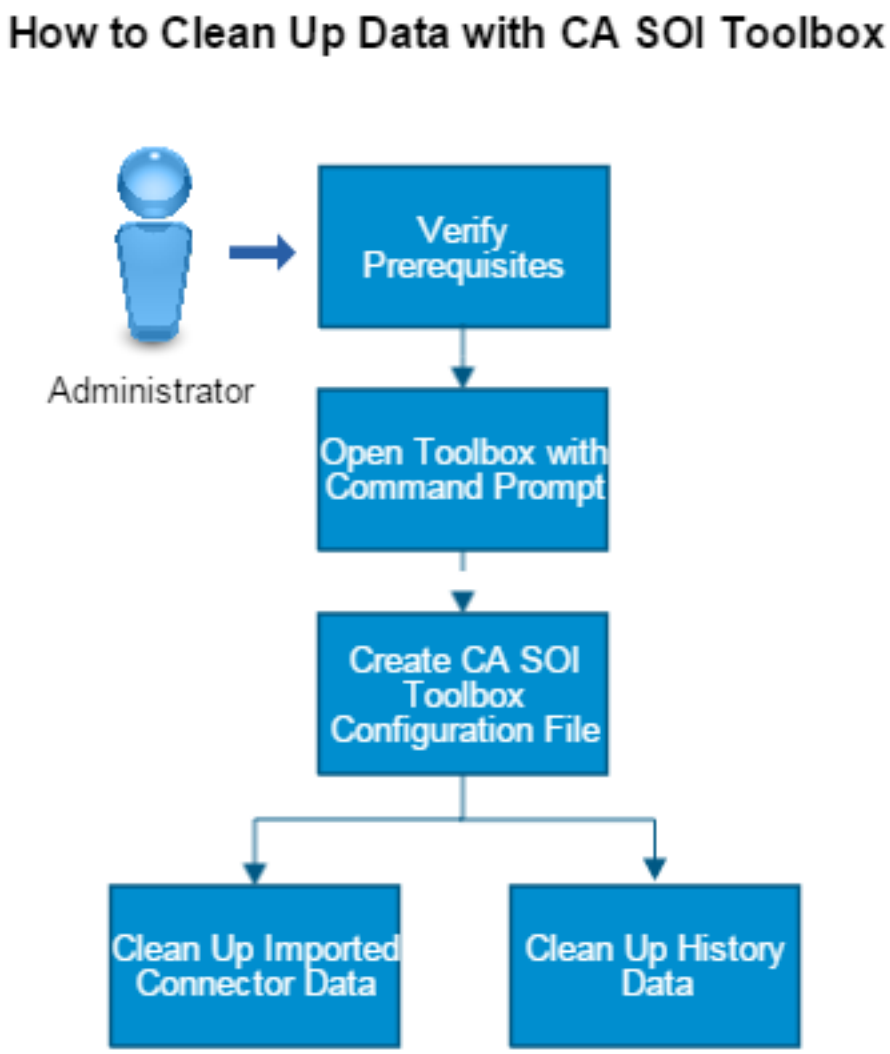
You may want to clean up data for the following reasons:

- You want to clean up data from a connector in your CA SOI instance that was not written correctly. Thus, the connector provides flawed data (for example, CIs using incorrect naming conventions) that does not correlate with data provided from other connectors.
- You want to maintain a more efficient CA SOI instance that does not maintain history that you do not need and consumes needed disk space. Thus, you clean up the history data that is no longer of use.
- Clear all security data for all users and user groups in a CA SOI instance.
- You want to restart a CA SOI instance from a clean state and clear all its data. This option is useful during the product implementation phase when incorrect connector configurations can cause database *pollution*.

> **NOTE**
> This scenario provides examples of commands for removing connector and history data. For detailed definitions of every parameter available, see the command help that comes with the toolbox.

Use this scenario to guide you through the process:

Figure 49: Clean Up Data with CA SOI Toolbox



1. Verify the prerequisites.
2. Determine the database sizing.
3. Open the toolbox with a command prompt.
4. (Optional) Create a CA SOI Toolbox configuration file.
5. Do any of the following:
   – Clean up the history data.
   – Clean up imported connector data.

CA SOI Toolbox Prerequisites

To use the CA SOI toolbox, you must have the following information:

- The server credentials for the CA SOI instance
- The database location and credentials for the CA SOI instance if the database location is on a remote system

NOTE
If you run an instance all on one server locally, you do not need credentials for the database.

- The name of the connector, If you want to perform a connector-specific cleanup

System Requirement:

- The CA SOI Toolbox requires that your system has the Microsoft Visual C++ 2008 Redistributable Package (x86) installed.

Determine Database Sizing

Maintaining the SA Store database is required to sustain consistent performance and maintain product function. Use the data provided in this section that is measured against the disk space and performance capabilities of your database server. You can then determine how often to perform maintenance.

The following tables display estimates of how you can expect the database to grow over time. They project growth over a three-year period for several implementation sizes.

NOTE
All tables assume that you have defined an SLA for every managed service.

Small (100 services, 1000 alerts per day)

Item	Amount	Average Row Size (KB)	Number of Rows	Disk Space Requirement (MB)
Managed CIs	20,000	.85	60,000	51
Staged CIs	100,000	1.345	300,000	403.5
Alerts	1,095,000	.6	3,285,000	1,971
Alert History		1.062	2,190,000	2,325.78
SLA	100	.3	109,500	32.85
History tables		.65	5,256,000	3,416
Total table size				8,201
Database log file				4,920
Total size				13,121

Mid-sized (500 services, 5000 alerts per day)

Item	Amount	Average Row Size (KB)	Number of Rows	Disk Space Requirement (MB)
Managed CIs	50,000	.85	150,000	127.5
Staged CIs	500,000	1.345	1,500,000	2,018
Alerts	5,475,000	.6	16,425,000	9,855
Alert History		1.062	10,950,000	11,629
SLA	500	.3	547,500	164.25
History tables		.65	26,280,000	17,082
Total table size				40,875
Database log file				24,525
Total size				65,400

Large (1,000 services, 10,000 alerts per day)

Item	Amount	Average Row Size (KB)	Number of Rows	Disk Space Requirement (MB)
Managed CIs	100,000	.85	300,000	255
Staged CIs	1,000,000	1.345	3,000,000	4,035
Alerts	10,950,000	.6	32,850,000	19,710
Alert History		1.062	21,900,000	23,258
SLA	1,000	.3	1,095,000	328.5
History tables		.65	52,560,000	34,164
Total table size				81,750
Database log file				49,050
Total size				130,800

- The Amount column lists the actual amount of each entity in CA SOI at the end of the three-year period. The Number of Rows column is larger than the actual amount in most cases.
- The SA Store tables not represented in this table are small enough that they have a negligible impact on database size.
- The History table row calculation is based on the fact that the three history tables (DBAvailHistory, DBQualityHistory, and DBRiskHistory) create a row every thirty minutes per service. Therefore, the history tables generate 48 rows per day per service.
- Plug your numbers into the table (total number of managed and staged CIs, average alerts received per day, and number of SLAs defined) that most closely approximates your implementation size to calculate the projected database growth rate for your implementation.
- The database log file is approximately 60 percent of the total database size.

Based on this data, you can make the following assumptions:

- At the end of a three-year period, the total disk space consumed by the SA Store database would be between 10,000-100,000 MB or 10-100 GB depending on the number of managed services, CIs, and alerts.
- You can expect a growth rate of roughly 2.7 GB per 1000 services per month, or 1.35 GB per 500 services per month, or 270 MB per 100 services per month.

Consider these estimates when planning the frequency of maintenance to help ensure that the SA Store does not grow to an unmanageable size. As a best practice, the row count for all tables should be limited so that SQL queries on any table can complete within a few seconds. For example, if you find that this limit is around 10,000,000 rows per table on your database server, you would have to at least archive and purge the history tables periodically depending on implementation size to keep them from growing beyond this threshold.

Open the Toolbox with a Command Prompt

To use the toolbox to run its commands, open the toolbox with a command prompt.

Follow these steps:

1. Navigate to the CA SOI toolbox on the SA Manager in the <SOI_Home>\Tools folder.
2. Run the following command from the command prompt:

   ```
   soitoolbox
   ```

   The toolbox file opens in the command window and lists all the commands of the toolbox in the command help.

   NOTE
   The following steps of this scenario provide two examples of different ways you can use the toolbox to clean up data.

Create a CA SOI Toolbox Configuration File

Clean Up History Data

When history data is no longer useful and you do not want to store in your database, you can clean up the history data. Maintenance of history data is important, because the history data can increase over time and can consume memory, thus adversely affecting the operation of the database. You can avoid buildup of unwanted data by cleaning up history data at regular intervals. With this command, you clean data which is older than a specified number of days. To clear unwanted history data in a database, run a cleanup command from the toolbox folder on a command prompt.

Follow these steps:

1. Open the toolbox with a command prompt.
2. Run the history cleanup command and specify the database connection password. For example, if your database connection password is *yourpw*, and you want to delete history data that is older than 2 days, you would run:
   ```
   soitoolbox -p yourpw --cleanHistoryData=2
   ```
   – **-p**
     Specifies the password. In this example, the password is *yourpw.*
   – **--cleanHistoryData**
     Runs the command for deleting history data.
   – **2**
     Specifies that only history data that is older than two days is deleted.

   The toolbox confirms that it found the services, connects to the database, and warns that you selected a destructive operation.

   WARNING
   The example command assumes that your CA SOI instance runs entirely on a local system. If you are running a CA SOI instance that is not located entirely in one location, create a toolbox configuration file. For information about creating the toolbox configuration file, see Create a CA SOI Toolbox Configuration File.
3. When prompted, confirm the destructive operation by typing Y for Yes.
   The toolbox verifies that CA SOI is running, and then asks you to stop the services before proceeding.
4. Confirm that you want to stop the services by typing Y for Yes.
   The toolbox stops the services and then deletes all the history data in the database.
5. Verify that the imported connector data was deleted by running the *--getStatisticalData* command.
   The toolbox returns information about the statistical data for the connector, confirming that the history data was deleted.

Clean Up Imported Connector Data

When a connector provides data that you do not want to store in your database, you can clean up data from the connector. To clean up imported connector data, run a cleanup command from the toolbox folder on a command prompt.

Follow these steps:

1. Open the toolbox with a command prompt.
2. Run the cleanup command with the information about the connector name location and the location of the database where the connector data is stored.
   For example, if you wanted to delete data from the connector *Example_con* stored in a database that resides on the server *server1* with the database connection password *yourpw*, you would run:
   ```
   soitoolbox -p yourpw -m server1 --cleanImportedData --connector=Example_con
   ```
   The toolbox confirms that it found the services, connects to the database, and warns that you selected a destructive operation.

WARNING
The example command assumes that your CA SOI instance runs entirely on a single server, *server1* in this case. If you are running a CA SOI instance that is not located entirely in one location, create a toolbox configuration file. For information about creating a toolbox configuration file, see Create a CA SOI Toolbox Configuration File.

NOTE
If you do not know your connector name, run the *--getAvailableConnectors* command first.

3. When prompted, confirm the destructive operation by typing Y for Yes.
   The toolbox verifies that SOI is running, and then asks you to stop the services before proceeding.
4. Confirm that you want to stop the services by typing Y for Yes.
   The toolbox stops the services and then deletes the imported connector data in the database.
5. Verify that the imported connector data was deleted by specifying the connector name with the *--connector* option and running the *--getStatisticalData* command.
   The toolbox returns information about the statistical data for the connector, confirming that the data was deleted.

   NOTE
   If you are cleaning up data from a connector that is providing incorrect data, fix the connector problems before restarting its services.

Task Scheduler for SOIToolbox Utility

You can use Windows Task Scheduler to run SOIToolbox utility commands.

Follow these steps:

1. Click Control Panel, Administrative Tools, Task Scheduler.
2. Click Create Task in the Actions panel.
3. Type the name and description of the task.
4. Click the security option and click the Triggers tab.
5. Click New, define the triggering schedule, and click the Actions tab.
6. Click New and define the parameters as follows:
   - **Action:** Start a program
   - **Program/script:** <*SOI Install folder*>\SOI\Tools\soitoolbox.exe
   - **Start in:** <*SOI Install folder*>\SOI\Tools
   - **Add argument:** -x -q -s <*toolbox command*> 100
     - -x: for configuration file(Mandatory)
     - -q: for silent running(Mandatory
     - -s: service status and all(Optional)
     - toolbox commands: Commands such as --purgeClearedAlerts, --cleanHistoryData, --cleanImportedData, --purgeDbInconsistencies
7. (Optional) Click the Conditions and Settings tab and define the parameters.
8. Click OK.

SOI ToolBox Configuration

SOIToolbox utility reads credential from a configuration file. So, you can configure encrypted or plain password in the configuration file.

Follow these steps:

1. Execute the following command from the <SOI Install folder>\Tools location: soitoolbox.exe -x
   soitoolbox.cfg file is created exist.

2. Type the Windows credentials for all machines where CA SOI components are deployed:
   – yourSOImanager.yourcompany.com;administrator;paswd123
   – yourSOIUI.yourcompany.com;administrator;mpasswd
3. Type the credentials for CA SOI UI server access. This configuration required for the tool to list/start/stop the individual connectors.
   – soiui=soi-ui-machine;http;7070;samuser;oneclick
4. Type the database details and database location
   Syntax: database=<database name>;<machine>;<instance>;<port>;<sa>;<mydbpwd>
   Example: database=SAMStore;machineC;instance;port;sa;mydbpwd
   <mydbpwd> is encrypted password

Password Encryption

If you need encrypted password in the configuration file, follow these steps:

1. Execute the following command on the SOIToolbox utility for each password: soitoolbox.exe –-encryptPassword
2. Copy-paste the encrypted result into the file in the appropriate place.
   Syntax: database=<database name>;<machine>;<instance>;<port>;<sa>;<mydbpwd>
   Example: database=SAMStore;machineC;instance;port;sa;mydbpwd
   <mydbpwd> is encrypted password
3. Execute the following command: soitoolbox -x –getAvailableConnectors
   Where –x is the credential item taken from the configuration file instead of user input while command is running.

Using

This section describes how operators use CA SOI to manage services, respond to events, and run reports.

Operations Console Basics

This section describes how operators use the Operations Console.

Start the Operations Console

As an administrator or an operator you access the Operations Console from the Dashboard.

On the Dashboard, click Console.

The Java Web Start dialog appears and indicates that the application is downloading then launching.

> **NOTE**
> You can receive an error that indicates you must install a higher JRE version. If the automatic installation does not work, manually download and install the latest JRE from the Java website and try to launch the Operations Console again.

Operations Console Panes

Contents

The Operations Console is the user interface that provides most of the CA SOI functionality. The Operations Console is where you define and maintain services, create user groups and assign them access privileges, create event policies and alert queues, manage alerts that warn about fault conditions, and more.

The Operations Console is a unified alert console with features that let you centralize the management of all actionable conditions within a single interface. This provides a single view for your operations teams to consider the conditions requiring attention, and a single escalation point for those conditions to reduce the overhead associated with alarm management across multiple domains and to help ensure consistent policies are always applied.

The Operations Console is comprised of three panes that display information about your services, customers, users, and alerts: the Navigation pane, the Contents pane, and the Component Detail pane.

The information displayed in each of the panes depends on the items selected in the other panes. Selected items in the Navigation pane determine what is displayed in the Contents pane. Likewise, selected items in the Contents pane determine the information shown in the Component Detail pane.

Navigation Pane

The Navigation pane contains the following tabs:

- **Services**

  Lists services that have been imported from the domain managers or defined in CA SOI that you have the access privileges to view. You can navigate to the resources in the services, like servers and routers. The columns in the Services tab indicate whether each object is in maintenance mode, the granularity level, and the number of alerts of each severity currently open for each item.

Use the Filter fields in the List and Services tabs of the Contents pane to find specific CIs and services in the Services tab. Double-click a result to open the CI or service in the Services tab.

- **Alert Queues**

  Lists the set of alert queues and alerts on services that you have the access privileges to view. Each column displays the total number of alerts with that severity and the summation symbol column indicates the total number of alerts. Clicking an alert queue displays the alerts in that queue in the Contents pane. The alert queue icon color indicates the highest severity of any alert in the queue. Clicking the Information tab displays general information about the alert queue, associated escalation policies, cleared alert history, and user groups assigned to the alert queue.
- **Customers**
  Lists the defined customers. You create customers and associate them with service models to see the impact of service degradation on the service consumer.
- **Users**
  Lists the users and user groups that create service definitions, monitor alerts, and resolve the situations that cause the alerts.

Contents Pane

The Contents pane in the upper right of the Operations Console displays information about services, alerts, customers, or users, depending on the tab selected in the Navigation pane.

> **NOTE**
> Some tabs are not available based on selections in the Navigation pane.

When you select Services, the Contents pane displays the following tabs that contain information about services:

- **Alerts**
  Shows alerts for the service or CI selected in the Navigation pane. Select an alert to display details about it in the Component Detail section.

  > **NOTE**
  > Your administrator may have configured your view to be a subset of available alerts based on your role in the organization.
- **List**
  Displays resource name, health, quality impact, risk, granularity level, class, family, and IP address for resources that are direct children of the object selected on the Navigation pane. Use the Filter field in this tab to filter the displayed child CIs and services based on specified text. Double-click any object in this tab to open it in the Services tab.

  > **NOTE**
  > Quality impact and risk only apply to services.
- **Services**
  Displays service name, SLA status, health, quality impact, risk, priority, granularity level, operational mode, and management tier for subservices of the selected service, or for all services if you select the top-level Services item on the Services tab. Use the Filter field in this tab to filter the displayed services or subservices based on specified text. Double-click any object in this tab to open it in the Services tab.
  Consider the following:
  - The Tier column is hidden by default. Add this column if you have multiple CA SOI tiers. The column displays a value of Remote for services that come from remote CA SOI tiers. Any service with a value of Remote cannot be modified in the local Operations Console.
  - The total number of services column (designated by a summation sign) is not shown by default. You can show this column by changing your Preferences for the Services Tab, Service Table Columns.
- **Topology**
  Displays a diagram that shows the relationships among the resources and subservices of the service selected in the Navigation pane.

NOTE
For information about topology, see Navigate the Topology View.

- **Customers**
  Displays a list of all the customers that are associated with the service selected in the Navigation pane. This tab includes information about the customer name, customer identity, customer metrics, priority, and description.
- **Information**
  Shows details such as SLAs, maintenance schedules, role-based security for user groups, and the connector status for the selected service or CI. You can set service or CI properties such as Operational Mode, Priority, Location, maintenance schedules, and access privileges from the Information tab.
- **CMDB View**
  Displays the CA Service Desk interface in the context of CA CMDB CIs or services in CA SOI. The CMDB View tab only appears when a CA Service Desk/CA CMDB connector is present in any status on the SA Manager, and it is only selectable when you select an object from CA CMDB in the Operations Console. When you select a CA CMDB object and click the CMDB View tab, the CA Service Desk Log In page appears. Enter CA Service Desk user credentials to open the CA Service Desk interface that displays CI details, and click Visualizer to open the CMDB Visualizer, which displays the selected object and objects that relate to it.

  NOTE

  For more information about using and interpreting the CMDB Visualizer, see the CA CMDB documentation.

  NOTE

  The CMDB link works only when you start the Operation Console with 32 bit JRE.

When you select Alert Queues, the Contents pane displays the Alerts tab, which shows the alerts that belong to the alert queue selected in the Navigation pane.

When you select Users, the Contents pane displays the following tabs that contain information about users:

- **Users List**
  Lists user name and type information for users in the group selected in the Navigation pane.
- **Privileges**
  Lists access privileges that you can assign to user groups.
- **Service Access**
  Lists services to which the selected user group has access.
- **Alert Queues Access**
  Lists alert queues to which the selected user group has access.

When you select Customers, the Contents pane displays the following tabs that contain information about customers:

- **Alerts**
  Shows alerts for the customer (or sub-customer) selected in the Navigation pane. Only alerts from the services that are assigned to the selected customer are shown in the tab. Select an alert to display details about it in the Component Detail section. For example, select the Customer Impact tab to view all the customers that the selected alert is impacting.
- **Services**
  Displays services associated with the customer (or sub-customer) selected in the Navigation pane.
- **List**
  Displays a list of sub-customers for the selected customer.
- **Information**
  Displays information (such as customer name, priority, identity, and description) about the selected customer (or sub-customer).

Component Detail Pane

The Component Detail pane in the lower right of the Operations Console displays detailed information about alerts, resources, services, service impact, customer impact, users, or user groups. The information varies, depending on which tab you select in the Contents pane.

The Component Detail pane contains the following tabs, some of which may be unavailable based on selections in the Contents pane and access privileges:

- **Alert Details**Shows general details, annotations, update history, user groups with access to the alert queue, escalation action history, user-defined attributes, USM attributes, and so on, for the alert selected on the Alerts tab in the Contents pane. You can set alert properties and make annotations from this tab.
- **Information**
  Shows more information (such as general, SLAs, maintenance schedules, role-based security for user groups, associated escalation policies, connector status, and granularity) about the item selectedon the Contents pane. If an alert is selected, the Information tab shows information about the service or CI associated with the alert. You can set service or CI properties such as Operational Mode, Priority, Location, maintenance schedules, and access privileges from the Information tab.
- **Root Cause**

  Displays the alerts that have the highest impact on a service. If multiple alerts have equally high impact, you may see more than one root cause alert. Root cause alerts are categorized as Root Cause, Symptom, or Unclassified:

  - **Root Cause**
    Identifies the root cause alert is the actual cause.
  - **Symptom**
    Identifies the root cause alert is part of a root cause rule, but is not the root cause of the alert.
  - **Unclassified**
    Identifies the root cause alert as neither Root Cause nor Symptom classifications.

  Service Impact

  Shows the business impact associated with the resource where the alert originated. This tab shows the name of the resource, the name of the associated service, the impact of the resource, and the health of the resource. Root cause alerts display all services impacted, including parent and associated services other than the service in which the CI is directly located. Non-root cause alerts only display the service in which the CI is located.

  Customer Impact

  Shows the customer impact level for all the customers that the selected service alert is impacting.

  Alerts

  Shows the alerts associated with the service selected from the List, Services, or Topology tab.

  Cleared Alert History

  Shows the alerts that have been created and cleared from a service or CI selected from the List, Services, or Topology tab. By default, this tab displays alerts cleared within the last 24 hours. You can change the time range using alert filters.
- **USM Properties**
  Shows the USM properties of the selected CI from the List, Services, or Topology tab.
- **USM Notebook**
  Shows the USM properties of the CI from the perspective of all data sources and the reconciled set of USM properties. When cross-type correlation occurs, you can see the USM properties for all correlated types, not just the reconciled sheet type, on the USM Notebook tab. To view USM properties for a projection sheet of a different CI type than

the reconciled sheet, select that type from the Type drop-down list on the USM Notebook tab. Entries exist for the reconciled sheet, the CA SOI CI projection sheet, and any other projection sheet of different types.

- **SOI Properties**
  Shows the CA SOI-specific properties of the selected CI. These properties are unique values that CA SOI uses to identify each entity.

Status Bar

The Status bar at the bottom of the Operations Console provides status information about the current CA SOI connection. The Status bar contains the following buttons:

-
  Opens the Connection Status dialog, which displays the current connection status of all CA SOI components (SA Manager, UI Server, connectors, and so on). The Connection Status dialog also displays a connection log and an Open Consolidated Error Log button, which opens the consolidated samerror.log file in a web browser.
  This button displays a green icon to indicate that all components have a connected status. The button displays a red icon to indicate that one or more components have lost connection.

  NOTE
  For the User Management Service, even if CA EEM displays without a version number in the Description column, it does not indicate a problem with the product. The service still displays as Online.

-
  Opens the Messages dialog, which displays any new messages from the administrator.

If the SA Manager connection is lost, the Status bar also displays your user name, the login server, and an alert message.

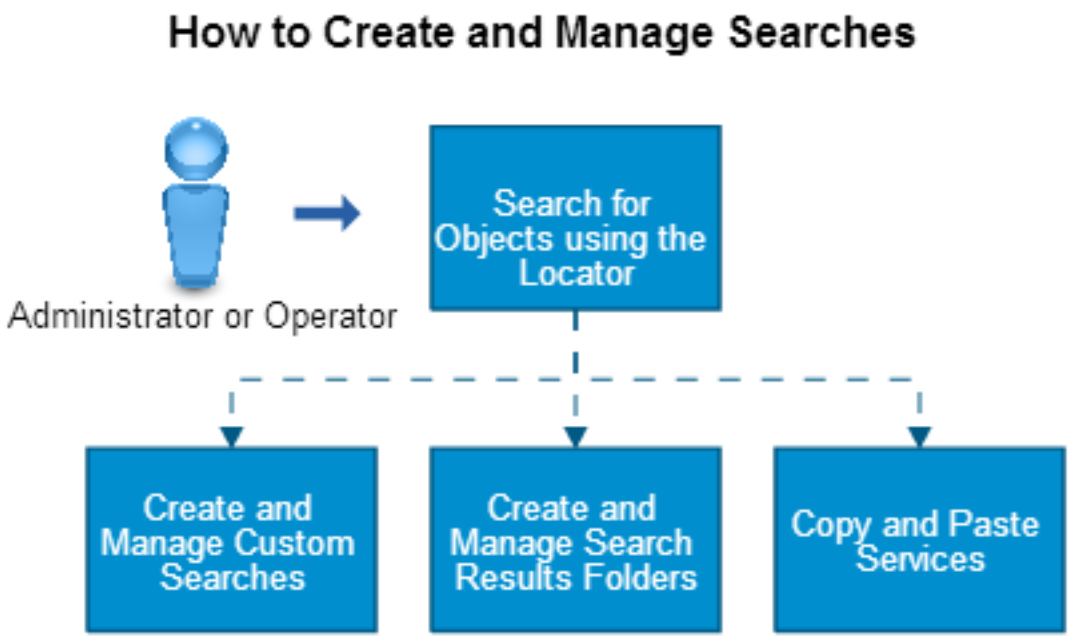
How to Create and Manage Object Searches

Contents

As an administrator or an operator, you can search for objects in the CA SOI database with the Locator tool. Searches either run automatically when activated or prompt you for an input before performing the search. The Locator is useful when you know only partial information such as a part of an IP address. You can search monitored objects that belong to a service and staged objects that do not belong to a service.

Use this scenario to guide you through the process:

Figure 50: how to create and manage searches



1. Search for objects using the Locator.
2. (Optional) Create custom searches.
3. (Optional) Create search result folders.
4. (Optional) Copy and paste services.

Search for Objects Using the Locator

You can search for objects in the SA Store database with the Locator tool. Searches either run automatically when activated or prompt you for input before performing the search. The Locator is useful when you know only partial information such as a part of an IP address. You can search both monitored and staged objects.

> **NOTE**
>
> CA SOI only supports case-insensitive databases. Therefore, if you enable the Ignore Case check box, the database cannot differentiate between two values in the search if the only difference between them is their case.

Follow these steps:

1. Open the Operations Console and click the Locator icon.
2. Double-click the item you want to search by in the Search Objects By tree.
   You can search by the following items:
   – *Custom Search Name*
     Custom searches are user-created and named searches that either run automatically when activated or prompt you for input when activated.
   – **Category**

Specifies the object category, which is an optional string passed by an individual silo for each CI. Enter a partial or full search term.
Examples: Windows, Linux, or database

- **CIID**
  Specifies the CI identification number, which CA SOI generates as a unique number for each CI in the database. Enter a partial or full search term.
- **Class**
  Specifies the class to search by. Select the class type from the drop-down list.
- **Description**
  Specifies a manually entered object text description. Enter a partial or full search term.
- **Device ID**
  Specifies the device identification number. Enter a partial or full search term.
- **Granularity**
  Specifies the granularity level (Low or Normal). Select the granularity level from the drop-down list.
- **Instance ID**
  Specifies the instance identification number. Enter a partial or full search term.
- **IP Address**
  Specifies the IP address. Enter a partial or full search term.
- **Launch in Context URL**
  Specifies the URL for launching the source application. Enter a partial or full search term.
- **Location**
  Specifies the physical address where the object resides.
- **Name**
  Specifies the object name, which you see in the Console. Enter a partial or full search term.
- **Namespace Map ID**
  Specifies the CI namespace map identification number. Enter a partial or full search term.
- **Notebook ID**
  Specifies the USM notebook ID number. Enter a partial or full search term.
- **Operational Mode**
  Specifies the operational mode. Select the operational mode from the drop-down list. You most often search for an operational mode of Maintenance.
- **Sheet ID**
  Specifies the identification number for a USM projection sheet. Enter a partial or full search term.
- **Source**
  Specifies the source domain manager from which the CI originated. For example, you could search for all CIs from a specific CA Spectrum installation. Each source is defined uniquely using a five-digit ID number defined by the USM schema. Select the object source from the drop-down list.

  NOTE
  For a list of connectors and their five-digit MdrProduct ID numbers, see Connector Identification Numbers.

3. If a Search dialog displays, complete the dialog to perform the search:
   a. Enter a search value.

      NOTE
      Leave the field empty to search all objects.

   b. (Optional) Select the Search Monitored Objects only check box to restrict the search to return only objects associated with a managed service. Clearing the check box includes all items in the search results, whether or not they are part of a managed service.
4. Click OK.
   Either the Results tab displays in the right pane with the results or a dialog indicates that no search objects were found.

NOTE
Search results are limited to 5,000 by default. For more information about changing the default limit, see Set Preferences.

5. (Optional) Right-click on a results row and select an item from the pop-up dialog:
   – Copy the monitored service into memory. You can then paste the object into the Navigation tree. For more information, see Copy and Paste Services.
   – Add the selected objects to the modeled service in the Modeler. This option is available only if the Modeler is open; if multiple Modeler windows are open, the object is added to the last window opened.
   – Locate the object in the Navigation tree.

Create Custom Searches

You can create a custom search using comparisons and Boolean operators then save the search for reuse. You can also edit and delete the searches using the respective icons.

Follow these steps:

1. Open the Operations Console and click the Locator icon.
2. Click the Create a new search icon.
3. Select an attribute and comparison type.
4. Perform one of the following actions:
   – Select the Prompt check box then complete the Prompt for Value field. You are then prompted to enter an attribute value when you launch the search.
   – Enter an attribute value or select from the drop-down list (if available).
5. (Optional) Click Show Advanced and add more attribute criteria and create advanced logic using the logic buttons on the right-hand side of the dialog, then click Add.

   NOTE
   For more information about creating advanced attribute criteria, click the Hints link above the expression pane.

   The attribute expression appears in the lower pane.
6. You can perform the following actions:
   – Click Launch to run the custom search.
   – Click Save As to set/change the search name or user access privileges for the custom search.
   – Click OK to add the custom search to the Search Objects by list once you have set a custom search name using Save As.

Create Custom Search Folders

You can create folders and subfolders to organize your searches. You can also edit and delete the folders.

Follow these steps:

1. Open the Operations Console and click the Locator

   icon.
2. Click the Organize icon.
3. (Optional) Select a folder if you want to create a subfolder.
4. Click Create Folder.
5. Enter a folder name and click OK.

Copy and Paste Services

You can copy and paste monitored services from the Locator search results and paste them as subservices in the Navigation pane.

Follow these steps:

1. Right-click a monitored service from the Locator results pane.
2. Click the Copy Service

   icon.

   NOTE
   If the service is unmonitored, the icon is dimmed.
3. Switch to the Navigation pane.
4. Right-click the service or subservice under which you want to paste the copied service and select Paste.

   NOTE
   If you copy and paste a top-level service, the original service is removed from its original top level and pasted as a subservice. However, if you paste a copied subservice, the original subservice appears in its original location and the new location.

The copied service appears under the selected location.

How to View Object Audit Trails

Contents

As an administrator or an operator (with access privileges), you can use the Auditor to view the audit trail for objects (such as services, CIs, alerts, connectors, and so on). An audit trail shows updates to a selected object as tracked in the CA SOI database.

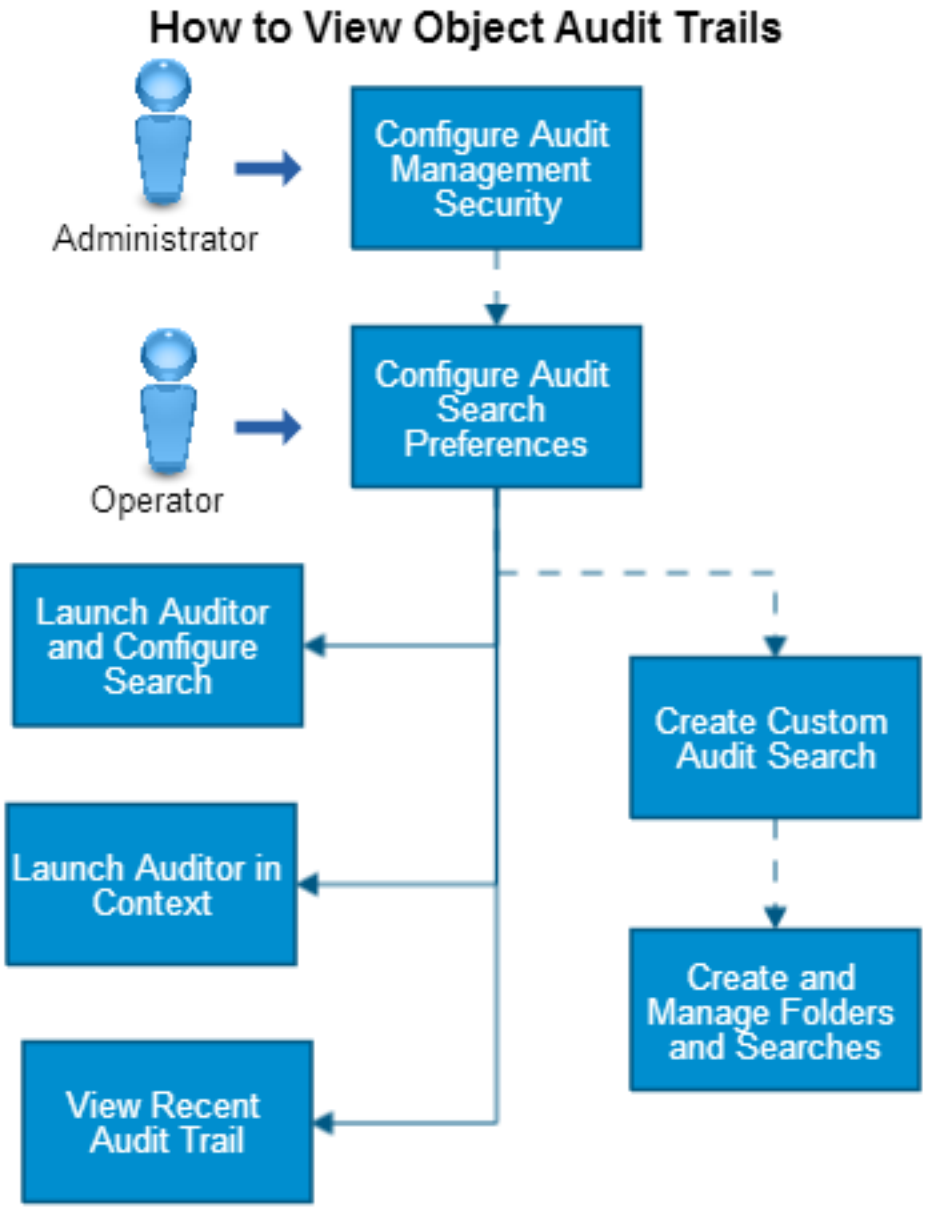
You can also create and manage custom searches and folders.

The following examples are a small subset of the many audit trail updates available:

- a state change
- acknowledged
- cleared
- created
- deleted
- maintenance mode

Use this scenario to guide you through the process:

Figure 51: how to view object audit trails



1. (Optional) Configure the audit management security.
2. (Optional) Configure your audit search preferences.
3. Perform any of the audit trail searches:
   - Launch the Auditor and manually enter the audit trail search parameters.
   - Launch the Auditor in the context of an object.
   - View the recent audit trail for an object.
4. (Optional) Create custom audit trail searches.
5. (Optional) Create and manage your custom audit trail search folders and searches.

Configure Audit Management Security

The default user groups that CA SOI provides have Auditor access. However, if an administrator creates a user group, the administrator must configure the user group access in the Privileges tab Audit Management section.

Configure Audit Search Preferences

You can configure the maximum number of audit trail entries that CA SOI retrieves in the Auditor and the Information tab sub-views.

On the Operations Console, select View, Preferences, then select Auditor.

Launch Auditor and Manually Enter Audit Trail Search Parameters

You can launch the Auditor and manually select or enter search criteria.

Follow these steps:

1. On the Operations Console toolbar, click the Auditor

   icon.
2. Double-click a search type and perform the indicated action:
   - **Action Type**
     Specifies the type of action that either CA SOI or a user performed. Select an action from the drop-down list.
   - **Internal ID**
     Specifies the internal object or model identification number. This number varies depending on the object type. For alerts, the ID is the alert ID, for CIs and services, the ID is the CIID, and so on. Enter the number.
   - **Record Type**
     Specifies the object record type. Select a record from the drop-down list.
   - **Time Stamp Range**
     Specifies the audit trail time stamp begin and end time search range. Select or enter the date and time for the range.
   - **User Name**
     Specifies the user login that performed the action. Enter the user name.
3. Click OK.

Launch Auditor in Context

You can launch the Auditor in context so that CA SOI automatically generates a search that is based on the context selection. For example, if you launch the Auditor with an alert selected, the Auditor shows the audit trail for the alert.

Right-click an object and select Launch CI Audit on objects in the following Operations Console locations:

- Navigation pane
- Topology chart
- Locator results

View Recent Audit Trail

You can view the recent audit trail entries for a selected object.

Follow these steps:

1. Select an object in any of the following locations:

- Navigation pane
- Topology chart

2. In the Component Details pane Alerts tab or Information tab, expand the Most Recent Audit Trail section.

> **NOTE**
> If you select an alert in the Contents pane, the Information tab Recent Audit Trail section shows the entries for the CI related to the alert.

Create Custom Audit Trail Searches

You can create a custom audit trail search using comparisons and Boolean operators then save the search for reuse. You can also edit or organize the searches using the respective icons.

Follow these steps:

1. On the Operations Console toolbar, click the Auditor

   icon.
2. Click the Create a new search

   icon.
3. Select an attribute and comparison type.
4. Perform one of the following actions:
   - Select the Prompt check box then complete the Prompt for Value field. You are then prompted to enter an attribute value when you launch the search.
   - Enter an attribute value or select from the drop-down list (if available).
5. (Optional) Click Show Advanced and add more attribute criteria and create advanced logic using the logic buttons on the right-hand side of the dialog. Then click Add.

   > **NOTE**
   > For more information about creating advanced attribute criteria, click the Hints link above the expression pane.

   The attribute expression appears in the lower pane.
6. You can perform the following actions:
   - Click Launch to run the custom audit trail search.
   - Click Save As to set/change the audit trail search name for the custom audit trail search. You can also select a folder for the custom search.

Create and Manage Custom Audit Trail Search Folders and Searches

You can create folders and subfolders to organize your audit trail searches. You can also rename, move, and delete the folders and custom audit trail searches.

Follow these steps:

1. On the Operations Console toolbar, click the Auditor

   icon.

2. Click the Organize

   icon.
3. (Optional) If you want to create a subfolder, select a folder.
4. Click Create Folder.
5. Enter a folder name and click OK.

View the Service Membership of a CI

In complex CA SOI environments, CIs can belong to multiple services. Understanding which services a CI belongs to is important when a problem with a CI occurs or when you are planning infrastructure changes. The Operations Console offers a convenient method to derive the Service Membership of a CI.

Follow these steps:

1. Open the Operations Console.
2. Do one of the following to locate the CI:
   – Drill down into a service from the Services pane and select the CI.
   – Click the Locator icon on the menu bar, search for the CI by name, and double-click it to see the CI in the Services pane.
3. Right-click the CI and select Location.
   A list of services to which the CI belongs appears in the menu.

Alert Management for Operators

This section describes how operators monitor and manage alerts in the Operations Console.

See also:

- Introduction to Alert Management

View Alerts, Alert Details, and Extended Information

Contents

As an operator, you can view alert properties, service and customer impact, root cause, alert source, alert history, alert tables, and create alert filters.

You can view alerts in several different ways in the Operations Console.

- To view all alerts impacting services, select the Services object at the top of the tree on the Services tab in the Navigation pane.
  Alerts that impact all services are displayed on the Alerts tab in the Contents pane.
- To view all alerts impacting a specific service, expand the tree on the Services tab in the Navigation pane (if necessary) and select the service whose alerts you want to see.
  Alerts that impact the service are displayed on the Alerts tab in the Contents pane.
- To view all collected alerts (which may or may not impact services), select the Alert Queues tab.
  The Alert Queues tab appears with the Alert Queues folder selected. Select a user-defined queue to view the alerts in that queue, or select the Default queue to view all alerts that do not belong in any other queue.
- To view details about an alert, click the alert in the Contents pane.

Additional details about the alert appear in the Alert Details tab of the Component Detail pane. Other details are also shown such as annotations, update history, and escalation history. Click the plus sign (+) icon in these sections to view a history of actions performed on the alert.

> **NOTE**
> You can also open the alert details as a separate window by right clicking the alert and selecting Alert Detail from the shortcut menu.

You can sort alerts by clicking the column headings on the Alerts tab.
The USM Properties and USM Notebook tabs display the USM properties for the alert. These properties differ from the properties displayed in the Operations Console, and you interact with these properties when you use Event Management functionality.

Alert Properties and Extended Alert Information

CA SOI alerts contain the following properties. Some properties originate from the domain manager. Other properties (for example, service impact and number of impacted services) originate from CA SOI.

 # Impacted Customers: Indicates the number of customers that the alert impacts. This number is based on the number of customers that are assigned to the service that the alert impacts.

 # Impacted Services: Indicates the number of services the alert impacts based on the number of services its associated CI is included in.

Acknowledged: Indicates whether an operator has acknowledged the alert. Assigned Indicates the name of the operator that is assigned to the alert.

Category: Indicates whether this alert condition affects the quality or risk of the services it impacts.

Class: Indicates the class (USM type) of the CI the alert is associated with.

Date / Time: Indicates the date and time when this alert was generated.

Family: Indicates the CI class family that the alert is associated with.

Highest Customer Impact: Indicates the highest impact value that the alert causes for an associated customer.

Highest Customer Priority: Indicates the highest customer priority of a customer that is associated with a customer associated with a related service.

Is Clearable: Defines whether the alert can be cleared in the domain manager. If you enabled the 'Respect Underlying MDR Clear Alert Setting', this property also determines whether the alert can be cleared in CA SOI.

Is Exempt: Indicates whether the alert is excluded from impact analysis calculations.

Maintenance: Indicates whether the CI associated with the alert is currently in maintenance mode.

Name: Indicates the associated CI that the alert condition impacted.

Service Impact: Indicates the impact, which is calculated by multiplying alert severity and the significance of the CI to the service. When multiple services are impacted, the most affected service is displayed.

Service Impact Value: Indicates the impact value of the service alert. This value is always a factor of 10. The Service Impact Value displayed in the Alerts table can be different from the service impact value for the corresponding service in the Topology tab. The Topology tab displays how the child objects impacted the service.

Severity: Indicates the alert severity that the originating domain manager assigned.

Source: Indicates the domain manager where the alert originated. The format is MdrProduct_domainserver@connectorserver. For example, CA:00005_spectrohost.ca.com@spectrohost.ca.com refers to a CA Spectrum connector installed on spectrohost.ca.com monitoring a CA Spectrum instance that is installed on the same system.

Source Alert ID: Indicates the ID number of the alert in the source domain manager. Only infrastructure alerts have a Source Alert ID, because service alerts are generated in CA SOI, not from a source domain manager. Summary Describes the alert condition.

Ticket ID: Indicates the ID of the associated help desk ticket. Unmanaged Indicates whether the alert is associated with any services. An unmanaged alert does not have a service association.

User Attribute (1-10): Indicates any configured customized values. These attributes are blank by default, but you can send values to the attributes through Event Management. You can also customize the attribute names.

You can also view the correlatable USM properties for the alert's associated CI:

- ModificationTime
- PrimaryIPV4Address
- PrimaryIPV4AddressWithDomain
- PrimaryIPV6Address
- PrimaryIPV6AddressWithDomain
- PrimaryMacAddress
- PhysSerialNumber
- BioSystemID
- Vendor
- AssetNumber
- PrimaryDnsName
- SysName

Note: For more information about USM properties, see the USM schema documentation. For information see How to Access the USM Schema Documentation.

In addition to these properties, alerts have associated extended information such as annotations, update history, and escalation history in the Alert Details tab. This information provides a full audit trail of the manual and automated actions that are taken to help diagnose and remedy an alert condition.

Annotations: Indicates the interim steps that were taken to resolve the situation that caused an alert. These comments highlight the incident management process in real time, and they can provide information for the problem management process.

Update History: Indicates how the alert has evolved since alert creation. Updates can include changes in severity and properties (such as the acknowledged flag).

Escalation Action History: Indicates the automated actions that notify, diagnose, or remedy the problem and the results of those actions. For example, if an email notification is sent, confirmation that it was sent successfully is included. If a remote device was pinged, the results are included. Escalation history therefore provides a detailed audit trail of the automated actions taken in response to an alert condition.

Alert Queues: Show the alert queues to which the alert belongs.

User Defined Attributes: Displays the names and values of the user-defined attributes.

Most Recent Audit Trail: Provides a list of recent object actions.

View All Services Impacted by an Alert

One CI may support more than one service; therefore, alert conditions affecting that CI may impact multiple services. CA SOI lets you see all services impacted by a root cause alert and shows an impact value based on how critical that CI is to each service.

Follow these steps:

1. Expand the tree on the Services tab in the Navigation pane (if necessary) and select the service whose alerts you want to view, or select the Services object to display all alerts.
   The alerts are displayed in the Contents pane on the Alerts tab.
2. Click the alert whose impacted services you want to see.
   The Alert Details tab opens by default in the Component Detail pane.
3. Click the Service Impact tab in the Component Detail pane.
   The Service Impact tab displays a list of services this alert impacts, the extent of the impact on each service, and the current health of the service. Root cause alerts display all impacted services, including parent and associated services other than the services where the associated CI is located. Non-root cause alerts only display the service in which the associated CI is located.

View All Customers Impacted by an Alert

Alerts can impact customers that are associated with impacted services. If a customer is associated with a service and an alert is generated that impacts that service, the alert also impacts the associated customer. An alert can impact multiple customers when multiple customers are associated with a single service, or when an alert impacts multiple services that each have associated customers.

Follow these steps:

1. Select an entity on any tab other than Users in the Navigation pane.
   The Alerts tab in the Contents pane shows all alerts related to the selected entity in the Navigation pane.
2. Click the alert whose impacted customers you want to see.
   The Alert Details tab opens by default in the Component Detail pane.
3. Click the Customer Impact tab in the Component Detail pane.
   The Customer Impact tab displays a list of customers this alert impacts and the customer impact value for each customer. Customer impact derives its value from alert severity and customer priority.

View the Root Cause of a Service Alert

CA SOI analyzes the alerts associated with a service to determine which alert has the highest impact and identifies this as the root cause alert. CA SOI classifies root cause alerts as Root Cause, Symptom, or Unclassified. CA SOI provides rules that determine the root cause alert classification. CA SOI determines the classification in the following order:

- **Root Cause**
  Identifies the root cause alert is the actual root cause.
- **Symptom**
  Identifies the root cause alert is part of a root cause rule, but is not the root cause of the alert.
- **Unclassified**
  Identifies the root cause alert as neither Root Cause nor Symptom classifications.

For example, ComputerSystem A is low on memory, which causes Application X to run out of memory. The Root Cause is the low memory alert associated with ComputerSystem A. The Symptom is the out of memory alert associated with Application X. The root cause rules establish that the low system memory can cause the out of memory errors on the application, and the service model ensures that Application X is running on ComputerSystem A.

You can use the root cause classification as an attribute criteria for creating escalation policy and alert queue rules.

Follow these steps:

1. Expand the tree on the Services tab in the Navigation pane (if necessary) and select the service whose alerts you want to view, or select the Services object to display all alerts.
   The alerts are displayed in the Contents pane on the Alerts tab.

2. (Optional) Select the filter Service Alerts from the Available filters drop-down list on the Alerts tab of the Contents pane.
   The Alerts tab displays only service alerts.
3. Click the alert whose root cause you want to see.
   The Alert Details tab opens by default in the Component Detail pane.
4. Click the Root Cause tab in the Component Detail pane.
   The Root Cause tab displays the alert that corresponds to the root cause of the fault condition that affects the service.

   NOTE
   If multiple alerts have equally high impact, you may see more than one alert.

Launch Alert Source

From the Operations Console, you can launch the domain manager application that is the source of an infrastructure alert to view more information about the issue.

To launch alert source, right-click an infrastructure alert and select Launch <*Domain Manager*>, where *Domain Manager* is the name of the domain manager interface to launch.

The domain manager interface opens in the context of the alert. You may have to enter valid product credentials to log in to the interface.

If the Launch option is not available, configure launch in context in the connector. For information about how to configure launch in context in CA Catalyst connectors, see the *Connector Guide*.

View Cleared Alert History

CA SOI maintains a history of cleared alerts in its database. You can view alerts that have been cleared from services or alert queues in the Operations Console. The Cleared Alert History table contains the following information:

- Associated CI name
- Severity
- Creation and clear date
- ClearedBy
- Category and summary
- Acknowledged status
- Data source

Follow these steps:

1. Select a service or alert queue from the Services or Alert Queues tab.
   The Contents and Component Details panes populate with information about the selected service or alert queue.
2. (Services only) Select the Cleared Alert History tab in the Component Detail pane.
   The Cleared Alert History tab displays cleared alerts that previously belonged to the service.
3. (Alert queues only) Select the Information tab in the Contents pane and scroll to the Cleared Alert History table.
   The Cleared Alert History table displays cleared alerts that previously belonged to the alert queue.

Print a Table of Alerts

You can print the alerts associated with a resource or service in tabular format on a local or network printer. The table is in the same format as the container on the Alerts tab. If one of your printers is Adobe PDF, you can create a PDF file.

Notes:

- You can print the entire table or only selected alerts. If one alert is selected, you can print alert details for it.
- You can print alert annotations, history, and associated queues in the Component Detail pane.

Follow these steps:

1. Right-click any alert and select Print.

   NOTE
   An alternative is selecting File, Print, and then selecting *Alerts* from the drop-down list on the dialog that opens.
2. Select printer options, if necessary, and click OK.

Export a Table of Alerts

You can export the alerts associated with a service to a CSV file. The table is in the same format as the container on the Alerts tab. You can use the file in a spreadsheet or other application that reads comma-separated value files.

NOTE
You can also export alert annotations, history, and associated queues in the Component Detail pane.

Follow these steps:

1. Expand the tree on the Services or Alert Queues tab in the Navigation pane (if necessary) and select the service or alert queue whose alerts you want to export.
   The alerts for that service or alert queue are displayed in the Contents pane on the Alerts tab.
2. Click

   on the toolbar.
3. Select a location and click Save.

Create an Alert Filter

An *alert filter* limits the number and alert types that are shown on the Operations Console.

For example, you might filter alerts with the severity Minor from devices you are not responsible for monitoring, or that are acknowledged. You can suppress alerts based on information in the following tabs:

- **Severity**
  Lets you filter alerts based on severity. The available severities are Unknown, Down, Critical, Major, and Minor.

  NOTE
  By default, alerts with a severity or Unknown are automatically converted to Minor.
- **Service Impact**
  Lets you filter alerts based on service impact. The available service impact values are Down, Moderate, None, Severe, and Slight.
- **Family**
  Lets you filter alerts based on a specified grouping of classes. The filter hides any alert belonging to a family that you hide.
- **Class**
  Lets you filter alerts based on the USM type of the CI that caused the alert.
- **State**
  Lets you filter alerts based on the acknowledged state and context.
- **Attribute**
  Lets you filter alerts based on specific attribute values.

You can create simple alert filters in which all conditions set on all tabs must be met for alert suppression. Alternatively, you can create complex alert filters, in which you use OR logic to create several filtering conditions independent of one another.

Follow these steps:

1. Open the Operations Console and click an item displayed in the left pane that has the alerts you want to filter.
   The right pane refreshes to display alerts for the selected items.
2. Right-click an alert in the right pane, and select Set Filter from the shortcut menu.

   NOTE
   Alternatively, you can select the alert and click the Filter button

   on the toolbar.

   To create an AND expression, navigate the tabs as described in Steps 3 through 6.
3. (For the Severity, Service Impact, Family, and Class tabs) Select the alert severities, impact values, families, and classes to filter in the Show pane of each tab, and click the Add Selected button

   .

   The selected values appear on the Hide pane of each tab.
4. Click the State tab, and use the following panes to configure state filtering:
   - **Acknowledged State**
     Lets you filter alerts by whether they are acknowledged. Select Acknowledged to show only acknowledged alerts; select Not Acknowledged to show only unacknowledged alerts. The default setting (Both) does not filter alerts by their acknowledged state.
   - **Selected Context**
     Lets you filter alerts by your selection and the alert context.
     Select the 'Show Only Service Alerts For Services/All for Infrastructure Items' check box to show only service alerts when you select a service in the Operations Console. When you select this setting, infrastructure alerts only display when you select their corresponding CIs.
     The 'Hide Alerts for Services and Infrastructure Items that are currently in Maintenance' check box, which is selected by default, hides alerts for services and CIs in maintenance.
5. Click the Attribute tab, select an attribute (which includes CA SOI attributes and a subset of USM attributes for the associated CI) on which to filter, a comparison type, and a comparison value for the attribute, and click Add.

   NOTE
   To use regular expressions, select Matches regex from the Comparison Type drop-down list. Click Test Regex to open the Regex Tester and test the regular expression against a string. Regular expressions are not available for all attributes.

   The attribute expression appears in the lower pane.
6. (Optional) Add more attribute filters and create advanced logic using the logic buttons on the right of the dialog.
   For more information about creating advanced attribute filters, click the Hints link above the expression pane.

   NOTE
   If you want to create a simple alert filter with no advanced logic, omit Steps 7 through 9 and continue with Step 10.
7. Click Show Advanced.
8. Click Add.
   An expression appears in the text pane that links all conditions you specified using AND logic.

   NOTE
   Some conditions display the hidden values while others display the shown values, depending on the text length of each option. The Severity condition always displays the hidden values (such as Severity (Hide Minor)) and the Service Impact condition always displays the shown values (such as Service Impact (Show Moderate, Severe) regardless of text length.

9. Create additional conditions using the Alert Filter tabs, and click Add when you finish.
   Each additional expression appears in the text pane related to every other condition with an OR operator. Therefore, add all conditions that you want linked by AND logic in the same expression. Separate conditions into different expressions to link them through OR logic.
10. Click Add at the bottom of the dialog when you finish.
11. Enter a filter name and click OK.

    NOTE
    The filter name must have fewer than 47 characters to prevent window display distortion.
12. Click OK on the Alert Filter dialog.
    The filter is created.

CA SOI alerts contain the following properties. Some properties originate from the domain manager. Other properties (for example, service impact and number of impacted services) originate from CA SOI.

- **# Impacted Customers**
  Indicates the number of customers that the alert impacts. This number is based on the number of customers that are assigned to the service that the alert impacts.
- **# Impacted Services**
  Indicates the number of services the alert impacts based on the number of services its associated CI is included in.
- **Acknowledged**
  Indicates whether an operator has acknowledged the alert.
- **Assigned**
  Indicates the name of the operator that is assigned to the alert.
- **Category**
  Indicates whether this alert condition affects the quality or risk of the services it impacts.
- **Class**
  Indicates the class (USM type) of the CI the alert is associated with.
- **Date / Time**
  Indicates the date and time when this alert was generated.
- **Family**
  Indicates the CI class family that the alert is associated with.
- **Highest Customer Impact**
  Indicates the highest impact value that the alert causes for an associated customer.
- **Highest Customer Priority**
  Indicates the highest customer priority of a customer that is associated with a customer associated with a related service.
- **Is Exempt**
  Indicates whether the alert is excluded from impact analysis calculations.
- **Maintenance**
  Indicates whether the CI associated with the alert is currently in maintenance mode.
- **Name**
  Indicates the associated CI that the alert condition impacted.
- **Service Impact**
  Indicates the impact, which is calculated by multiplying alert severity and the significance of the CI to the service. When multiple services are impacted, the most affected service is displayed.
- **Service Impact Value**
  Indicates the impact value of the service alert. This value is always a factor of 10. The Service Impact Value displayed in the Alerts table can be different from the service impact value for the corresponding service in the Topology tab. The Topology tab displays how the child objects impacted the service.
- **Severity**

Indicates the alert severity that the originating domain manager assigned.

- **Source**
  Indicates the domain manager where the alert originated. The format is *MdrProduct_domainserver@connectorserver*. For example, CA:00005_spectrohost.ca.com@spectrohost.ca.com refers to a CA Spectrum connector installed on spectrohost.ca.com monitoring a CA Spectrum instance that is installed on the same system.
- **Source Alert ID**
  Indicates the ID number of the alert in the source domain manager. Only infrastructure alerts have a Source Alert ID, because service alerts are generated in CA SOI, not from a source domain manager.
- **Summary**
  Describes the alert condition.
- **Ticket ID**
  Indicates the ID of the associated help desk ticket.
- **Unmanaged**
  Indicates whether the alert is associated with any services. An unmanaged alert does not have a service association.
- **User Attribute (1-5)**
  Indicates any configured customized values. These attributes are blank by default, but you can send values to the attributes through Event Management. You can also customize the attribute names.

You can also view the correlatable USM properties for the alert's associated CI:

- ModificationTime
- PrimaryIPV4Address
- PrimaryIPV4AddressWithDomain
- PrimaryIPV6Address
- PrimaryIPV6AddressWithDomain
- PrimaryMacAddress
- PhysSerialNumber
- BioSystemID
- Vendor
- AssetNumber
- PrimaryDnsName
- SysName

> **NOTE**
> For more information about USM properties, see USM schema documentation.

In addition to these properties, alerts have associated extended information such as annotations, update history, and escalation history in the Alert Details tab. This information provides a full audit trail of the manual and automated actions that are taken to help diagnose and remedy an alert condition.

- **Annotations**
  Indicates the interim steps that were taken to resolve the situation that caused an alert. These comments highlight the incident management process in real time, and they can provide information for the problem management process.
- **Update History**
  Indicates how the alert has evolved since alert creation. Updates can include changes in severity and properties (such as the acknowledged flag).
- **Escalation Action History**
  Indicates the automated actions that notify, diagnose, or remedy the problem and the results of those actions. For example, if an email notification is sent, confirmation that it was sent successfully is included. If a remote device was pinged, the results are included. Escalation history therefore provides a detailed audit trail of the automated actions taken in response to an alert condition.
- **Alert Queues**

Show the alert queues to which the alert belongs.

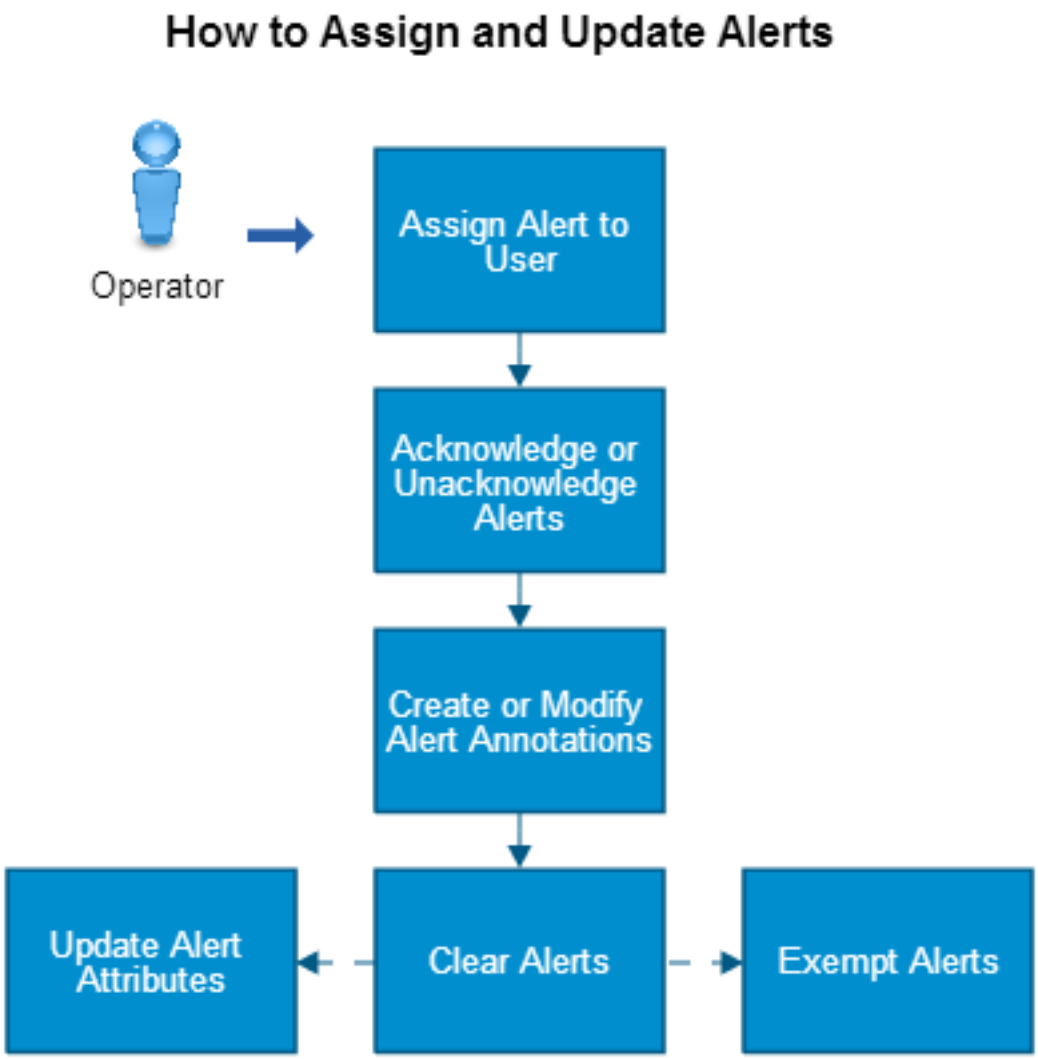
- **User Defined Attributes**
  Displays the names and values of the user-defined attributes.
- **Most Recent Audit Trail**
  Provides a list of recent object actions.

How to Assign and Update Alerts

As an operator you can assign alerts, acknolwedge/
unacknowledge alerts, annotate alerts, clear alerts, exempt alerts, and update alert attributes. You can also notify other users. Your administrator sets the features available to you. If a feature is not available, contact your administrator.

Use the following scenario to guide you through the process:

Figure 52: How to Assign and Update Alerts



1. Assign an alert to a user to resolve the issue.
2. Acknowledge or unacknowledge the alert.
3. Add an alert annotation.
4. Clear an alert.
5. (Optional) Update the alert attributes.
6. (Optional) Exempt alerts.

Assign Alerts

When an alert arrives on the Operations Console, you assign someone to resolve the situation that caused it. The steps taken to solve the situation are recorded in the Update History section on the Alert Details tab of the Component Detail pane.

NOTE
This feature is available only if you have appropriate access privileges. For more information, contact your CA SOI administrator.

Follow these steps:

1. Select an alert to assign in the Alerts tab.
2. Click the *set* link next to the Assigned property in the General Information section.
3. Enter an assignee, and press Enter.

Acknowledge or Unacknowledge Alerts

The first step in resolving an alert is acknowledging its existence. Because acknowledgment creates an audit history entry that identifies who acknowledged the alert, you can also use acknowledgment to indicate alert ownership.

NOTE
This feature is available only if you have appropriate access privileges. For more information, contact your CA SOI administrator.

To acknowledge an alert, right-click an alert and select Acknowledge.

NOTE
An alternative is to select one or more alerts and click the icon *Acknowledge selected alerts* on the toolbar. This icon lets you acknowledge multiple alerts at a time.

A checkmark appears in the Acknowledged column.

NOTE
If you acknowledged the wrong alert, you can remove the checkmark by clicking the *Unacknowledge selected alerts* icon.

Create or Modify Alert Annotations

Annotations are comments that can track the steps to resolve the situation that caused an alert.

NOTE
This feature is available only if you have appropriate access privileges. For more information, contact your CA SOI administrator.

For information about annotating multiple alerts at the same time, see Update Alert Attributes.

Follow these steps:

1. Click the alert to annotate.

   NOTE
   If you want a larger detail window, right-click the alert and select Alert Detail.
2. Scroll down to the Annotations section on the Alert Details tab, and click the plus sign icon (+) to open the section.
   A small toolbar and a container for multiple annotations appear.
3. Complete one or both of these actions:
   – Click *Adds a new annotation*

     ,
     enter text in the dialog that opens, and click OK.
   – Select the annotation to modify, click *Modifies the selected annotation*

     ,
     update the text in the dialog that opens, and click OK.

NOTE
You can also print the annotations and export them to a CSV (comma separate values) file. Click the Print or the Export icons in the Annotations section.

Clear an Alert

You clear an alert when you resolve the situation that caused creation of the alert.

NOTE
This feature is available only if you have appropriate access privileges. For more information, contact your CA SOI administrator.

If the clear alert option is available for some alerts but not others, the administrator may have enabled the 'Respect Underlying MDR Clear Alert Setting' option. This option prevents you from clearing alerts in CA SOI that are not clearable in the source domain manager.

After you clear an alert, you can view cleared alert history in the following places for historical analysis:

- In the Cleared Alert History tab of the Component Detail pane when you select a service. This tab displays all cleared alerts that were once associated with the selected service.
- In the Cleared Alert History table of the Contents pane Information tab when you select an alert queue. This tab displays all cleared alerts that were once associated with the selected alert queue. You can also view who cleared the alert in the **ClearedBy** field. The following table describes the mapping of the cleared alerts in the ClearedBy field:

Alerts	Data that are displayed in the ClearedBy field
Escalation Policy	Auto - <Escalation Policy Name>
Connector	Auto - <Connector Name>
Rest\SOAP web service call	Auto - Web Service
Manual	Person Name
Custom Policy	Auto - Customer Policy
Help Desk	Auto - Help Desk
Auto Cleared Service Alarm	Auto Cleared
Reassign Existing Alarm (In case of Managed/Unmanaged)	Auto -Reassign

Follow these steps:

1. Right-click the alert and select Clear.
2. Click OK.
   The alert is cleared and removed from the Alerts tab. Any associated help desk ticket is also closed.

NOTE
ClearedBy feature of the Cleared Alert history tab is not supported for Escalation policy.

Update Alert Attributes

You can manually add values for the following attributes from the Write Alerts dialog:

- Annotations
- Assigned
- Ticket ID

This feature lets you quickly make a note in the alert, specify an assigned technician, or manually link the alert with a corresponding help desk ticket.

If you have integrated CA SOI with a help desk application such as CA Service Desk or BMC Remedy, the Ticket ID attribute should populate automatically for incidents created based on the Create Ticket escalation policy.

> **NOTE**
> This feature is available only if you have appropriate access privileges. For more information, contact your CA SOI administrator.

Follow these steps:

1. Perform one of the following actions:
   – Right-click an alert and select Write Alerts.
   – Select multiple alerts using the Ctrl key, right-click any of the selected alerts, and select Write Alerts.
2. Select the attribute to update in the Attribute drop-down list, enter an attribute value in the Attribute Value field, and click OK.

   > **NOTE**
   > If you selected multiple alerts, all alerts inherit the specified annotation, assignment, or ticket ID.

Exempt or Unexempt Alerts

You can exempt an alert if you do not want the condition of the alert to affect the overall status of a service. You can only exempt infrastructure alerts, not service alerts. When a user exempts an alert, the name of the alert is dimmed to let other users know it is exempt. To exempt or unexempt an alert, use the following feature.

You can manually exempt alerts on the Operations Console. The exempted alerts appear dimmed in the Alerts tab.

Follow these steps:

1. Select a service in the Services Tab or an alert queue in the Alert Queues tab.
2. In the Contents pane Alerts tab, select an alert or press Ctrl/Shift + click to select multiple alerts.
3. Click the Exempt selected alerts

   icon.
   The exempted alerts are dimmed.
   Consider the following items:
   – You can also right-click an alert and exempt the alert with the context menu.
   – To unexempt the alerts, repeat the same steps and click the Unexempt selected alerts

     icon.
   – You can add the Exempt column to the Alerts tab. The column displays Yes or No for Exempt/Unexempt alerts. For more information about adding columns, see Operations Console Customization.
   – The Alert Details tab lets you view and change the Exempt status of the selected alert.

How to Escalate Alerts

Contents

As an operator, you can manually escalate alerts by taking a defined action or sending an email notification.

Alert escalation is the ability to enact some escalating action as a result of an alert. Actions include opening a help desk ticket, running a command, sending an email, and so on.

Depending on your access privileges, you can manually escalate an alert in the following ways:

- Take a defined escalation action
- Send an email notification

Any user can view a help desk ticket associated with an alert, regardless of user access privileges.

Take Action on an Alert

You can take any defined action on alerts that are displayed either on the Services tab or on the Alert Queues tab in the Contents pane.

Follow these steps:

1. Right-click an alert or multiple alerts in the Operations Console Alerts tab and select Take Action.

   NOTE
   If there is already a default action set, the default action performs. To change the default action, change the preference from the Alerts Tab folder in the View, Preferences menu item.
2. Perform one of the following actions:
   – Select an existing escalation policy action.
   – Click Create and create an action in the Escalation Action Editor dialog.
3. (Optional) Select the Use this selection as the default and do not show this dialog again check box. This option hides the dialog in the future and uses the default action.
4. Click OK.
   CA SOI attempts to perform the action. A dialog opens indicating whether the action succeeded or failed. If you try to create a ticket, and another ticket creation is in progress on the alert, an error message appears to prevent you from creating a duplicate.
5. (Optional) Click Show Details to view successful or failure information.
6. Click OK.

Send an Alert Email Notification

You can notify a technician about an alert situation. For example, the technician may need to fix a resource or restart a service. You can send an email that contains the information in the alert.

NOTE
If this feature is unavailable, contact your administrator.

Follow these steps:

1. Right-click an alert and select Mail.

   NOTE
   An alternative is to select one or more alerts and click

   on the toolbar. This icon lets you send information about multiple alerts.
2. Perform one of the following actions:
   – Enter an address in the To: field and other addresses in the CC: field. Only the To: field is required.
   – The From: field is automatically populated with the email address of the logged in user. To change the default email, see Configure Email and Failure Notifications.
   – Enter a subject in the Subject field.
   – Select email or pager in the Template field.
   – (Optional) Edit any text in the body of the message.
3. (Optional) Click Edit to remove one or more alert fields from the message. Select the check box for the fields you want to send and click OK.

4. In the Mail Selected Alerts dialog, click Send.

Work with Alert Tables

Contents

Alerts associated with a resource or a service are organized in alert tables. You can print or export alert tables.

Export a Table of Alerts

You can export the alerts associated with a service to a CSV file. The table is in the same format as the container on the Alerts tab. You can use the file in a spreadsheet or other application that reads comma-separated value files.

> **NOTE**
> You can also export alert annotations, history, and associated queues in the Component Detail pane.

Follow these steps:

1. Expand the tree on the Services or Alert Queues tab in the Navigation pane (if necessary) and select the service or alert queue whose alerts you want to export.
   The alerts for that service or alert queue are displayed in the Contents pane on the Alerts tab.
2. Click

   on the toolbar.
3. Select a location and click Save.

Print a Table of Alerts

You can print the alerts associated with a resource or service in tabular format on a local or network printer. The table is in the same format as the container on the Alerts tab. If one of your printers is Adobe PDF, you can create a PDF file.

Notes:

- You can print the entire table or only selected alerts. If one alert is selected, you can print alert details for it.
- You can print alert annotations, history, and associated queues in the Component Detail pane.

Follow these steps:

1. Right-click any alert and select Print.

   > **NOTE**
   > An alternative is selecting File, Print, and then selecting *Alerts* from the drop-down list on the dialog that opens.
2. Select printer options, if necessary, and click OK.

How to View Alert Queues

Contents

As an operator, you can view alert queues to which an administrator has assigned you access.

Alert queues are user-defined alert groups. CA SOI auto-assigns alerts to a particular alert queue based on user-defined policy, which can include alert content and associated CIs.

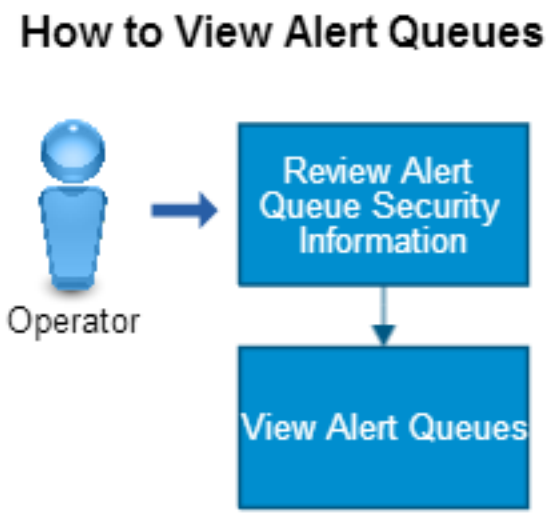
Alert queues let you group alerts as they come in based on specific criteria to monitor the status of your infrastructure more efficiently. You can add global and non-global escalation policies to alert queues to take a specified action automatically on alerts that come into a queue.

For example, consider a company with engineers responsible for different aspects of the infrastructure, such as networks, systems, and databases. Without defined queues, alerts from all integrated domain managers appear in one consolidated view on the Alert Queues tab. The administrator can define queues by domain (such as Network Alerts or Database Alerts). With organized queues, engineers can quickly find and resolve their alerts. Additional queues could be defined based on other alert categories, such as severity, assignment status, and description to enable an optimized unified alert management system.

Services provide a similar organizational function as alert queues at a higher level with the additional benefit of resource topology and impact analysis. Defining alert queues is less intensive than modeling services, and they can simplify alert management as you make the transition to a service-oriented management paradigm. Alert queues can also remain useful in an environment with services defined to provide a supplemental management perspective outside of services. For example, you can define a queue for alerts that have not been acknowledged or a queue for alerts from the same source domain manager.

Use this scenario to guide you through the process:

Figure 53: how to view alert queues



1. Review alert queue and security information.
2. View your alert queues.

Review Alert Queue Security Information

The following topics explain how the administrator provides you access to alert queues.

Alert Queues and Security

Your CA SOI administrator assigns you to user groups. The administrator sets user group access to specific CA SOI features, services, alert queues, customers, and so on. Consider the following items when viewing alert queues:

- You can view all nonservice-impacting alerts and they can show in your alert queues. Because non service-impacting alerts are not associated with a service, they cannot be restricted with access privileges.
- You can view alert queues only to which you have access privileges.
- You can view the alerts in alert queues only to which you have access privileges.
- You can view all the alerts of a service regardless of the access privileges.
- You can edit alert queues for which you have access privileges. Only administrators can edit the Default queue.
- You can only delete alert queues that you created.

Example User Group Access to Alert Queues

In this example, we have the following data in CA SOI:

Services: Sales, Finance, Operations

> **NOTE**
> Because service privileges also dictate the services that appear in a particular alert queue, this example also includes the service access settings.

Alert Queues: Database Alerts, Critical Alerts

> **NOTE**
> For this example, the alert queue names indicate the type of alerts that each alert queue is configured to show. For example, if the Sales service has a critical alert, the Sales service appears in the Critical Alert queue (assuming the User Group has access privileges for the Sales service.)

User Groups: Group1, Group2, Group3, Admin

The following table shows the User Groups and their access to available services and alert queues:

User Group	Service Access	Alert Queue Access
Group1	Sales, Operations	Database Alerts
Group2	Finance, Operations	Critical Alerts
Group3	Operations	Database Alerts, Critical Alerts
Admin	All Services	All Alert Queues

The following table shows the User Groups and what they see in CA SOI based on their service and alert queue access:

User Group	Sees on the Services Tab	Sees on the Alert Queues Tab
Group1	Sales, Operations	Database Alerts queue with database alerts impacting to Sales and Operations services and all unmanaged database alerts.
Group2	Finance, Operations	Critical Alerts queue with critical alerts related to the Finance and Operations services and all unmanaged critical alerts only.
Group3	Operations	Database Alerts and Critical Alerts queues with critical alerts related to the Operations service and unmanaged critical database alerts only.

Admin	All Services	All alert queues with all managed and unmanaged alerts.

View Alert Queues

You can display the alerts in a selected alert queue and view detailed information about a selected alert.

Follow these steps:

1. Start the Operations Console and click the Alerts Queues tab in the Navigation pane.
   The alerts queues for which you have access privileges display. The columns to the right of the queue name display the number of alerts of each severity in the queue and the total number of alerts in the queue. The alert queue icon color indicates the highest severity of any alert in the queue.
2. Select an alert queue.
   The alerts in the selected alert queue displays in the Contents pane. The Contents label displays the currently selected queue.
3. Select the Information tab in the Contents pane.
   The Information tab displays queue details, such as description, criteria, priority, associated escalation policies, and cleared alerts that once belonged to the queue.
4. Return to the Alerts tab, and click an alert to display detailed information about that alert.
   For more information about viewing alert details, see View Alerts, Alert Details, and Extended Information.

How to View Customers and Customer Details

Contents

As an operator, you can view customers and see how alerts impact the customers.

A *customer* in CA SOI is any consumer of a managed service. The CA SOI administrator creates customers and associates them with service models to see the impact of service degradation on the customer.

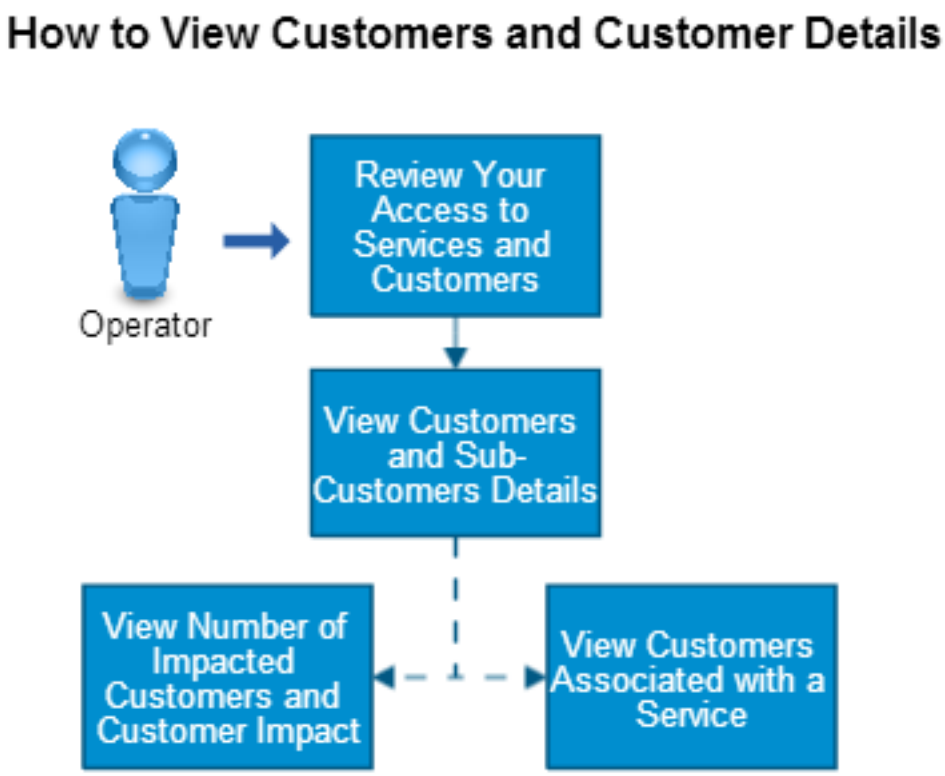
Your CA SOI administrator assigns you a customer and sets the specific services that you can view for that customer. Therefore, the following access privileges limit your service access for each customer:

- Your user group access to services.
- Your customer access to services.

If you need access to additional services or customers, contact your administrator.

Use this scenario to guide you through the process:

Figure 54: how to view customers



1. Review the information about your access to services and customers.
2. View customers and sub-customer details.
3. (Optional) View the number of impacted customers and the customer impact.
4. (Optional) View the customers that are associated with a service.

Your Access to Services and Customers

Your CA SOI administrator assigns you to user groups. The administrator sets user group access to specific CA SOI features, services, alert queues, customers, and so on.

If you require access to a specific service or customer, contact your administrator.

View Customer and Sub-Customer Details

You can view information about customers and sub-customers, including list of sub-customers, associated services, and associated service alerts.

Follow these steps:

1. Open the Operations Console and click the Customers tab in the Navigation pane.
   A list of all customers displays. The columns to the right of the customer name display the number of alerts of each severity for the customer and the total number of alerts for the customer. These alerts are the alerts that are impacting the services that are assigned to the customer.

If a customer is associated with a service and an alert is generated that impacts that service, the alert also impacts the associated customer. An alert can impact multiple customers for the following reasons:
- Because multiple customers are associated with a single service.
- Because an alert impacts multiple services and each service has an associated customer.

2. View the customer tree icon color for any customer to see the overall customer health. Of all the services that are assigned to the customer, the service with the worst health represents the customer health. The customer tree icon color, therefore, shows the color that is based on that service's health.
3. Select a customer (or sub-customer) in the Customers tree and do any of the following tasks:
   - Click the Alerts tab to view all the alerts (from the services that are assigned to the selected customer) for the customer. This tab includes information about alert severity, category, summary, count of impacted customers, and so on.

     NOTE
     You can also display the number of impacted customers by adding the # Impacted Customers column.

     When you remove or add a service to a customer, the associated alert list is changed accordingly.

     NOTE
     The alert list for the parent customer shows the aggregate of all alerts from all of its child customers.
   - Click the Services tab in the Contents pane to view a list of all services that are associated with the customer. This tab includes information about the service name, health, risk, granularity, and so on.
   - Click the List tab in the Contents pane to view a list of sub-customers. This tab includes information about the all the sub-customers available under the selected parent customer. The tab displays the customer name, ID, priority, impact (or customer health), quality, risk, and description.

     NOTE
     Of all the services that are assigned to the customer, the service with the worst health represents the customer health. Similarly, the service with the worst quality represents the customer quality, and the service with the maximum risk represents the customer risk.
   - Click the Information tab in the Contents pane to view general information about customers. This tab includes general information about the customer: name, ID, and priority. This tab also includes current metric information: health, quality impact, risk, and priority. The Security pane shows the user groups that have access to the customer.
   - Click the Customer Impact tab to view the customer impact level. This tab is available to user groups with access privileges only.
4. Review the information. For the Services and Alerts tabs, you can use the additional tabs in the Component Detail pane to get detailed information.

View Number of Impacted Customers and Customer Impact

You can view the number of impacted customers due to an alert and the customer impact.

Follow these steps:

1. Open the Operations Console and click either the Services tab or the Alert Queues tab.
2. Select a service or an alert queue.
3. The Contents pane displays the column "# Impacted Customers" which indicates the number of customers that the alert impacts.
4. Select an alert and select the Customer Impact tab in the Component Details pane.

View Customers Associated with a Service

You can view all the customers that are associated with a specific service. This information helps you analyze the impact that a particular service has on different customers. When an administrator associates or removes a service from a customer, CA SOI updates the information in the Customers tab accordingly.

Follow these steps:

1. Open the Operations Console and click the Services tab.
2. Select a service for which you want to see the associated customers.
3. Click the Customers tab in the Contents pane.
   All customers that are associated with the selected service display in the pane. The pane also provides detailed information about the related customers; for example, name, ID, description, priority, and so on.

CA SOI Dashboard

This section describes how operators monitor the Dashboard to manage service status.

Access the Dashboard on a PC

As an administrator or an operator, you access the Dashboard on a PC, log in to CA SOI and click the Dashboard tab.

Dashboard Metrics

Contents

This section describes some common terms as they relate to the CA SOI dashboard.

Risk

Risk indicates the likelihood of delivering the quality of service that is required to support the overall business objectives. The highest propagated impact of an associated risk alert determines the service risk value.

If an alert has no defined type, it is a risk alert by default.

In an IT infrastructure, risk is the measure of how much the problems that are currently associated with an IT element impacts the likelihood that the required service quality levels are delivered. In other words, as IT elements encounter problems, the risk of service quality degradation increases.

Due to typical IT risk-mitigation measures such as redundancy, fault tolerance, and high availability, faults in the IT infrastructure may not directly result in service quality degradation. However, these faults do increase the *risk* of delivering the required service quality.

For example, consider a server farm that has 12 servers that support an online application:

- If three of the servers are unavailable, the risk of a service outage can be considered Slight.
- If six of the servers are unavailable, the risk of a service outage can be considered Moderate.
- If ten of the servers are unavailable, the risk of a service outage can be considered Severe.

The risk settings are Down, Severe, Moderate, Slight, None, and Unknown.

Quality

Quality indicates the level of excellence that consumers of an IT service experience, whether the consumers are customers, end users, or other IT services. The levels of quality are Operational, Slightly Degraded, Moderately Degraded, Severely Degraded, Down, and Unknown. The highest propagated impact of an associated quality alert determines the service quality value.

For example, the transaction time that is associated with completing a user task, such as logging in to the system, is a quality metric. Quality metrics are typically associated with a threshold. For example, if transaction time exceeds 5 seconds, the service quality could be considered to be Moderately Degraded.

Health

Health is a reflection of the worst state that of either Quality or Risk. Health provides a high-level summary of the service health according to those metrics.

For example, if the Quality is Operational and the Risk is Severe, the service health shows a Critical status. Severe is the worst state of Quality and Risk.

Availability

Availability is an abstracted measure of service uptime and downtime that is based on the health of the service.

The SA Manager measures service availability based on the service health:

Service Health	Service Availability
Normal|Minor|Major	Up
Critical|Down|Unknown	Down

For example, a severely degraded service has Down status even though the service is partially active. If the service maintained the Down status for 12 of the last 24 hours, availability would show as 50 percent for that period.

Priority

Priority indicates the importance of a service to the business.

Priority determines the following orders:

- The order in which the Dashboards display all services.
- The order in which escalation policies run when they affect more than one service.

SLA

A *service-level agreement* (SLA) is a contract that specifies the service expectations of internal or external customers. An example is the downtime that is acceptable for various resources.

View Service Status and Details

Contents

As an administrator or an operator, you view the status of services and the service details.

The Dashboard tab contains the Services table. The table displays information about the services that CA SOI is monitoring and managing. Below the table, charts display additional detail about a selected service.

Follow these steps:

1. Click the Dashboard tab.
   By default, the Services table includes the following columns:

   NOTE
   You can resize the table height to display more or less services per page; however, you cannot adjust the width.

   – **Services**
     Displays the name of the service. If the service name includes a number in parentheses next to its name, that number represents the count of subservices in the next level only (not subservices of those subservices).

Use the Services column filter
( )
to filter data in the table.

– **Priority**
Displays the priority setting of the service. The priority settings are Critical, High, Medium, Low, None, and Unspecified.
Use the Priority column filter
( )
to filter data in the table.

– **Current SLA**
Specifies if a service level agreement (SLA) is defined for the corresponding service. The column is blank if no SLA is defined. If an SLA is defined, the column displays one of the following icons:
- Green checkThe service is compliant for the current SLA period.
- Red XThe service is not compliant; it has been violated for the current SLA period.
- Red circle with a slashThe SLA is inactive.

Use the Current SLA column filter
( )
to filter data in the table.

– **Health**
Displays the health rating of the service. Each icon represents the health metric as follows:
- If the service is in the Production Operational Mode, the color-coded health settings are Down (burgundy), Critical (red), Major (orange), Minor (yellow), and Normal (green).
- If the service is in Maintenance Operational Mode, the health rating can be one of the production modes (Down, Critical, Major, Minor, or Normal) or Unknown.
- If the service is in Testing Operational Mode, the health rating is set to Unknown.

Use the Services column filter
( )
to filter data in the table.

– **Quality**
Displays the quality rating of the service. The color-coded quality settings are Down (burgundy), Severely Degraded (red), Moderately Degraded (orange), Slightly Degraded (yellow), and Operational (green).
Use the Quality column filter
( )
to filter data in the table.

– **Risk**
Displays the risk that is associated with the service. The color-coded risk settings are Down (burgundy), Severe (red), Moderate (orange), Slight (yellow), and None (green).
Use the Risk column filter
( )
to filter data in the table.

– **Availability [24 hours]**
Displays the availability of the service, which is expressed as a percentage that is calculated over the past 24 hours.
Note: An asterisk (*) next to an Availability value indicates that the value does not represent a complete set of data, which is a full 24 hour period.
Use the Availability [24 hours] column filter
( )
to filter data in the table.

– **Operational Mode**
Displays the mode of the service. The modes are Testing, Maintenance, and Production.

Use the Operational Mode column filter
( )
to filter data in the table.

– **Launch To**
Contains an Action button and drop-down menu that allow you to open the Operations Console or the corresponding domain manager application for the associated service.
Use the Launch To column filter
( )
to filter the data in the table.

2. (Optional) Click the column heading to switch the sort order between ascending and descending for that column. Ctrl + click to select multiple columns.
3. (Optional) Rearrange the display order of the columns by dragging and dropping appropriate columns. For example, if you want the Priority column to appear before the Services column, you can drag-and-drop the Priority column before the Services column.

   NOTE
   If you rearrange the column order, CA SOI does not save the order. To change the column order permanently, see Customize the Services Table.
4. (Optional) Enter a search string in the Find field. If a service name contains the entered string, the string is underlined in the Services column. However, CA SOI does not remove any entries. Click the arrows (< and >) to move among underlined services.
   The sections that follow provide information about the Details panel and the Quality, Risk, Availability, Alerts, and SLA tabs. These tabs show various charts for the selected service. You can resize these charts as necessary.
5. (Optional) Double-click a service row to display the associated service detail charts in carousel mode.

View Service Information

To view details about a service select a service in the Services tab and click the Information tab in the Component Detail pane.

The Information tab displays the following service information:

NOTE
Depending on your user group privileges, you may not be able to edit all properties in this tab.

- **Service Name**
  Displays the name of the service at the top of the tab.
- **General Information**
  Displays a list of the following basic service information:
  – **Health**
  Displays the service health state.
  – **Quality Impact**
  Displays the impact of infrastructure alerts on service quality.
  – **Risk**
  Displays the risk of infrastructure alerts on service quality.
  – **Family**
  Indicates the CI class family that the alert is associated with.
  – **Operational Mode**
  Displays the current operational mode, either Production or Maintenance.
  – **Priority**
  Displays the service priority value.
  – **Location**

  Displays the service location.
  - **Description**
    Displays the service description. Click Set to enter a new description.
- **Service Level Agreements**
  Displays the SLA associated with the service with information such as name, status, and description.
- **Maintenance Schedules**
  Displays maintenance schedules associated with the service. From this table, you can edit an existing maintenance schedule.
- **Connectors**
  Displays the connectors managing the service CIs.

Column Filters

Each column in the Services table includes a filter icon
( )
that lets you find and display specific information. Applied filters have a red icon
( )
to distinguish between columns with applied and unapplied filters. You can perform the following actions with filters:

- Click an unapplied filter icon
  ( ).
  Depending on the column, you enter a Search term, adjust a slider, or select check boxes to filter the data.
- Further refine the information by applying filters to multiple columns.
- Prioritize the filter order by Ctrl + left-clicking columns. A number appears in each column to show the filter priority. For example, you can first apply a filter to the Service column and can then enter the Search term "Alpha." You apply a second filter to the Health column and clear all check boxes except the burgundy icon (which indicates a Health state of Down) red icon (Critical). These multiple filters result in the Dashboard displaying only those services that contain the string "Alpha" *and* that have a Down or Critical Health status.
- Switch between ascending
  ( )
  and descending
  ( )
  order by left-clicking a column.
- Clear a filter by clicking an applied filter icon
  ( )
  and clearing any search fields or selecting all checkboxes.

> **NOTE**
> Sorting changes are not permanent; however, you can save filters. For more information, see Customize the Services Table.

View Risk Details

The Risk tab in the Service History portlet displays the risk summary as a pie chart and an area chart for the selected service.

Follow these steps:

1. Select a service entry in the Services portlet.
   The service details appear in the Service History portlet.
2. Click the Risk tab.
   Note the following items in the Risk pie and area charts:

- The area chart shows Risk as a percentage, with the higher number representing a higher risk of the associated service being unavailable.
- The pie chart shows color-coded sections that correspond with the number of hours or days the risk is unchanged.
- The pie chart shows the following periods:
  - The number of hours equaling 24 when the Last 24 Hours option is selected.
  - The number of days equaling 30 when Last 30 Days option is selected.
  - The number of hours equaling 168 when the Last 7 Days option is selected. The conversion of days to hours allows information to be displayed with greater detail and avoids the need to display fractions of days.
- The pie chart shows the time that the service was in Maintenance mode.
- The pie chart displays Unknown if the data is not available for the entire selected time period. For example, if you select Last 30 Days and you have only 20 days of data, the Unknown section is displayed with a label showing 10.00.

View Quality Details

The Quality tab in the Service History portlet displays the quality summary as a pie chart and an area chart for the selected service.

Follow these steps:

1. Select a service entry in the Services table.
   The service details appear in the Service History portlet.
2. (Optional) Click the Quality tab.
   Note the following items in the pie and area charts on this tab and the other tabs:
   - The charts can take a short time to display.
   - You can mouseover the sections of the pie chart or points on the area chart to display summary details for the selected day or hour.
   - You can select the time period that displays from the drop-down list for each chart.
   - If the Report server is configured, you can click the chart to open the reporting interface in the context of the selected service and time period.
   - The charts are color-coded. The pie chart shows the following states: Down (burgundy), Severely Degraded (red), Moderately Degraded (orange), Slightly Degraded (yellow), and Operational (green). The area chart contains a legend that describes the color-coding.
   - The summary shows the status for the last full hour or day depending on which time period is selected. For example, if you are using the Last 24 Hours mode, and mouseover 2 PM in the area chart, the summary shows the period from 1:30 PM until 2 PM.
   - The area chart only displays the times when Quality, Risk, or Availability states are something other than Normal or None. That is, the area chart displays data in the following situations:
     - Quality is Minor, Major, Critical, Down, Maintenance, or Unknown.
     - Risk is Slight, Moderate, Severe, Down, Maintenance, or Unknown.
     - Availability is Down, Maintenance, or Unknown.
   - The pie chart and the area chart treat Unknown time differently.
     For example, if you select Last 24 Hours, the pie chart shows a full 24 hours. If there are only 23 hours of data in the database, the pie chart classifies the 24th hour as Unknown as the data is not available. CA SOI does *not* obtain the Unknown state from the database.
     The area chart does not infer data; the chart reflects only the state values, other than Normal, that are present in the database. If the service was in a normal state for all 23 hours, the area chart is empty. Additionally, if there is no

data in the database with an actual state of Unknown, there is no Unknown time displayed in the chart. The same behavior occurs when Last 7 Days or Last 30 Days is selected.

– The area chart displays the data for a specific time period. For example, if the Quality data between 2 PM and 2:30 PM is Major 80% of the time, and from 2:30 PM-3 PM it is also Major 80% of the time, the area chart starts at 2:30 PM with 50% and ends at 3 PM with 80%.

> **NOTE**
> As another example, if the service is placed in Maintenance mode between 2AM and 4AM, the area chart for Maintenance data starts drawing the curve from 2 AM and goes up to 100% Maintenance at 2:30 AM. This is because the maintenance data between 3:30 AM and 4 AM is 100% and from 4 PM to 4:30 PM it is 0%. The area chart starts to curve down at 4 PM, and reads 0% at 4:30 PM. You can mouseover each data point to display the details.

View Availability Details

The Availability tab displays the availability summary as a pie chart and an area chart for the selected service.

Follow these steps:

1. Select a service entry in the Services table.
2. Click the Availability tab.
   Note the following items in the Availability pie and area charts:
   – The area chart shows Availability as either 100% (available) or 0% (not available).
   – The pie chart shows the actual state: Critical status and Down status indicate that the service is not available. The other states indicate that the service is available.
   – The pie chart shows the time that the service was in Maintenance mode.
   – The pie chart displays Unknown if the data is not available for the entire selected time period. For example, if you select Last 24 Hours and you have only 20 hours of data, the Unknown section is displayed with a label showing 4.00.

View Alert Details

The Alerts tab displays the service alerts and direct cause alerts that are associated with the selected service.

Follow these steps:

1. Select a service entry in the Services table.
2. Click the Alerts tab.
   The following types of alerts display:
   – **Service Alerts**
   Displays the alert condition that CA SOI generates based on analysis of a service model that it is monitoring. Service alerts result when the condition of one or more configuration items combines to impact the overall quality or risk that is associated with the service. The policy that is defined for that service model determines how configuration item alert conditions impact other configuration items and the overall service.
   If a help desk ticket exists, the Last Alert timestamp and summary are hyperlinked to the ticket. If the help desk integration is not configured as described in Configure Help Desk Integration, the number of tickets is 0 (zero). Click the link to open the ticket in the corresponding help desk product.
   – **Direct Cause Alerts**
   Displays the following information for the corresponding configuration item:
   • Alert category and icon. The valid categories are defined in USM, such as Activity, Application, Cloud, Database, Issue, Network, Relationship, Service, System, and Other.

Note: The Other icon represents New categories.

- Number of alerts for the category
- Number of open alerts for the category and the number of associated help desk tickets
- Last alert in the category (the timestamp and summary are hyperlinked to the corresponding help desk ticket, clicking it opens the ticket in the corresponding help desk product)

Direct cause alerts are assigned an impact value that is calculated based on the fault condition seriousness and the CI importance to the services or subservices it supports. The categories in the list are sorted based on the severity of the open alerts (Critical comes before Major, Major before Minor).

View SLA Details

The SLA tab displays the current SLA status as a pie chart and the SLA History as a bar chart.

Follow these steps:

1. Select a service entry in the Services table.
2. Click the SLA tab.
   Note the following items in the SLA charts:
   – The charts display the following color-coded states: Up (green), Unplanned (red), Maintenance (brown), Unknown (gray).
   – Unplanned is the total of outage and violation time (it does not include maintenance time). Violation is the time after the threshold has been reached.
   – The pie chart pane also includes the following information:
     - **SLA Current Status:** Displays date and time that the SLA was last calculated.
     - **Type:** Displays the type and state of SLA. SLAs can be based on Availability, Health, Quality, or Risk.
     - **Threshold:** Displays the actual time (in minutes and seconds) or percentage of time that an SLA has been violated.
     - **Description:** Displays the description that was entered when the SLA was created in the Operations Console.
     - **Violation Time:** Displays the time that exceeds the threshold. For example, if the threshold is 300 seconds and outage time is 800 seconds, then the violation time is 500 seconds.
     - **Updated:** Displays the time that the chart was last updated. Midnight is represented as 00:00:00.
   – You can select Line Chart from the drop-down list to display the SLA History in a line chart instead of the default bar chart.
   – The bar chart can show a single SLA, but the line chart requires at least two SLA data points to draw the line.
   – You can mouseover the sections of the bar chart or data points on the line chart to display summary details for the corresponding day.
   – You can click any of the charts to generate an SLA History report for the selected service. Click the pie chart to generate a report for the last 24 hours. Click the bar or line chart to generate a report for the SLA time period.
   – The SLA charts display the date using the yyyy/mm/dd format, for example, 2009/05/08 is May 8th, 2009. The reports that are generated from the SLA charts use the mm/dd/yyyy format.

Run Reports from the Dashboard

As an administrator or an operator, you can generate JasperReports Server reports from the Dashboard.

CABI JasperReports Server users can can generate the following reports from the CA SOI Dashboard:

- SLA Current Status
- SLA History
- Quality Summary Status
- Quality Status
- Risk Summary Status
- Risk Status
- Availability Summary Status
- Availability Status

NOTE

- You can also run various additional reports in CABI JasperReports Server. For more information, see Generating Reports in CABI JasperReports Server.

Follow these steps:

1. Log in to the CA SOI interface, and click the Dashboard tab.
2. Select a service entry, on which you want to report, in the Services table.
   The Details of Selected Service pane for the selected service opens and displays the Quality tab.
3. (Optional) Click the Risk or Availability tab to generate a report of that type.
4. Perform one of the following actions:
   – To generate a Summary Status report for the current tab, select the time period on which to report (Last 24 Hours, Last 7 Days, Last 30 Days), then click the pie chart.
   – To generate a Status report for the tab you are on, click the data point in the area chart for the time period on which to report.

View Services in Google Map

Contents

As an administrator or an operator, use Google Maps API to view services that CA SOI monitors. You can associate a service with a location and then use Google Maps to view it based on the specified location. If you enter a city, Google Maps shows the service in the center of the city. If you enter a street address (for example, 710 Ashbury, San Francisco, CA) Google Maps shows the service at that location if Google Mapss can validate the address. You can include additional information (for example, Building 12 or Floor 3) as long as Google Mapss can resolve it as a valid address. You can view the Mapsped CA SOI services from the Dashboard.

After installing CA SOI, the Maps View link appears on the CA SOI dashboard. Set the service location in Operation Console, click the Maps View link to open Google Maps from the CA SOI UI, and view the Mapsped services.

Prerequisite:

- Ensure that you generate an API Key to view the location of a service on the Google Maps. Click here to learn about obtaining the Google Mapss JavaScript API key.
  If you obtain a free key, you might experience slowness on loading high numbers of services. You can purchase a premium key to speed up loading of services. However, there are limitations on usage of Google API. For more information about Google Maps usage limit, see Google Mapss Geocoding API Usage Limits.

Follow these steps in the system where CA SOI UI server is installed:

1. Browse to <SOI_Home>/CA/SOI/SamUI/webapps/sam and edit googleAPI.config.
2. Specify the API key and save the file.
3. Open the Operations Console and click a service in the Navigation pane on the left side.
   The alerts for that service appear in the Contents pane, and general service information appears in the Information tab of the Component Detail pane.

4. Click **set** next to Location in the General Information section of the Information tab.
5. Type a location and press Enter. The location can be as broad as a country, or as specific as street address with city and state.
   The location value appears on the Information tab.
6. Open CA SOI UI.
   The Maps View link appears on the top right of the page.
7. Click the **Maps View** link.

   NOTE

   If *"Error - Invalid API key. See browser console logs for more details*" error appears when you click the **Maps View** link, try regenerating the API key.

   Google Maps appears with the following items:
   – Color-coded push-pin icons (known as *placemarks* in Google Maps) on the Maps for each CA SOI service. The placemarks appear on the Maps according to the location you had set.
   – Color-coded filter on the left to filter service based on the health. Click a check box to filter the services.
     - Green lists services with Normal health.
     - Yellow lists services with Minor health.
     - Orange lists services with Major health.
     - Red lists services with Critical health.
     - Burgundy lists services with Down health.
     - Gray lists services with Unknown health.
   – Summary of the services at the lower-left corner. Normal indicates the services in normal health. Degraded indicates all services that are not in normal health.
8. To view the Maps in the Night view, click **Night View** on the top left corner of the Google Maps.

   NOTE

   To navigate to the default view, click Default View or refresh the Google Maps page.
9. Click a placemark to view the details of the services and zoom to the most detailed view of the location available.
10. Type the service name in the **Search Services** text box to view the specific service.

Improve the Performance of Google Mapss API

You can change the number of addresses that are sent or interval to load the Maps so that the performance of the Google Mapss API is improved.

Follow these steps:

1. Browse to <SOI_Home>/CA/SOI/SamUI/webapps/sam and edit googleAPI.config.
2. Change the following parameters, if needed:

```
# Interval (in milliseconds) at which Maps View will load the locations data from UI
 Server
refresh_interval=30000
# Interval (in milliseconds) at which Maps View will load the latitude/longitude data
 from Google
wait_time=5000
# Number of addresses sent to Google per request
addresses_per_request=5
```

3. Save the file.

Restrictions in Viewing CIs in Google Maps

In the following situations, you cannot view the CIs in the Maps:

- By default, an administrator can access the Maps View link and set privileges to users. If you cannot view the services in Google Maps, check your privileges or contact your administrator.
- If no domain information is provided in the URL or if you are not registered to use Google Mapss, 'Oops! Something went wrong' error message appears.
- If the Google Maps is opened in a browser and if CA SOI services stopped working for some reason, an error message appears on the Mapss.
- In certain countries or regions, such as Crimea, Cuba, Iran, North Korea, Sudan, and Syria, Google had restricted its services. In the restricted regions, you cannot Maps or view the services.

Display Service Detail Charts in Carousel Mode

As an administrator or an operator, you display service detail charts in carousel mode. Carousel mode is an interactive graphical display that allows you to rotate among available charts and generate reports.

Follow these steps:

1. Double-click a service row under any of the following headings: Current SLA, Quality, Risk, or Availability.
   The service detail charts for the selected service display in carousel mode.
2. Perform any of the following actions to rotate among charts:
   – Select a chart from the drop-down list above the chart carousel.
   – Double-click a chart.
   – Use the scroll bar below the chart carousel.
3. (Optional) Select the chart time period from the drop-down list above the chart carousel.
4. (Optional) If available, select a chart type (Bar or Line) from the chart drop-down list.
5. (Optional) Double-click a chart to generate the associated detail report.

   NOTE
   The system administrator must configure the report server before users can launch the reports.

The reports that are generated are the same as the reports you generate from InfoView. For report descriptions, see the report list.

CA SOI Mobile Dashboard

This section describes how operators monitor the Mobile Dashboard to manage service status and escalate alerts.

Access the Dashboard on a Mobile Device

As an administrator or an operator, you can access a version of the Dashboard that is designed for mobile devices. On the Mobile Dashboard, you can view and perform a subset of actions available on the PC version of the Dashboard and the Operations Console.

NOTE
For a list of supported mobile devices, see Mobile Device Support .

Follow these steps:

1. Open a web browser and enter the following URL:
   ```
   https://<UI server>:<port>/mobile
   ```

NOTE
You are automatically redirected to an https connection if you enter http. You can change this behavior in the mobile dashboard configuration file.

– *UI server*
Defines the name of the system where the UI Server is installed.
– *port*
Defines the port on which the specified server listens.

NOTE
The default port is 7070 for a non-SSL connection, and 7403 for an SSL connection.

2. Enter your login credentials and tap Login.

NOTE
Your login credentials are the same as your CA SOI credentials. The samuser administrator account cannot be used here.

Navigate the Mobile Dashboard

Contents

As an administrator or an operator, you can view Dashboard data on a mobile device.

The Mobile Dashboard provides three tabs at the top of the home page that provide access to services, alert queues, and customers.

NOTE
You can tap the navigation bar at the top of any page to see and navigate the path of the current page location. The icon to the left of the navigation bar indicates the path through which you came to your current location (through a Services page, an Alerts page, or a Customers page). A Home icon indicates that you are on a home page.

Tap a tab to access the following information:

- Services
Displays the Services home page which contains a list of all services with status indicators that provide metrics overviews.

NOTE
By default, this page is automatically displayed when you first log in.

- Alert Queues
Displays the Alert Queues home page which contains a list of the alert queues to which you have access.
- Customers
Displays the Customers home page which contains a list of customers with status indicators that provide an overview of associated services for each customer.

View Services

The Services home page displays a list of your services and a status overview for each service.

Follow these steps:

1. Access the Mobile Dashboard.
2. You can perform the following actions on this page:

- Tap the plus sign (+) next to a service to show the parent and child services of that service. Tap All Services to return to the original list.
- Tap a service or child service to display the Alerts for the selected item.
- Tap the yellow panel containing the down arrow to change the metrics that are displayed, change the display order to ascending or descending, or set the page to Favorite services only.
- Tap Alert Queues to display the available alert queues.
- Tap Customers to display customers.

Manage Favorite Services

You can manage services using a Favorite Services list, so that you can quickly find the services that matter to you. After you define favorite services, you can configure the Services home page to display only your favorites.

Follow these steps:

1. Access the Mobile Dashboard.
2. Tap the Menu button in the upper-right corner.
3. Tap Favorite Services.

   NOTE
   An active star

   indicates services on your Favorite Services list.
4. You can perform the following actions on this page to update the services on your favorites list:
   - Tap the gray box to the right of a service to add or remove it from your favorites.
   - Use the Service filter slide bar to switch the view from All to Favorite to display all services or favorites only, on this page.
     Note: The Favorite view is more effective for removing favorites from your Favorite Services list.

View Alert Queues

The Alert Queues home page displays the alert queues to which you have access. You can view the alerts in a selected alert queue by tapping it.

Follow these steps:

1. Access the Mobile Dashboard.
2. Tap Alert Queues.
   The number to the right of an alert queue name indicates the alert count in that queue. The count can include alerts that you do not have user access privileges to view. Therefore, the alert count can be higher than the actual alerts that display in the alert queue.
3. You can perform the following actions on this page:
   - Tap the Services tab to view services.
   - Tap the Customers tab to view customers.
   - Tap the yellow panel containing the down arrow to change the sort order (by name or by alert number) or the sort direction (ascending or descending).
   - Tap an alert queue to display the alerts in the alert queue, then tap an alert to display the available actions for the selected alert. These alerts include alerts affecting a service to which you have user access privileges and alerts that do not impact any services. If an alert affects a service that you do not have user access privileges to view, the alert does not appear in the list.

View Customers

The Customers home page displays the customers to which you have access. You can also view the overall health, risk, and quality of the services for the selected customers. By default, CA SOI orders customers by health in ascending order.

Follow these steps:

1. Access the Mobile Dashboard.
2. Tap Customers.
   The indicator to the right of a customer name displays, by default, the health for services that are associated with the customer.
3. You can perform the following actions on this page:
   – Tap a customer name to display a page where you can see the alerts and services associated with the customer, customer details and hierarchy (parent and child customer).
   – Tap the plus sign (+) next to any customer to see its subcustomers.
   – Tap the yellow panel containing the down arrow to open the configuration page. On the configuration page you can change the sort order (by health, risk, or quality) or the sort direction (ascending or descending).

     NOTE
     When you change the sort order for health, risk, or quality, the indicator next to the customer changes to display the associated service status.
   – Tap the Services tab to view services.
   – Tap the Alert Queues tab to display the available alert queues.

View Service Metrics

The Metrics page displays the health, risk, availability, quality, and SLA (if defined) for the selected service.

Follow these steps:

1. Access the Mobile Dashboard.
2. Tap the Status Indicator to the right of a given service.
3. Tap Metrics.
4. You can perform the following actions on this page:
   – Tap Details to display the USM properties for the service.
   – Tap Alerts to display active alerts for the service.
   – Tap Hierarchy to display the parent and child services of the service.

View Service USM Properties

The Details page displays the USM properties for the selected service.

Follow these steps:

1. Access the Mobile Dashboard.
2. Tap the Status Indicator to the right of a given service.
3. Tap Details.
4. You can perform the following actions on this page:
   – Tap Metrics to display the health, quality, risk, and availability for the selected service.
   – Tap Alerts to display active alerts for the service.
   – Tap Hierarchy to display the parent and child services of the service.

     NOTE
     If this service does not have any parent or child services, the Hierarchy tab does not appear.

View Service Hierarchy

You can view the parent and child services of a selected service.

> **NOTE**
> The hierarchy is for services only, so there are no CIs provided, only parent and child services. Also, if this service does not have any parent or child services, this page is not available and the Hierarchy tab does not appear.

Follow these steps:

1. Access the Mobile Dashboard.
2. Tap the Status Indicator to the right of a given service.
3. Tap Hierarchy.
4. You can perform the following actions on this page:
   - Tap a parent or child service to select that service and display its parent and child services.
   - Tap Details to display the USM properties for the service.
   - Tap Alerts to display active alerts for the service.
   - Tap Metrics to display the health, quality, risk, and availability for the selected service.

View Service Alerts

The Alerts page displays active alerts for the selected service. You can also access alerts by viewing an alert queue.

Follow these steps:

1. Access the Mobile Dashboard.
2. Tap the Status Indicator to the right of a given service.
   Each alert displays a colored icon that indicates the severity of the alert. If an alert displays a blue check mark, the alert is acknowledged.
   - **Direct Alerts**
     Lists the alerts that are raised on the service itself.
   - **Affecting Alerts**
     Lists the alerts that are raised on the CIs and child services in the service.
3. You can perform the following actions on this page:
   - Tap Metrics to display the health, quality, risk, and availability for the selected service.
   - Tap Details to display the USM properties for the service.
   - Tap Hierarchy to display the parent and child services of the service.

     > **NOTE**
     > If this service does not have any parent or child services, the Hierarchy tab does not appear.
   - Tap an alert to see the alert details and to take action.

View Alert USM Properties

The Alert USM Properties page displays the USM properties for the selected alert.

Follow these steps:

1. Access the Mobile Dashboard.
2. Tap the Status Indicator to the right of a given service.
3. Tap an alert.
4. Tap Details.
5. You can perform the following actions on this page:

- Tap Actions to perform available actions on the alert.
- Tap Affected CIs to display the CIs that the alert impacts.

View CIs Affected by an Alert

You can display the CIs affected by a selected alert.

Follow these steps:

1. Access the Mobile Dashboard.
2. Tap the Status Indicator to the right of a given service.
3. Tap an alert.
4. Tap Affected CIs.
5. You can perform the following actions on this page:
   - Tap Details to display the USM properties for the selected alert.
   - Tap Actions to perform available actions on the alert.
   - Tap a service to display the Alerts page for the selected service.
   - Tap a CI to display the USM properties for the CI.

View CI USM Properties

You can view the USM properties for a selected CI.

Follow these steps:

1. Access the Mobile Dashboard.
2. Tap the Status Indicator to the right of a given service.
3. Tap an alert.
4. Tap Affected CIs.
5. Tap a CI.
6. You can perform the following actions on this page:
   - Tap Alerts to display alerts impacting the CI.
   - Tap Hierarchy to display the CI hierarchy.

View CI Hierarchy

You can view the parent-child relationship between CIs and services for a selected CI.

Follow these steps:

1. Access the Mobile Dashboard.
2. Tap the Status Indicator to the right of a given service.
3. Tap an alert.
4. Tap Affected CIs.
5. Tap a CI.
6. Tap Hierarchy.
7. You can perform the following actions on this page:
   - Tap Alerts to display alerts impacting the CI.
   - Tap Details to view the CI USM properties.
   - Tap a service to display the metrics for that service.

View CI Alerts

You can view the alerts impacting a CI.

Follow these steps:

1. Access the Mobile Dashboard.
2. Tap the Status Indicator to the right of a given service.
3. Tap an alert.
4. Tap Affected CIs.
5. Tap a CI.
6. Tap Alerts.
7. You can perform the following actions on this page:
   – Tap Details to view the CI USM properties.
   – Tap Hierarchy to display the CI hierarchy.
   – Tap an alert to see the Actions page for the alert.

Perform Actions on Alerts on the Mobile Dashboard

Contents

As an administrator or an operator (with access privileges), you can take action on alerts: email alerts, acknowledge or unacknowledge alerts, clear alerts, exempt or unexempt alerts, and view root causes of alerts. You can also access the Actions page from alert queues.

Follow these steps:

1. Access the Mobile Dashboard.
2. Tap the Status Indicator to the right of a given service.
3. Tap an alert.
4. You can perform the following actions on this page:
   – Tap E-mail to email alert information.
   – Tap Acknowledge Alert or Unacknowledge Alert to acknowledge/unacknowledge an alert.
   – Tap Exempt Alert or Unexempt Alert to exempt/unexempt an alert.
   – Tap Clear Alert to clear an alert.

     NOTE
     The Exempt, Unexempt, and Clear actions can be used only with infrastructure alerts.
   – Tap any other escalation action that is listed to perform it. For more information about creating escalation actions, see Create Escalation Actions.
   – Tap Details to display the USM properties for the selected alert.
   – Tap Affected CIs to display the CIs that the alert impacts.

Email Alerts

You can send an email from your mobile device that details the alert conditions and provides a link to the alert on the Mobile Dashboard.

NOTE

For this feature to work, an SMTP server needs to be configured in the Administration, E-mail Configuration section of the CA SOI Dashboard.

Follow these steps:

1. Access the Mobile Dashboard.
2. Tap the Status Indicator to the right of a given service.
3. Tap an alert.

4. Tap Email.
   The Notification Methods dialog opens with an automatically populated message containing the alert details. The subject of this message is the alert label.
5. Modify the subject and message if necessary, then enter an email recipient, then tap Send.
   CA SOI sends the email.

Escalate Alerts

You can send an escalation action for a selected alert. Only escalation actions defined in the Operations Console appear on the list.

> **NOTE**
> This feature is available only if you have appropriate access privileges. For more information, contact your CA SOI administrator.

Follow these steps:

1. Access the Mobile Dashboard.
2. Tap the Status Indicator to the right of a given service.
3. Tap an alert.
4. Tap an escalation action.
   The escalation action is performed and a message appears indicating the action was performed successfully. If the action opened a ticket, then a request number appears also.

View Alert Root Cause

A *root cause alert* is an alert that CA SOI determines after analyzing the alerts associated with a service which is based on one of the following criteria:

1. A triggered root cause rule determining the alert that is the true root cause of the service degradation which is based on relationships and topology.
2. The alert with the highest impact if no root cause rules have been triggered.

Follow these steps:

1. Access the Mobile Dashboard.
2. Tap the Status Indicator to the right of a given service.
3. Tap an alert.

   > **NOTE**
   > Root cause links are provided for service alerts only.
4. Tap Go to root cause.
   The Actions page opens for the selected root cause alert.

Acknowledge or Unacknowledge Alerts

You can acknowledge or unacknowledge alerts to let other users know that you are acting on the alert. When a user acknowledges an alert, a blue check icon displays next to the alert to let other users know it is acknowledged. To acknowledge or unacknowledge an alert, use the following feature.

> **NOTE**
> This feature is available only if you have appropriate access privileges. For more information, contact your CA SOI administrator.

Follow these steps:

1. Access the Mobile Dashboard.

2. Tap the Status Indicator to the right of a given service.
3. Tap an alert.
4. Tap Acknowledge Alert or Unacknowledge Alert.
   A confirmation dialog opens.
5. Tap Yes.
   The alert is acknowledged or unacknowledged.

Exempt Alerts on a Mobile Device

You can exempt or unexempt alerts on the Mobile Dashboard.

Follow these steps:

1. Access the Mobile Dashboard.
2. Tap the Status Indicator to the right of a given service.
3. Tap an alert.
4. Tap Exempt Alert or Unexempt Alert and confirm the operation.

Clear Alerts

You clear an alert when you resolve the situation that caused the creation of the alert. You can only clear infrastructure alerts, not service alerts.

> **NOTE**
> This feature is available only if you have appropriate access privileges. For more information, contact your CA SOI administrator.
>
> **NOTE**
> Service alerts imported from the CA SOI Domain connector are treated similarly to alerts imported from any domain manager, and therefore can be cleared.

Follow these steps:

1. Access the Mobile Dashboard.
2. Tap the Status Indicator to the right of a given service.
3. Tap an alert.
4. Tap Clear Alert.
   A confirmation dialog opens.
5. Tap Yes.
   The alert is cleared.

> **NOTE**
>
> If an error message appears or the clear alert was unsuccessful, the administrator may have enabled the 'Respect Underlying MDR Clear Alert Setting' option. This option prevents you from clearing alerts in CA SOI that are not clearable in the source domain manager.

Generating Reports in CABI JasperReports Server

Contents

As an operator, you can add or remove users in CA SOI report group and schedule or run predefined reports in JasperReports Server. You use these reports to view metrics and service details over specified time ranges. Your administrator defines your access privilege to generate reports.

As the viewing and managing reports are performed in JasperReports Server, you can access the online help from JasperReports Server.

Log in to CABI JasperReports Server

Log in to JasperReports Server to access reporting functionality.

Follow these steps:

1. Click the Reports link in the top right corner of the CA SOI interface.

   NOTE

   If the Reports link does not appear next to the Console link, the reporting functionality is not configured. Contact your CA SOI administrator or perform the procedure that is described in Configure Report Server on the Dashboard if you have the required user permission.
2. Enter a valid user name and password and then click Log In.

- – Organization: soi
  – User ID/Password: Contact your CA SOI administrator.

Running a Simple Report

To run a report in CABI JasperReports Server, follow these steps:

1. Log in to CABI JasperReports Server.
2. Click View, Repository.
   The folder available in the repository is listed.
3. Expand root, public, ca, Service Operations Insight, reports.
4. Expand one of the three reports. The list appears in the right pane.
5. Click on a report title to generate report.
   Report page and Input Controls appear.
6. Specify the parameters in Input Controls, click Apply, and OK.

   NOTE

   Click

   to invoke the Input Controls.

Scheduling Reports

Scheduling of reports is done to automate the generation and distribution of reports. You need to do the settings in JasperReports Server to schedule the reports.

Visit the Scheduling Reports section of the JasperReports Server page to generate and view reports.

Managing Reports

You can save and export the reports to various formats.

Visit the Exporting the Report section for more information.

Emailing Reports

You can configure CABI JasperReports Server to receive the reports through email. Visit the Configure Email section in the CABI JasperReports Server documentation.

USM Web View for PC

This section describes how operators use the USM Web View to work with CA SOI data.

Access the USM Web View Starting Page on a PC

As an administrator or an operator, you can search or browse for USM data and create CIs by accessing the USM Web View Starting Page. You can access the Starting page from either the Dashboard or by entering the URL directly into your browser.

To access the Starting page from the Dashboard

On the Dashboard, click the USM Web View link.

> **NOTE**
> The user validation for USM Web View only verifies users defined in CA EEM. Therefore, the administrator defined during installation (samuser by default) is invalid.

To access the Starting page with a URL

1. Open a web browser and enter the following URL:
   ```
   http://<UI server>:<port>/ssaweb
   ```
   – *UI server*

     Defines the name of the system where the UI Server is installed.

     - *port*
       Defines the port on which the specified server listens.

     Note: The default port is 7070 for a non-SSL connection, and 7403 for an SSL connection.

     Enter your login credentials and click OK.

     > **NOTE**
     > The user validation for USM Web View only verifies users that are defined in CA EEM. Therefore, the administrator that was defined during installation (samuser by default) is invalid.

Perform a Search with USM Web View

Contents

As an administrator or an operator, you can perform a USM Web View search using a keyword.

Follow these steps:

1. Access the Starting page.
2. Enter a search term in the Search field and click Search.

   > **NOTE**
   > The search field provides an autocomplete feature, which suggests search terms as you type.

   The search results display.

Advanced Search Queries

The topics in this section provide advanced methods for creating search queries.

CA SOI uses Apache Lucene and Apache Solr as the search platform. The complete syntax information can be found on the following websites:

- Apache Lucene
- Solr Wiki

Special Characters

You can define special characters that should be part of your query by escaping the special characters with backslashes (\). The special characters include the following: \ : ? + - && || {} [] () ! ^ * "

For example, to search for the text "(A:M)", you enter the following query:

```
\(A\:M\)\
```

Wildcards

You can use the following wildcards to substitute for single or multiple characters in search queries:

- **?**
  Performs a substitution on a single character.
  Example: A query of "d?g" returns "dog" and "dig."
- *****
  Performs a substitution on multiple characters.
  Example: A query of "rain*" returns "rainbow" and "rains."

Fuzzy Searches

A *fuzzy search* returns items that are similar to your search term. You add a tilde (~) to the end of your search query to perform a fuzzy search. For example, if you entered "well~" the following items return: "sell" and "tell."

You can add a value between 0 through 1 to force the search to find less or more similar terms where a value of 0 is less similar and 1 is more similar. The default value is 0.5.

For example, the following query forces the search for more similar matches to "well":

```
well~0.9
```

Proximity

You can create a search to find words that are within a specified number of words of each other. The following search looks for the words "XP" and "Vista" within 15 words of each other:

```
"XP Vista"~15
```

Boolean Expressions

You can include the following Boolean expressions to refine your searches:

- **AND**
  Specifies that both terms must be found anywhere within the document.
  Example: The following query searches for all ssa_type of person with a value of "John":
  ```
  John AND ssa_type:Person
  ```
- **+**

Specifies that the term must be found anywhere within the document.
Example: The following query returns all pages that contain "Microsoft Windows":

```
+"Microsoft Windows"
```

- **OR**
  Specifies that either term can be found anywhere in the document.
  Example: The following query returns all pages that contain "DB2" or "Microsoft Windows":

  ```
  DB2 OR "Microsoft Windows"
  ```

- **NOT or !**
  Specifies that the term must not appear in the document.
  Example: The following query returns all pages that contain "DB2" but not "Microsoft Windows":

  ```
  DB2 NOT "Microsoft Windows"
  ```

  You could enter the query as follows:

  ```
  DB2 ! "Microsoft Windows"
  ```

- **-**
  Specifies that the term must not appear in the document.
  Example: The following query returns all pages that do not contain "Microsoft Windows":

  ```
  -"Microsoft Windows"
  ```

  NOTE
  The operators must be in the upper case for the search queries to work.

Groups

You can group parts of the query in parentheses to create subqueries. Subqueries are evaluated before the rest of the query. Consider the following example:

```
(DB2 or Oracle) AND Windows
```

In this query, all pages are returned that contain either "DB2" or "Oracle" only if the document also contain Windows.

Consider another example:

```
(DB2 AND Oracle) OR (XP and Vista)
```

In this query, a page returns if *either* of the following conditions are met:

- The page contains the "DB2" and "Oracle."
- The page contains "XP" and "Vista."

You can also group fields when creating your search query. Consider the following example:

```
Description:(+Cisco -Microsoft)
```

This query returns all pages in which the Description field value contains "Cisco" but does not contain "Microsoft."

Browse the USM Data with USM Web View

Contents

As an administrator or an operator, you can browse the USM data in the following ways:

- By CI type
- By CI attribute

Browse by CI Type

You can browse for objects by USM type.

Follow these steps:

1. Access the Starting page.
2. Click Browse by CI Type, located in the Browse section.
   The Browse by CI Type options display.

   NOTE
   Icons appear for CI types that currently appear in the Persistent Store only.
3. (Optional) Select a specific data source from the drop-down list.
4. Click a CI type.
   The search results for the selected CI type display.

Browse by CI Attribute

You can browse for objects by a specific USM attribute.

Follow these steps:

1. Access the Starting page.
2. Click Browse by CI Attribute, located in the Browse section.
   The available CI attributes display in an alphabetical format.
3. Click a CI attribute.
   The Browse Attribute page displays.
4. You can do any of the following:
   – Select a result from the top ten results, which display based on highest occurrence in the repository.
   – Enter an attribute search value in the field and click Search.
   The search results display.
5. (Optional) Click Start Again to return to the Starting page.
6. (Optional) Click Back to the list of attributes to return to the Browse by CI Attribute page.

Work with USM Properties in USM Web View

As an administrator or an operator, you view the USM properties page. The page displays the USM properties for the selected CI, which can be a service, alert, relationship, and so on.

You can do the following in the Navigation section:

- Click Search again to return to the Starting page.
- Click Relationships to jump to the Relationships section.
- Click Alerts to jump to the Alert section.
- Click Go back to return to the search results.

You can do the following in the USM Properties section:

- Click Show/Hide Empty Properties to toggle USM properties that have no value entered.
- Click Override Properties to enter or override specific USM properties.

NOTE
Overriding the USM properties creates the Update to Persistent Store (rest-api) data source for that CI.

- Click Delete relationship to delete a manually created relationship.
- Click Delete entity to delete a manually created CI.
- Click Delete manual overrides to delete any manually-entered USM properties using Override Properties.
- Click Correlate to .
- Click More options and do any of the following:
  - Select a data source from the drop-down list.
  - Click the search link to search for more items of the same type.
  - Click RSS feed

    to subscribe to the RSS feed for updates performed on the current CI.
- Click a property to display search results for all CIs with the same USM property value.

The Relationships section shows the relationship of the current USM object to other USM object. The data appears depends on the CI type (service, alert, relationship, and so on).

You can do the following in the Relationships section:

- Click Show/Hide Empty Relationships to toggle showing empty relationships.
- Click From this CI or To this CI to create a relationship, which is creating an association between CIs.
- Click any relationship items to display its properties.

The Alerts section displays current alerts on all domain managers and on the currently displayed domain manager. The data that displays varies, depending on the CI type (service, alert, relationship, and so on).

You can do the following in the Alerts section:

- Click Show/Hide Empty alerts to toggle whether to display information if there is no alert.
- You can view the USM XML content for the current USM object by clicking View USM XML content.

Create and Manage Relationships with USM Web View

Contents

As an administrator or an operator, you can create and delete CI relationships with USM Web View on your PC.

Create Relationships

You can create relationships to establish associations between CIs. When you create relationships, consider the following nonrecommended scenarios that the Web View does not prevent:

- Circular relationships are not supported. Circular relationships connect the same CIs in opposite directions. For example, if Service A manages Service B through a relationship, do not create another relationship between the services where Service B is required by Service A.
- Do not create multiple relationships between the same CIs that follow the same direction (with the same source and target).

Follow these steps:

1. Perform a search or browse for the CI that you want to be the source in the relationship.
   The USM Properties page opens for the CI you select.
2. Click From this CI or To this CI in the Relationships section to create the correct directional relationship.
   The Create New Relationship page displays.

3. Complete the following mandatory fields:
   - **Semantic**
     Specifies the relationship type. Select one of the BinaryRelationship types defined in the USM schema.
   - **Source**
     Defines the source CI in the relationship. This field is already populated with the CI you originally selected. You can change the source CI if necessary.
   - **Target**
     Defines the target CI in the relationship. Click the field to open an embedded instance of the USM Web View Starting page from which you can find and select the target CI. Click the Select this link when you find the appropriate CI to populate the Target field.

   All required fields are defined.
4. Complete the remaining optional fields.

   NOTE
   The remaining optional fields vary depending on the CI selected.
5. (Optional) Complete any of the optional fields as necessary. For more information, see the descriptions to the right of each field.
6. Click Submit Changes.
   The relationship is created between the CIs within the service defined in the scope field. If the CIs did not already exist in the service, they are added with the new relationship.

   NOTE
   Creating/modifying entities using the USM Web View creates/modifies CIs/relationships in the same way as they would be created by a separate connector, which shows up as the Update of the Persistent Store data source. The properties in these entities take priority over properties from other connectors in the reconciliation process, and therefore they override the values from other connectors.

Delete Relationships

You can delete a relationship that was manually created in USM Web View using Create relationship on the USM Properties page.

Follow these steps:

1. Search or browse to locate the relationship object.
2. View the USM properties for the relationship object.
3. Click More options.
4. Select Update of the Persistent Store(rest-api) from the By Data Source drop-down list.
5. Click Delete relationship.
   The relationship is deleted.

Manage CIs with USM Web View

Contents

As an administrator or an operator, you can create, delete, and correlate CIs with USM Web View on your PC.

Create CIs

You can manually create CIs on the Starting page.

Follow these steps:

1. Access the Starting page.

2. Select the USM type from the drop-down list in the Create section.
3. Click Create.
   The CI creation page displays.
4. Complete the mandatory fields and the optional fields and click Submit Changes.
   The CI is created in the Persistent Store.

Delete CIs

You can delete CIs that were manually created in USM Web View using Create a new CI on the Starting page.

Follow these steps:

1. Search or browse to locate the CI.
2. View the USM properties for the CI.
3. Click More options.
4. Select Update of the Persistent Store(rest-api) from the By Data Source drop-down list.
5. Click Delete entity.
   The CI is deleted.

Correlate CIs

Correlation is the act of comparing CIs to determine equivalencieswhether the CIs represent the same underlying entity.

You can recorrelate objects or pick CIs and manually correlate them.

> **NOTE**
> You can only correlate CIs that are of the same type.

To correlate CIs

1. Search or browse objects and navigate to the USM Properties page.
2. Click Correlate in the USM Properties section.
3. Click in the Correlate with field.
   An embedded instance of the Web View Starting page displays, from which you can find the CI to correlate.
4. Search or browse objects to find the correlation CI.
5. Click on the CI to view the details.
6. Click the Select this link when you find the appropriate CI to correlate.
   The Source Entity field populates with the selected CI.
7. Click Merge Selected Entities.

To recorrelate a CI

1. Search or browse objects and navigate to the USM Properties page.
2. Click Correlate in the USM Properties section.
3. Click Re-correlate this entity.
   The correlation engine recorrelates the CI against all projection sheets in the Persistent Store.

Delete Manual Overrides with USM Web View

As an administrator or an operator, you can delete properties entered manually on the USM Properties page, which deletes the Update to Persistent Store(rest-api) data source that was created along with the custom properties.

Follow these steps:

1. Search or browse to locate the CI.
2. View the USM properties for the CI.
3. Click More options.
4. Select Update of the Persistent Store(rest-api) from the By Data Source drop-down list.
5. Click Delete manual overrides.
   The manual overrides are deleted.

Favorite Views in USM Web View

Contents

As an administrator or an operator, you can save views to a favorite list that you can easily access from the Starting Page.

Create Favorite Views

After searching or browsing the USM data, you can save the results to a favorites list that appears on the Starting Page.

Follow these steps:

1. Search or browse the USM data to generate a results list.
2. Click More options.
3. Click Add to favorite views.

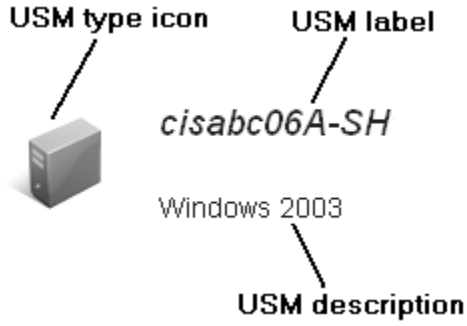
Delete Favorite Views

You can remove a favorite view from either the Starting Page or from the view itself:

- To remove a favorite view from the Starting Page, mouse over a favorite link and click Remove view .
- To remove a favorite view from the view itself, click More options and click Remove from favorite views.

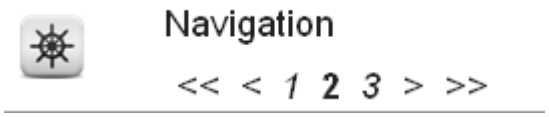
Work with Results in USM Web View

As an administrator or an operator, you can browse and search results in USM Web View, manage favorites, and subscribe to RSS feeds. The results display as a list of object entries where each entry displays the USM type icon, label, and description as shown in the following graphic:



You can do the following on the results page:

- Click an object to display the USM properties for the selected object.
- Click More/Less Results per page to change the number of items displayed per page.
- Click More Options and do any of the following:
  - Select a data source (domain manager) and a CI type from the Narrow Search drop-down list to filter the results.
  - Click RSS feed
    to subs
    to the RSS feed for the current results.
  - Click Add/Remove to favorite views to add/remove the current search to your favorites.
- Click the Navigation controls to page through the results. The following graphic shows typical navigation controls:



USM Web View for Mobile Devices

This section describes how operators use the mobile USM Web View to work with CA SOI data.

Access the USM Web View Mobile Starting Page

As an administrator or an operator, you can access the USM Web View mobile home page by URL.

Follow these steps:

1. Open a web browser and enter the following URL:
   `http://<UI_server>:<port>/ssaweb/m`
   - UI_server
     Defines the name of the system where the UI Server is installed.
   - port
     Defines the port on which the specified server listens.
     The default port is 7070 for a non-SSL connection, and 7403 for an SSL connection.
2. Enter your login credentials and tap OK.

   NOTE

   The user validation for USM Web View only verifies users that are defined in CA EEM. Therefore, the administrator that was defined during installation (samuser by default) is invalid.

Perform a USM Web View Mobile Search

As an administrator or an operator, you perform a search using a keyword and optionally on a specific domain manager on your mobile device.

Follow these steps:

1. Access the Starting page.
2. Enter a search term in the Search the IT repository field.
   Note: For information about creating advanced queries, see Advanced Search Queries.
3. (Optional) Tap Expand

   and select a domain manager from the drop-down list.

4. Tap Start

   searching
   The search results display.

Browse the USM Data with USM Web View Mobile

Contents

As an administrator or an operator, you can browse the USM data using USM Web View in the following ways on your mobile device:

- By type
- By attribute

Browse by CI Type

You can browse for objects by type.

Follow these steps:

1. Access the Starting page.
2. Tap Go

   to the right of Browse by CI Type.
3. Tap Go

   to the right of the entity you want to browse.
   The available USM types display.
4. Tap Go

   to the right of the USM type.
   The search results display.

Browse by CI Attribute

You can browse for objects by attribute.

Follow these steps:

1. Access the Starting page.
2. Tap Go

   to the right of Browse by CI Attribute.
   The browse entities by attribute page displays.

3. Tap Go

   to the right of an attribute.
   The Browse by CI Attribute page displays.
4. Tap Go

   to the right of a USM attribute.
   The Browsing by page displays.
5. Enter a value for the selected attribute and tap

   Submit .
   The search results display.

Results and USM Properties in USM Web View Mobile

Contents

As an administrator or an operator, after you search or browse the database using USM Web View, you can view the search results and USM properties on your mobile device.

Work with Results

The browse and search results display as a list of object entries.

You can do the following on this page:

- Tap RSS feed

  to subscribe to an RSS feed for the current search results. For more information, see Subscribe to RSS Feeds.
- Tap Go

  to the right of a CI to display the USM properties for the CI.
- Tap Expand

  Other Data Repositories to display available domain managers. Click Go for

to the right of a domain manager to repeat the current search for the selected domain manager.

- Tap paging controls

(if active) to navigate the results page. The USM properties page displays the USM properties for the selected USM object.

Work with USM Properties

The USM Properties page displays the USM properties for the USM object. You can perform searches related to the object's properties and manually created relationships with the object.

You can do the following in the banner and USM Properties section:

- Tap the USM type icon to display all USM objects of that type in the current MDR.
- Tap RSS feed

to subscribe to an RSS feed. For more information, see Subscribe to RSS Feeds.

- Tap Override

Properties
to override the current attributes with values you enter.

- Tap Go

to the right of a USM property to display the USM property value and allow you to search for other CIs with the same property.

You can do the following in the Alerts on this MDR section:

- Tap Go

to the right of an alert under Alerts on All MDRs to display the alert's USM properties, alerted item, and alerted service.

You can do the following in the Edit actions section:

- Tap Go

  to the right of Create new relationship to create a new USM relationship.
- Tap Go

  to the right of Delete relationship to delete the relationship.
- Tap Go

  to the right of Delete entity to delete the CI.
- Tap Go

  to the right of Delete manual overrides to delete any manually-entered USM properties using Override Properties.

Manage Relationships with USM Web View Mobile

Contents

As an administrator or an operator, you can define and remove CI relationships using USM Web View on your mobile device.

Create Relationships

You can create relationships to establish associations between CIs. When you create relationships, consider the following:

- Circular relationships are not supported.
- Do not create multiple relationships between the same CIs that follow the same direction (with the same source and target).

Follow these steps:

1. Perform a search or browse for the CI that you want to be the source in the relationship.
   The Properties page opens for the CI you select.
2. Tap Go

   to the right of Create New Relationship.
3. Complete the following required fields:

   NOTE
   The CI property fields vary depending on the entity type you select. To view information about a CI property, mouseover the question mark

   to the right of a CI property.

- **Semantic**
  Defines the relationship type. Select one of the BinaryRelationship types defined in the USM schema.
- **Source**
  Defines the source CI in the relationship. This field is already populated with the CI you originally selected. You can change the source CI if necessary.
- **Target**
  Defines the target CI in the relationship. Enter the hexadecimal string contained in the URL of the USM Properties page for the target CI.

4. (Optional) Complete any of the optional fields as necessary. To view information about a CI property, mouseover the question mark

   to the right of a CI property.
5. Tap Submit changed values

   .
   The relationship is created between the CIs within the service defined in the scope field. If the CIs did not already exist in the service, they are added with the new relationship.

Delete Relationships

You can delete a relationship that was manually created in USM Web View using Create relationship on the USM Properties page.

Follow these steps:

1. Search or browse to locate the relationship object.
2. View the USM properties for the relationship object.
3. Tap Expand

   for
   Other Data Repositories to display data sources.
4. Tap Go

   to the right of Update of the Persistent Store.
5. Tap Go

   to the right of Delete relationship.
   The relationship is deleted.

Manage CIs with USM Web View Mobile

Contents

As an administrator or an operator, you can create and delete CIs with USM Web View on your mobile device.

Create CIs

You can manually create CIs on the Starting page.

> **NOTE**
> You cannot create CIs that correlate to existing CIs.

Follow these steps:

1. Access the Starting page.
2. Tap Go

   to the right of Create a new CI.
   The USM types display.
   Tap Go

   to the right of a USM type.
3. Complete the fields.

   Consider the following:

   - The CI property fields vary depending on the entity type you select. To view information about a CI property, mouseover the question mark

     to the right of a CI property.
   - Fields marked with an asterisk (*) are for correlation. You must complete at least one of these fields.
4. Tap the button to the right of Create entity.

Delete CIs

You can delete CIs that were manually created in USM Web View using Create a new CI on the Starting page.

Follow these steps:

1. Search or browse to locate the CI.
2. View the USM properties for the CI.
3. Tap Expand

   for Other Data Repositories to display data sources.
4. Tap Go

   to the right of Update of the Persistent Store.

5. Tap Go

   to the right of Delete entity.

Delete Manual Overrides with USM Web View Mobile

As an administrator or an operator, you can delete properties entered manually on the USM Properties page. This deletes the Update to Persistent Store(rest-api) data source that was created along with the custom properties.

Follow these steps:

1. Search or browse to locate the CI.
2. View the USM properties for the CI.
3. Tap Expand

   for Other Data Repositories to display data sources.
4. Tap Go

   to the right of Update of the Persistent Store.
5. Tap Go

   to the right of Delete manual overrides.

Subscribe to RSS Feeds in USM Web View Mobile

As an administrator or an operator, you can subscribe to RSS feeds using USM Web View on your mobile device.

Really Simple Syndication (RSS) feeds let you stay informed by having relevant and up-to-date information sent to you directly from the web sites in which you are interested. With RSS feeds, you do not need to keep checking back to a particular website to see if it has been updated. Simply subscribe to the RSS feed, much like you would subscribe to a magazine, but instead of being delivered to your physical mailbox each time the magazine is published, the information is delivered to you via an RSS feed every time your subscribed website is updated.

To subscribe and read RSS feeds you need an RSS feed reader. There are many different programs and plug-ins to view RSS feeds from such as Outlook, your internet browser (Internet Explorer, Firefox), web-based readers (My Yahoo!, Google Reader), desktop-based readers (Feed Demon), and cell phone readers. After you have subscribed to a feed, the RSS feed reader is able to check for new content at specified time intervals and retrieve the updates.

Your search and browse results allow you to subscribe to RSS feeds that allow you to monitor changes to CIs or queries by receiving the changes directly to an RSS reader.

Query feeds are based upon keyword, USM type, USM attribute, or CI relationship; the feeds update when a new item matches the query you subscribed to via RSS.

Consider the following issues:

- Internet Explorer 7 does not support authenticated RSS feeds.
- Internet Explorer 8 supports RSS authentication feeds; however, due to an apparent bug, password-protected feeds may not update correctly and no solution is available.
- Microsoft Outlook 2007 may have problems with certain RSS feeds. If you experience problems, refer to a possible solution at http://support.microsoft.com.

To subscribe to an RSS feed tap RSS

.

The result is dependent on your user agent and operating system settings. You can choose to read feeds in your browser, feed reader, and so on. On some mobile platforms, notably the iPhone, you are redirected to a web-based feed reading service. By default, clicking the icon opens the appropriate feed in your browser. From there, you can subscribe to the feed in your browser or copy the URL into an external application such as a feed reader.

You can browse to a specific CI and click the RSS icon for that CI to subscribe to a feed that reports changes on that CI. Click the appropriate icon to subscribe for CI updates, alerts on the CI, or alerts on the associated service. For example, you could subscribe to a feed that updates every time an alert occurs on a specific service. You can also subscribe for updates on a search result if you run a search or browse by a specific CI type. For example, you could browse the alert type and subscribe to a feed that updates every time an alert occurs.

Reference

This section contains the following articles:

CA SOI 4.2 Data Dictionary

This section describes the tables that are available in CA SOI SAMStore data dictionary.

dbo.ActionHistory

Schema	dbo
Name	ActionHistory
Description	This table contains historical data on the escalation actions that are performed on the alerts.

Columns

	PK	Name	Data type	Null	Attributes	Description
1	✓	ActionID	bigint			
2	✓	EscalationID	bigint			
3	✓	CIID	bigint			
4	✓	AlertID	bigint			
5		CompletedTime	datetime			
6		Succeeded	tinyint	✓	Default: 0	
7		Message	ntext	✓		
8		RetryTime	datetime	✓		

Unique keys

	Key name	Columns	Description
✓	PK_ActionHistory	ActionID, EscalationID, CIID, AlertID	

Used by

Name
dbo.PurgeClearedAlerts

dbo.ActionPropertySets

Schema	dbo
Name	ActionPropertySets

Columns

	PK	Name	Data type	Null	Attributes	Description
1		ActionPropertyID	bigint		Identity / Auto increment column	
2	✓	PropertySetID	bigint			
3	✓	ActionID	bigint			
4		PropertySetXML	nvarchar(MAX)			
5		CriteriaXML	nvarchar(MAX)	✓		
6		CriteriaDrlString	nvarchar(MAX)	✓		
7		CriteriaDrlString	datetime	✓		

Unique keys

	Key name	Columns	Description
✓	PK_ActionPropertySets	PropertySetID, ActionID	

dbo.AdminConfiguration

Schema	dbo
Name	AdminConfiguration

Columns

	PK	Name	Data type	Null	Attributes
1	✓	ConfID	bigint		Identity / Auto increment column
2		ConfType	varchar(128)		
3		ConfKey	varchar(512)		

	PK	Name	Data type	Null	Attributes
4		ConfValue	varchar(512)	✓	

Unique keys

	Key name	Columns
	PK_AdminConfiguration	ConfID

dbo.AlertActions

Schema	dbo
Name	Alert Actions

Columns

	PK	Name	Data type	Null	Attributes
1		ActionID	bigint		Identity / Auto increment column
2		UniqueID	varchar (50)		
3		ActionName	nvarchar(80)		
4		Description	nvarchar(256)	✓	
5		TargetType	int	✓	
6		ActionNode	varchar(1024)	✓	
7		ActionType	int		
8		ActionData	nvarchar(2048)	✓	
9		ActionMessage	ntext	✓	
10		ActionFrom	nvarchar(256)	✓	
11		CreatedTime	datetime		
12		CreatedBy	nvarchar(60)	✓	
13		ModifiedTime	datetime	✓	
14		ModifiedBy	nvarchar(60)	✓	
15		ActionSubject	nvarchar(1024)	✓	

	PK	Name	Data type	Null	Attributes
16		Enabled	tinyint	✓	Default: 1
17		IsDefaultAction	tinyint	✓	Default: 0

Unique keys

	Key name	Columns
	PK_AlertActions	ActionID

dbo.AlertAnnotation

Schema	dbo
Name	AlertAnnotation

Columns

	PK	Name	Data type	Null	Attributes
1		AnnotationID	bigint		Identity / Auto increment column
2		AnnotationText	nvarchar(1024)		
3		CreatedTime	datetime		
4		CreatedBy	nvarchar(60)	✓	
5		ModifiedBy	nvarchar(256)	✓	
6		AlertID	bigint		
7		ModifiedTime	int	✓	

Relations

Foreign table		Primary table	Join	Title / Name / Descriptiom
dbo.AlertAnnotation		dbo.Alerts	dbo.AlertAnnotation.AlertID = dbo.Alerts.AlertID	FK_AlertAnnotation_Alert_AlertID

Uses

Name
dbo.Alerts

Used by

Name
dbo.cleanAlertsFromRemovedConnectors
dbo.PurgeClearedAlerts

dbo.AlertEscalationActions

Schema	dbo
Name	AlertEscalationActions

Columns

	PK	Name	Data type	Null	Attributes
1		EscalActionID	bigint		Identity / Auto increment column
2		EscalationID	bigint		
3		ActionID	bigint		
4		Sequence	int		
6		AlertID	bigint		
7		ModifiedTime	int		

Unique keys

	Key name	Columns
	PK_AlertEscalationActions	EscalationID, ActionID

dbo.AlertEscalationPolicy

Schema	dbo
Name	AlertEscalationPolicy

Columns

	PK	Name	Data type	Null	Attributes
1		EscalationID	bigint		Identity / Auto increment column
2		EscalationName	nvarchar(80)		
3		UniqueID	varchar(50)		

	PK	Name	Data type	Null	Attributes
4		ScheduleType	tinyint	✓	
5		Description	nvarchar(256)	✓	
6		Global	tinyint		
7		ClassID	int	✓	
8		Enabled	tinyint		
9		RootCause	tinyint		
10		Symptom	tinyint		
11		ServiceAlarm	tinyint		
12		InfrastructureAlarm	tinyint		
13		MaintenanceMode	tinyint		
14		ServiceMaintenance Mode	tinyint		
15		XmlPolicyString	nvarchar(MAX)	✓	
16		DrlPolicyString	nvarchar(MAX)	✓	
17		AdvancedType	tinyint		
18		CalenderID	bigint	✓	
19		CreatedTime	datetime	✓	
20		CreatedBy	nvarchar(256)	✓	
21		ModifiedTime	datetime	✓	
22		ModifiedBy	nvarchar(256)	✓	
23		CompiledTime	datetime	✓	

Relations

Foreign table		Primary table	Join	Title / Name / Description
dbo.AlertEscalationPolicy	>	dbo.Class	dbo.AlertEscalationPolicy.ClassID = dbo.Class.ClassID	FK_AlertEscalationActions_Class_ClassID

Unique keys

	Key name	Columns	Description
	PK_AlertEscalationPolicy	EscalationID	

Uses

Name
dbo.Class

dbo.AlertHistory

Schema	dbo
Name	AlertHistory

Columns

	PK	Name	Data type	Null	Attributes	Description
1		AlertID	bigint			
2		CreatedTime	datetime			
3		CreatedBy	nvarchar(60)			
4		ColumnName	varchar(60)			
5		PreviousValue	nvarchar(2048)	✓		
6		Currentvalue	nvarchar(2048)	✓		

Relations

Foreign table		Primary table	Join	Title / Name / Description
dbo.AlertHistory		dbo.Alerts	dbo.AlertHistory.AlertID = dbo.Alerts.AlertID	FK_AlertHistory_Alerts_AlertID

Uses

Name
dbo.Alerts

Used by

Name
dbo.cleanAlertsFromRemovedConnectors
dbo.PurgeClearedAlerts

dbo.AlertImpact

Schema	dbo
Name	AlertImpact

Columns

	PK	Name	Data type	Null	Attributes	Description
1		AlertID	bigint			
2		ServiceCIID	bigint			
3		impact	int			
4		ModifiedTime	datetime	✓		

Relations

Foreign table		Primary table	Join	Title / Name / Description
dbo.AlertImpact		dbo.Alerts	dbo.AlertImpact.AlertID = dbo.Alerts.AlertID	FK_AlertImpact_Alerts_AlertID

Unique keys

	Key name	Columns	Description
	PK_AMS_IMPACT	AlertID, ServiceCIID	

Uses

Name
dbo.Alerts

Used by

Name
dbo.cleanAlertsFromRemovedConnectors
dbo.PurgeClearedAlerts

dbo.AlertQueueAssignments

Schema	dbo
Name	AlertQueueAssignments
Description	Contains alerts information that are assigned to the alert queues.

Columns

	PK	Name	Data type	Null	Attributes	Description
1	✓	AlertID	bigint			
2	✓	QueueID	bigint			
3		InsertedTime	datetime	✓		
4		ExitTime	datetime	✓		

Unique keys

	Key name	Columns	Description
✓	PK_AlertQueueAssignments	AlertID, QueueID	

Used by

Name
dbo.purgeAlertQueueAssignment
dbo.PurgeClearedAlerts

dbo.AlertQueues

Schema	dbo
Name	AlertQueues
Description	Contains information about the specific alerts queues that are available in the Operation Console.

Columns

	PK	Name	Data type	Null	Attributes	Description
1	✓	QueueID	bigint		Identity / Auto increment column	
2	✓	QueueName	nvarchar(256)			
3		Description	nvarchar(256)	✓		
4		CriteriaXML	nvarchar(MAX)	✓		
5		CriteriaDRL	nvarchar(MAX)	✓		
6		Priority	smallint			
7		CreatedTime	datetime	✓		
8		CreatedBy	nvarchar(256)	✓		
9		ModifiedTime	datetime	✓		
10		ModifiedBy	nvarchar(256)	✓		
11		CompiledTime	datetime	✓		
12		Deleted	smallint	✓		
13		DeletedTime	datetime	✓		

Unique keys

	Key name	Columns	Description
	PK_AlertQueues	QueueID, QueueName	

Used by

Name	
dbo.SecureAlertQueues	

dbo.AlertRelated

Schema	dbo
Name	AlertRelated

Columns

	PK	Name	Data type	Null	Attributes	Description
1		AlertID	bigint			
2		CIID	bigint			

Unique keys

	Key name	Columns	Description
	PK_AlertRelated	AlertID, CIID	

Used by

Name
dbo.PurgeClearedAlerts

dbo.AlertRelationship

Schema	dbo
Name	AlertRelationship

Columns

	PK	Name	Data type	Null	Attributes	Description
1		RootAlertID	bigint			
2		SymptomAlertID	bigint			

Unique keys

	Key name	Columns	Description
	PK_ALERT_RELATIONSHIP	RootAlertID, SymptomAlertID	

dbo.Alerts

Schema	dbo
Name	Alerts
Description	Contains details about the alerts.

Columns

	PK	Name	Data type	Null	Attributes	Description
1		AlertID	bigint		Identity / Auto increment column	
2		ConnectorID	int			
3		MDRAlarmID	varchar(256)			
4		DeviceID	nvarchar(256)	✓		
5		ModelElementID	nvarchar(256)	✓		
6		SituationMessage	nvarchar(256)	✓		
7		SituationType	varchar(64)	✓		
8		AlertDetail	nvarchar(2048)	✓		
9		ClassID	int			
10		CIID	bigint			
11		Active	tinyint		Default: 0	

	PK	Name	Data type	Null	Attributes	Description
12		Acknowledged	tinyint		Default: 0	
13		AssignedTo	nvarchar(256)	✓		
14		EscalationTime	datetime	✓		
15		LoggedTime	datetime	✓	Default: getdate()	
16		ReportedTime	datetime	✓		
17		EscalationStartTime	datetime	✓		
18		ClearedTime	datetime	✓		
19		SvcDeskTicket	varchar(64)	✓		
20		Severity	int			
21		AlertLICURL	varchar(1024)	✓		
22		TicketURL	varchar(2084)	✓		
23		AlertNamespaceMapID	varchar(256)	✓		
24		MDRticketURL	varchar(MAX)	✓		
25		SDTicketProps	text	✓		
26		UserAttribute1	nvarchar(1024)	✓		
27		UserAttribute2	nvarchar(1024)	✓		
28		UserAttribute3	nvarchar(1024)	✓		
29		UserAttribute4	nvarchar(1024)	✓		
30		UserAttribute5	nvarchar(1024)	✓		
31		UserAttribute6	nvarchar(1024)	✓		
32		UserAttribute7	nvarchar(1024)	✓		
33		UserAttribute8	nvarchar(1024)	✓		
34		UserAttribute9	nvarchar(1024)	✓		
35		UserAttribute10	nvarchar(1024)	✓		

	PK	Name	Data type	Null	Attributes	Description
36		ssaAcknowledged	tinyint			
37		ssaCleared	tinyint			
38		ssaAssigned	tinyint			
39		ElapsedTime	int	✓		
40		IsClearable	tinyint	✓		
41		RepeatCount	int	✓		
42		RetireTimestamp	datetime	✓		
43		SeverityTrend	tinyint	✓		
44		isExempt	tinyint	✓		
45		MdrIsSymptom	tinyint	✓		
46		MdrIsRootCause	tinyint	✓		
47		PendingClear	tinyint	✓		
48		ClearedBy	nvarchar(256)	✓		

Relations

Foreign table		Primary table	Join	Title / Name / Description
dbo.Alerts		dbo.Class	dbo.Alerts.ClassID = dbo.Class.ClassID	FK_Alerts_Class_ClassID
dbo.AlertAnnotation		dbo.Alerts	dbo.AlertAnnotation.AlertID = dbo.Alerts.AlertID	FK_AlertAnnotation_Alert_AlertID
dbo.AlertHistory		dbo.Alerts	dbo.AlertHistory.AlertID = dbo.Alerts.AlertID	FK_AlertHistory_Alerts_AlertID
dbo.AlertImpact		dbo.Alerts	dbo.AlertImpact.AlertID = dbo.Alerts.AlertID	FK_AlertImpact_Alerts_AlertID

Unique keys

	Key name	Columns	Description
	PK_Alerts	AlertID	

Uses

Name
dbo.Class
dbo.ActionHistory
dbo.AlertAnnotation
dbo.AlertHistory
dbo.AlertImpact
dbo.Alerts
dbo.AlertRelationship

Used by

Name
dbo.AllOutageAlerts
dbo.AllQualityAlerts
dbo.AllRiskAlerts
dbo.InfrastructureAlerts
dbo.ServiceAlerts
dbo.cleanAlertsFromRemovedConnectors
dbo.purgeAlertQueueAssignment
dbo.PurgeClearedAlerts
dbo.PurgeOutageAlert
dbo.AlertAnnotation
dbo.AlertHistory
dbo.AlertImpact

dbo.AssociationType

Schema	dbo
Name	AssociationType

Columns

	PK	Name	Data type	Null	Attributes	Description
1		AssociationID	int			
2		AssociationType	varchar(64)			
3		IsUSM	tinyint		Default: 0	
4		SSAAssocMapping	int		Default: 6	
5		DefaultPolicyID	bigint	✓	Default: 0	
6		AssociationTypeDesc	varchar(1024)			

Relations

Foreign table		Primary table	Join	Title / Name / Description
dbo.CIRelationship		dbo.AssociationType	dbo.CIRelationship.AssociationID = dbo.AssociationType.AssociationID	FK_CIRelationship_AssociationType

Unique keys

	Key name	Columns	Description
	PK_AssociationType	AssociationID	

Used by

Name
dbo.CIRelationship

dbo.AuditRecordActions

Schema	dbo
Name	AuditRecordActions

Columns

	PK	Name	Data type	Null	Attributes
1		ID	int		
2		Type	nvarchar(50)		

	PK	Name	Data type	Null	Attributes
3		Label	nvarchar(256)		
4		Locale	nvarchar(20)		

Unique keys

	Key name	Columns
	PK_AuditRecordActions	ID, Locale

dbo.AuditRecords

Schema	dbo
Name	AuditRecords

	PK	Name	Data type	Null	Attributes	Description
1		ID	bigint		Identity / Auto increment column	
2		Type	int			
3		TypeDetail	nvarchar(256)			
4		InternalID	bigint			
5		TenantID	bigint			
6		Action	int			
7		ActionDetail	nvarchar(256)	✓		
8		Component	nvarchar(256)			
9		UserName	nvarchar(256)	✓		
10		TimeStamp	datetime			

	Key name	Columns	Description
	PK_AuditRecords	ID	

dbo.AuditRecordTypes

Schema	dbo
Name	AuditRecordTypes

	PK	Name	Data type	Null	Attributes	Description
1		ID	int			
2		Type	nvarchar(50)			
3		Label	nvarchar(256)			
4		Locale	nvarchar(20)			

	Key name	Columns	Description
	PK_AuditRecordTypes	ID, Locale	

dbo.ca_reportstrings

Schema	dbo
Name	ca_reportstrings

	PK	Name	Data type	Null	Attributes	Description
1		ID	int		Identity / Auto increment column	
2		Label_Code	nvarchar(100)			
3		Language_Code	nvarchar(255)			
4		name	nvarchar(255)			
5		Description	nvarchar(255)	✓	Default: NULL	

	Key name	Columns	Description
	PK__ca_repor__3214EC27020 84FDA	ID	

dbo.ca_ssa_alert

Schema	dbo
Name	ca_ssa_alert

NOTE

The tables starting with dbo.ca_ssa are the catalyst tables. These tables contain details about correlated CIs and reconcile CIs.

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_alerttype	nvarchar(64)	✓		
4		c_alertedmdrelementid	nvarchar(256)	✓		
5		c_alertedmdrprodinstance	nvarchar(128)	✓		
6		c_alertedmdrproduct	nvarchar(64)	✓		
7		c_isacknowledged	bit	✓		
8		c_iscleared	bit	✓		
9		c_mdrelementid	nvarchar(256)	✓		
10		c_mdrprodinstance	nvarchar(128)	✓		
11		c_mdrproduct	nvarchar(64)	✓		
12		c_metricname	nvarchar(256)	✓		
13		c_occurrencetimestamp	datetime	✓		
14		c_relatedincident	nvarchar(512)	✓		
15		c_severity	nvarchar(64)	✓		

	PK	Name	Data type	Null	Attributes	Description
16		c_tenantid	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_alert		dbo.ca_ssa_ci_detail	dbo.ca_ssa_alert.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_alert__id__4B973090

	Key name	Columns	Description
	PK__ca_ssa_a__3213E83F49AEE81E	id	
	ix_ssa_alert_mdrid	id, c_mdrelementid, c_mdrproduct, c_mdrprodinstance	

Name	
dbo.ca_ssa_alert	
dbo.ca_ssa_ci_detail	

Name
dbo.PurgeClearedAlerts

dbo.ca_ssa_application

Schema	dbo
Name	ca_ssa_application

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_accessedviatcpport	int	✓		
4		c_administrativestatus	nvarchar(64)	✓		
5		c_deviceassetnumber	nvarchar(64)	✓		
6		c_devicebiossystemid	nvarchar(256)	✓		
7		c_devicednsname	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
8		c_deviceipv4address	nvarchar(16)	✓		
9		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
10		c_deviceipv6address	nvarchar(40)	✓		
11		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
12		c_devicemacaddress	nvarchar(18)	✓		
13		c_devicephysserialnumber	nvarchar(64)	✓		
14		c_devicesysname	nvarchar(256)	✓		
15		c_instancename	nvarchar(750)	✓		
16		c_isinmaintenance	bit	✓		
17		c_label	nvarchar(256)	✓		
18		c_lastmodtimestamp	datetime	✓		
19		c_lastmodusername	nvarchar(256)	✓		
20		c_mdrelementid	nvarchar(256)	✓		
21		c_mdrprodinstance	nvarchar(128)	✓		
22		c_mdrproduct	nvarchar(64)	✓		
23		c_processdistinguishingid	nvarchar(256)	✓		
24		c_processid	int	✓		
25		c_productname	nvarchar(128)	✓		
26		c_tags	nvarchar(750)	✓		
27		c_tenantid	nvarchar(256)	✓		
28		c_typename	nvarchar(256)	✓		
29		c_vendor	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
30		c_version	nvarchar(64)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_application	✈	dbo.ca_ssa_ci_detail	dbo.ca_ssa_application.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_appli__id__4613616

	Key name	Columns	Description
🔑	PK__ca_ssa_a__3213E83F442B18F2	id	

Name	
dbo.ca_ssa_ci_detail	

dbo.ca_ssa_applicationserver

Schema	dbo
Name	ca_ssa_applicationserver

	PK	Name	Data type	Null	Attributes	Description
1	🔑	id	binary(16)			
2		c_derived	bit	✓		
3		c_accessedviatcpport	int	✓		
4		c_administrativestatus	nvarchar(64)	✓		
5		c_appservertype	nvarchar(64)	✓		
6		c_deviceassetnumber	nvarchar(64)	✓		
7		c_devicebiossystemid	nvarchar(256)	✓		
8		c_devicednsname	nvarchar(256)	✓		
9		c_deviceipv4address	nvarchar(16)	✓		
10		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
11		c_deviceipv6address	nvarchar(40)	✓		

	PK	Name	Data type	Null	Attributes	Description
12		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
13		c_devicemacaddress	nvarchar(18)	✓		
14		c_devicephysserialnumber	nvarchar(64)	✓		
15		c_devicesysname	nvarchar(256)	✓		
16		c_instancename	nvarchar(750)	✓		
17		c_isinmaintenance	bit	✓		
18		c_label	nvarchar(256)	✓		
19		c_lastmodtimestamp	datetime	✓		
20		c_lastmodusername	nvarchar(256)	✓		
21		c_mdrelementid	nvarchar(256)	✓		
22		c_mdrprodinstance	nvarchar(128)	✓		
23		c_mdrproduct	nvarchar(64)	✓		
24		c_processdistinguishingid	nvarchar(256)	✓		
25		c_processid	int	✓		
26		c_productname	nvarchar(128)	✓		
27		c_tags	nvarchar(750)	✓		
28		c_tenantid	nvarchar(256)	✓		
29		c_typename	nvarchar(256)	✓		
30		c_vendor	nvarchar(256)	✓		
31		c_version	nvarchar(64)	✓		

Relations

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_applications erver		dbo.ca_ssa_ci_detail	dbo.ca_ssa_applicationserver.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_appli__id__46D27B7

	Key name	Columns	Description
	PK__ca_ssa_a__3213E83F44EA3301	id	

Name	
dbo.ca_ssa_ci_detail	

dbo.ca_ssa_applicationsystem

Schema	dbo
Name	ca_ssa_applicationsystem

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_accessedviatcpport	int	✓		
4		c_administrativestatus	nvarchar(64)	✓		
5		c_groupname	nvarchar(256)	✓		
6		c_grouptype	nvarchar(64)	✓		
7		c_instancename	nvarchar(750)	✓		
8		c_isinmaintenance	bit	✓		
9		c_label	nvarchar(256)	✓		
10		c_lastmodtimestamp	datetime	✓		
11		c_lastmodusername	nvarchar(256)	✓		
12		c_mdrelementid	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
13		c_mdrprodinstance	nvarchar(128)	✓		
14		c_mdrproduct	nvarchar(64)	✓		
15		c_tags	nvarchar(750)	✓		
16		c_tenantid	nvarchar(256)	✓		
17		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_applicationsystem		dbo.ca_ssa_ci_detail	dbo.ca_ssa_applicationsystem.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_appli__id__505BE5AD

	Key name	Columns	Description
	PK__ca_ssa_a__3213E83F4E739D3B	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_asset

Schema	dbo
Name	ca_ssa_asset

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_acquisitiontype	nvarchar(64)	✓		
4		c_assetid	nvarchar(256)	✓		
5		c_assetnumber	nvarchar(64)	✓		

	PK	Name	Data type	Null	Attributes	Description
6		c_costcenter	nvarchar(64)	✓		
7		c_currency	nvarchar(64)	✓		
8		c_currentcost	float	✓		
9		c_hasexpiringlease	bit	✓		
10		c_hasexpiringsupport	bit	✓		
11		c_hasmissinginfo	bit	✓		
12		c_instancename	nvarchar(750)	✓		
13		c_label	nvarchar(256)	✓		
14		c_lastmodtimestamp	datetime	✓		
15		c_lastmodusername	nvarchar(256)	✓		
16		c_mdrelementid	nvarchar(256)	✓		
17		c_mdrprodinstance	nvarchar(128)	✓		
18		c_mdrproduct	nvarchar(64)	✓		
19		c_tags	nvarchar(750)	✓		
20		c_tenantid	nvarchar(256)	✓		
21		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_asset		dbo.ca_ssa_ci_detail	dbo.ca_ssa_asset.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_asset__id__58671BC9

	Key name	Columns	Description
	PK__ca_ssa_a__3213E83F567ED357	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_backgroundprocess

Schema	dbo
Name	ca_ssa_backgroundprocess

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_derived	bit	✓		
3		c_accessedviatcpport	int	✓		
4		c_administrativestatus	nvarchar(64)	✓		
5		c_deviceassetnumber	nvarchar(64)	✓		
6		c_devicebiossystemid	nvarchar(256)	✓		
7		c_devicednsname	nvarchar(256)	✓		
8		c_deviceipv4address	nvarchar(16)	✓		
9		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
10		c_deviceipv6address	nvarchar(40)	✓		
11		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
12		c_devicemacaddress	nvarchar(18)	✓		
13		c_devicephysserialnumber	nvarchar(64)	✓		
14		c_devicesysname	nvarchar(256)	✓		
15		c_instancename	nvarchar(750)	✓		

	PK	Name	Data type	Null	Attributes	Description
16		c_isforos	bit	✓		
17		c_isinmaintenance	bit	✓		
18		c_label	nvarchar(256)	✓		
19		c_lastmodtimestamp	datetime	✓		
20		c_lastmodusername	nvarchar(256)	✓		
21		c_mdrelementid	nvarchar(256)	✓		
22		c_mdrprodinstance	nvarchar(128)	✓		
23		c_mdrproduct	nvarchar(64)	✓		
24		c_processdistinguishingid	nvarchar(256)	✓		
25		c_processid	int	✓		
26		c_processname	nvarchar(256)	✓		
27		c_processstate	nvarchar(64)	✓		
28		c_startuptype	nvarchar(64)	✓		
29		c_tags	nvarchar(750)	✓		
30		c_tenantid	nvarchar(256)	✓		
31		c_typename	nvarchar(256)	✓		
32		c_vendor	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_backgroundprocess	✗	dbo.ca_ssa_ci_detail	dbo.ca_ssa_backgroundprocess.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_backg__id__3BCADD1B

	Key name	Columns	Description
🔑	PK__ca_ssa_b__3213E83F39E294A9	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_binaryrelationship

Schema	dbo
Name	ca_ssa_binaryrelationship

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_derived	bit	✓		
3		c_connectionstate	nvarchar(64)	✓		
4		c_contactroles	nvarchar(512)	✓		
5		c_instancename	nvarchar(750)	✓		
6		c_isautomatically started	bit	✓		
7		c_ishead	bit	✓		
8		c_label	nvarchar(256)	✓		
9		c_lastmodtimestamp	datetime	✓		
10		c_lastmodusername	nvarchar(256)	✓		
11		c_mdrelementid	nvarchar(256)	✓		
12		c_mdrprodinstance	nvarchar(128)	✓		
13		c_mdrproduct	nvarchar(64)	✓		
14		c_order	int	✓		
15		c_requirementtype	nvarchar(64)	✓		
16		c_scopemdrelementid	nvarchar(256)	✓		
17		c_scopemdrprodinstance	nvarchar(128)	✓		
18		c_scopemdrproduct	nvarchar(64)	✓		

	PK	Name	Data type	Null	Attributes	Description
19		c_semantic	nvarchar(64)	✓		
20		c_significance	tinyint	✓		
21		c_sourcemdrelementid	nvarchar(256)	✓		
22		c_sourcemdrprodinstance	nvarchar(128)	✓		
23		c_sourcemdrproduct	nvarchar(64)	✓		
24		c_tags	nvarchar(750)	✓		
25		c_targetmdrelementid	nvarchar(256)	✓		
26		c_targetmdrprodinstance	nvarchar(128)	✓		
27		c_targetmdrproduct	nvarchar(64)	✓		
28		c_tenantid	nvarchar(256)	✓		
29		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_binaryrelationship		dbo.ca_ssa_ci_detail	dbo.ca_ssa_binaryrelationship.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_binar__id__1F2E9E6D

	Key name	Columns	Description
	PK__ca_ssa_b__3213E83F1D4655FB	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_bootsoftware

Schema	dbo
Name	ca_ssa_bootsoftware

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_derived	bit	✓		
3		c_accessedviatcpport	int	✓		
4		c_administrativestatus	nvarchar(64)	✓		
5		c_deviceassetnumber	nvarchar(64)	✓		
6		c_devicebiossystemid	nvarchar(256)	✓		
7		c_devicednsname	nvarchar(256)	✓		
8		c_deviceipv4address	nvarchar(16)	✓		
9		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
10		c_deviceipv6address	nvarchar(40)	✓		
11		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
12		c_devicemacaddress	nvarchar(18)	✓		
13		c_devicephysserialnumber	nvarchar(64)	✓		
14		c_devicesysname	nvarchar(256)	✓		
15		c_instancename	nvarchar(750)	✓		
16		c_isinmaintenance	bit	✓		
17		c_label	nvarchar(256)	✓		
18		c_lastmodtimestamp	datetime	✓		
19		c_lastmodusername	nvarchar(256)	✓		
20		c_mdrelementid	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
21		c_mdrprodinstance	nvarchar(128)	✓		
22		c_mdrproduct	nvarchar(64)	✓		
23		c_processdistinguishingid	nvarchar(256)	✓		
24		c_productname	nvarchar(128)	✓		
25		c_tags	nvarchar(750)	✓		
26		c_tenantid	nvarchar(256)	✓		
27		c_typename	nvarchar(256)	✓		
28		c_vendor	nvarchar(256)	✓		
29		c_version	nvarchar(64)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_bootsoftware		dbo.ca_ssa_ci_detail	dbo.ca_ssa_bootsoftware.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_boots__id__4A18FC72

	Key name	Columns	Description
	PK__ca_ssa_b__3213E83F4830B400	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_businessprocessserver

Schema	dbo
Name	ca_ssa_businessprocessserver

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_derived	bit	✓		
3		c_accessedviatcpport	int	✓		
4		c_administrativestatus	nvarchar(64)	✓		
5		c_deviceassetnumber	nvarchar(64)	✓		
6		c_devicebiossystemid	nvarchar(256)	✓		
7		c_devicednsname	nvarchar(256)	✓		
8		c_deviceipv4address	nvarchar(16)	✓		
9		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
10		c_deviceipv6address	nvarchar(40)	✓		
11		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
12		c_devicemacaddress	nvarchar(18)	✓		
13		c_devicephysserialnumber	nvarchar(64)	✓		
14		c_devicesysname	nvarchar(256)	✓		
15		c_instancename	nvarchar(750)	✓		
16		c_isinmaintenance	bit	✓		
17		c_label	nvarchar(256)	✓		
18		c_lastmodtimestamp	datetime	✓		
19		c_lastmodusername	nvarchar(256)	✓		
20		c_mdrelementid	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
21		c_mdrprodinstance	nvarchar(128)	✓		
22		c_mdrproduct	nvarchar(64)	✓		
23		c_processdistinguishingid	nvarchar(256)	✓		
24		c_processid	int	✓		
25		c_productname	nvarchar(128)	✓		
26		c_tags	nvarchar(750)	✓		
27		c_tenantid	nvarchar(256)	✓		
28		c_typename	nvarchar(256)	✓		
29		c_vendor	nvarchar(256)	✓		
30		c_version	nvarchar(64)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_businessprocessserver		dbo.ca_ssa_ci_detail	dbo.ca_ssa_businessprocessserver.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_busin__id__2D7CBDC4

	Key name	Columns	Description
	PK__ca_ssa_b__3213E83F2B947552	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_businesstransaction

Schema	dbo
Name	ca_ssa_businesstransaction

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_derived	bit	✓		
3		c_administrativestatus	nvarchar(64)	✓		
4		c_groupname	nvarchar(256)	✓		
5		c_instancename	nvarchar(750)	✓		
6		c_isinmaintenance	bit	✓		
7		c_label	nvarchar(256)	✓		
8		c_lastmodtimestamp	datetime	✓		
9		c_lastmodusername	nvarchar(256)	✓		
10		c_mdrelementid	nvarchar(256)	✓		
11		c_mdrprodinstance	nvarchar(128)	✓		
12		c_mdrproduct	nvarchar(64)	✓		
13		c_tags	nvarchar(750)	✓		
14		c_tenantid	nvarchar(256)	✓		
15		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_businesstransaction	✓	dbo.ca_ssa_ci_detail	dbo.ca_ssa_businesstransaction.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_busin__id__20ACD28B

	Key name	Columns	Description
	PK__ca_ssa_b__3213E83F1EC48A19	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_changeorder

Schema	dbo
Name	ca_ssa_changeorder

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_actualcost	float	✓		
4		c_actualtotaltime	nvarchar(MAX)	✓		
5		c_assignedid	nvarchar(128)	✓		
6		c_cocategory	nvarchar(64)	✓		
7		c_cocompletioncode	nvarchar(64)	✓		
8		c_costatus	nvarchar(64)	✓		
9		c_cotype	nvarchar(64)	✓		
10		c_changebackedout	nvarchar(64)	✓		
11		c_currency	nvarchar(64)	✓		
12		c_estimatedcost	float	✓		
13		c_estimatedtotaltime	nvarchar(MAX)	✓		
14		c_fulfillmentdata	nvarchar(750)	✓		
15		c_impact	nvarchar(64)	✓		

	PK	Name	Data type	Null	Attributes	Description
16		c_instancename	nvarchar(750)	✓		
17		c_istemplate	bit	✓		
18		c_label	nvarchar(256)	✓		
19		c_lastmodtimestamp	datetime	✓		
20		c_lastmodusername	nvarchar(256)	✓		
21		c_mdrelementid	nvarchar(256)	✓		
22		c_mdrprodinstance	nvarchar(128)	✓		
23		c_mdrproduct	nvarchar(64)	✓		
24		c_needbydate	nvarchar(MAX)	✓		
25		c_priority	nvarchar(64)	✓		
26		c_projectid	nvarchar(256)	✓		
27		c_requirescabapproval	bit	✓		
28		c_severity	nvarchar(64)	✓		
29		c_tags	nvarchar(750)	✓		
30		c_tenantid	nvarchar(256)	✓		
31		c_typename	nvarchar(256)	✓		
32		c_urgency	nvarchar(64)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_changeorder	⊱	dbo.ca_ssa_ci_detail	dbo.ca_ssa_changeorder.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_chang__id__3C89F72A

	Key name	Columns	Description
	PK__ca_ssa_c__3213E83F3AA1AEB8	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_changepackage

Schema	dbo
Name	ca_ssa_changepackage

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_instancename	nvarchar(750)	✓		
4		c_isapproved	bit	✓		
5		c_label	nvarchar(256)	✓		
6		c_lastmodtimestamp	datetime	✓		
7		c_lastmodusername	nvarchar(256)	✓		
8		c_mdrelementid	nvarchar(256)	✓		
9		c_mdrprodinstance	nvarchar(128)	✓		
10		c_mdrproduct	nvarchar(64)	✓		
11		c_packagelifecyclestate	nvarchar(64)	✓		
12		c_packagename	nvarchar(256)	✓		
13		c_projectname	nvarchar(256)	✓		
14		c_tags	nvarchar(750)	✓		
15		c_tenantid	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
16		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_changepackage		dbo.ca_ssa_ci_detail	dbo.ca_ssa_changepackage.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_chang__id__5EAA0504

	Key name	Columns	Description
	PK__ca_ssa_c__3213E83F5CC1BC92	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_ci_detail

Schema	dbo
Name	ca_ssa_ci_detail

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_type	nvarchar(50)	✓		
3		c_xmlcontent	ntext	✓		
4		c_mdrproduct	nvarchar(64)			
5		c_mdrprodinstance	nvarchar(64)			
6		c_mdrelementid	nvarchar(256)			

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_alert		dbo.ca_ssa_ci_detail	dbo.ca_ssa_alert.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_alert__id__4B973090
dbo.ca_ssa_application		dbo.ca_ssa_ci_detail	dbo.ca_ssa_application.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_appli__id__46136164
dbo.ca_ssa_applicationserver		dbo.ca_ssa_ci_detail	dbo.ca_ssa_applicationserver.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_appli__id__46D27B73
dbo.ca_ssa_applicationsystem		dbo.ca_ssa_ci_detail	dbo.ca_ssa_applicationsystem.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_appli__id__505BE5AD

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_asset		dbo.ca_ssa_ci_detail	dbo.ca_ssa_asset.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_asset__id__58671BC9
dbo.ca_ssa_background process		dbo.ca_ssa_ci_detail	dbo.ca_ssa_backgroundprocess.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_backg__id__3BCADD1B
dbo.ca_ssa_binaryrelationship		dbo.ca_ssa_ci_detail	dbo.ca_ssa_binaryrelationship.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_binar__id__1F2E9E6D
dbo.ca_ssa_bootsoftware		dbo.ca_ssa_ci_detail	dbo.ca_ssa_bootsoftware.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_boots__id__4A18FC72
dbo.ca_ssa_businessprocessserver		dbo.ca_ssa_ci_detail	dbo.ca_ssa_businessprocessserver.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_busin__id__2D7CBDC4
dbo.ca_ssa_businesstransaction		dbo.ca_ssa_ci_detail	dbo.ca_ssa_businesstransaction.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_busin__id__20ACD28B
dbo.ca_ssa_changeorder		dbo.ca_ssa_ci_detail	dbo.ca_ssa_changeorder.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_chang__id__3C89F72A
dbo.ca_ssa_changepackage		dbo.ca_ssa_ci_detail	dbo.ca_ssa_changepackage.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_chang__id__5EAA0504
dbo.ca_ssa_cluster		dbo.ca_ssa_ci_detail	dbo.ca_ssa_cluster.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_clust__id__2A363CC5
dbo.ca_ssa_comment		dbo.ca_ssa_ci_detail	dbo.ca_ssa_comment.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_comme__id__17236851
dbo.ca_ssa_communicationserver		dbo.ca_ssa_ci_detail	dbo.ca_ssa_communicationserver.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_commu__id__0D99FE17
dbo.ca_ssa_compliance status		dbo.ca_ssa_ci_detail	dbo.ca_ssa_complianceestatus.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_compl__id__79C80F94
dbo.ca_ssa_computersystem		dbo.ca_ssa_ci_detail	dbo.ca_ssa_computersystem.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_compu__id__2E3BD7D3
dbo.ca_ssa_database		dbo.ca_ssa_ci_detail	dbo.ca_ssa_database.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_datab__id__53A266AC
dbo.ca_ssa_databaseinstance		dbo.ca_ssa_ci_detail	dbo.ca_ssa_databaseinstance.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_datab__id__7A8729A3
dbo.ca_ssa_directoryserver		dbo.ca_ssa_ci_detail	dbo.ca_ssa_directoryserver.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_direc__id__1BE81D6E
dbo.ca_ssa_entity		dbo.ca_ssa_ci_detail	dbo.ca_ssa_entity.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_entit__id__15A53433
dbo.ca_ssa_environmentsensor		dbo.ca_ssa_ci_detail	dbo.ca_ssa_environmentsensor.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_envir__id__59E54FE7
dbo.ca_ssa_file		dbo.ca_ssa_ci_detail	dbo.ca_ssa_file.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_file__id__4AD81681
dbo.ca_ssa_genericipdevice		dbo.ca_ssa_ci_detail	dbo.ca_ssa_genericipdevice.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_gener__id__1FEDB87C
dbo.ca_ssa_group		dbo.ca_ssa_ci_detail	dbo.ca_ssa_group.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_group__id__62AFA012
dbo.ca_ssa_hypervisormanager		dbo.ca_ssa_ci_detail	dbo.ca_ssa_hypervisormanager.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_hyper__id__70FDBF69
dbo.ca_ssa_incident		dbo.ca_ssa_ci_detail	dbo.ca_ssa_incident.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_incid__id__55209ACA

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_interfacecard		dbo.ca_ssa_ci_detail	dbo.ca_ssa_interfacecard.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_inter__id__408F9238
dbo.ca_ssa_itactivity		dbo.ca_ssa_ci_detail	dbo.ca_ssa_itactivity.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_itact__id__08D548FA
dbo.ca_ssa_itactivityprofile		dbo.ca_ssa_ci_detail	dbo.ca_ssa_itactivityprofile.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_itact__id__08162EEB
dbo.ca_ssa_itactivitytemplate		dbo.ca_ssa_ci_detail	dbo.ca_ssa_itactivitytemplate.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_itact__id__119F9925
dbo.ca_ssa_location		dbo.ca_ssa_ci_detail	dbo.ca_ssa_location.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_locat__id__636EBA21
dbo.ca_ssa_mailserver		dbo.ca_ssa_ci_detail	dbo.ca_ssa_mailserver.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_mails__id__45544755
dbo.ca_ssa_managedaccess		dbo.ca_ssa_ci_detail	dbo.ca_ssa_managedaccess.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_manag__id__3D491139
dbo.ca_ssa_managementagent		dbo.ca_ssa_ci_detail	dbo.ca_ssa_managementagent.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_manag__id__703EA55A
dbo.ca_ssa_mediadrive		dbo.ca_ssa_ci_detail	dbo.ca_ssa_mediadrive.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_media__id__041093DD
dbo.ca_ssa_memory		dbo.ca_ssa_ci_detail	dbo.ca_ssa_memory.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_memor__id__23F3538A
dbo.ca_ssa_multifunctionentity		dbo.ca_ssa_ci_detail	dbo.ca_ssa_multifunctionentity.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_multi__id__24B26D99
dbo.ca_ssa_network		dbo.ca_ssa_ci_detail	dbo.ca_ssa_network.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_netwo__id__4EDDB18F
dbo.ca_ssa_networkserver		dbo.ca_ssa_ci_detail	dbo.ca_ssa_networkserver.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_netwo__id__0CDAE408
dbo.ca_ssa_notebooks		dbo.ca_ssa_ci_detail	dbo.ca_ssa_notebooks.sheet_id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_no__sheet__0EF836A4
dbo.ca_ssa_operatingsystem		dbo.ca_ssa_ci_detail	dbo.ca_ssa_operatingsystem.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_opera__id__592635D8
dbo.ca_ssa_organizationalentity		dbo.ca_ssa_ci_detail	dbo.ca_ssa_organizationalentity.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_organ__id__75035A77
dbo.ca_ssa_person		dbo.ca_ssa_ci_detail	dbo.ca_ssa_person.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_perso__id__4F9CCB9E
dbo.ca_ssa_port		dbo.ca_ssa_ci_detail	dbo.ca_ssa_port.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_port__id__28B808A7
dbo.ca_ssa_powersupply		dbo.ca_ssa_ci_detail	dbo.ca_ssa_powersupply.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_power__id__420DC656
dbo.ca_ssa_printer		dbo.ca_ssa_ci_detail	dbo.ca_ssa_printer.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_print__id__370627FE
dbo.ca_ssa_printserver		dbo.ca_ssa_ci_detail	dbo.ca_ssa_printserver.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_print__id__324172E1
dbo.ca_ssa_problem		dbo.ca_ssa_ci_detail	dbo.ca_ssa_problem.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_probl__id__414EAC47
dbo.ca_ssa_processor		dbo.ca_ssa_ci_detail	dbo.ca_ssa_processor.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_proce__id__6C390A4C

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_project		dbo.ca_ssa_ci_detail	dbo.ca_ssa_project.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_proje__id__297722B6
dbo.ca_ssa_provisionedsoftware		dbo.ca_ssa_ci_detail	dbo.ca_ssa_provisionedsoftware.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_provi__id__16644E42
dbo.ca_ssa_request		dbo.ca_ssa_ci_detail	dbo.ca_ssa_request.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_reque__id__257187A8
dbo.ca_ssa_resourceserver		dbo.ca_ssa_ci_detail	dbo.ca_ssa_resourceserver.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_resou__id__61F08603
dbo.ca_ssa_router		dbo.ca_ssa_ci_detail	dbo.ca_ssa_router.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_route__id__035179CE
dbo.ca_ssa_runninghardware		dbo.ca_ssa_ci_detail	dbo.ca_ssa_runninghardware.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_runni__id__125EB334
dbo.ca_ssa_runningsoftware		dbo.ca_ssa_ci_detail	dbo.ca_ssa_runningsoftware.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_runni__id__7F4BDEC0
dbo.ca_ssa_securityserver		dbo.ca_ssa_ci_detail	dbo.ca_ssa_securityserver.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_secur__id__5D2BD0E6
dbo.ca_ssa_service		dbo.ca_ssa_ci_detail	dbo.ca_ssa_service.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_servi__id__2EFAF1E2
dbo.ca_ssa_servicespecification		dbo.ca_ssa_ci_detail	dbo.ca_ssa_servicespecification.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_servi__id__6774552F
dbo.ca_ssa_snmpv1access		dbo.ca_ssa_ci_detail	dbo.ca_ssa_snmpv1access.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_snmpv__id__33BFA6FF
dbo.ca_ssa_snmpv3access		dbo.ca_ssa_ci_detail	dbo.ca_ssa_snmpv3access.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_snmpv__id__66B53B20
dbo.ca_ssa_softwarecomponent		dbo.ca_ssa_ci_detail	dbo.ca_ssa_softwarecomponent.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_softw__id__546180BB
dbo.ca_ssa_storagearray		dbo.ca_ssa_ci_detail	dbo.ca_ssa_storagearray.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_stora__id__6B79F03D
dbo.ca_ssa_switch		dbo.ca_ssa_ci_detail	dbo.ca_ssa_switch.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_switc__id__33008CF0
dbo.ca_ssa_tablespace		dbo.ca_ssa_ci_detail	dbo.ca_ssa_tablespace.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_table__id__7E8CC4B1
dbo.ca_ssa_transactioncontext		dbo.ca_ssa_ci_detail	dbo.ca_ssa_transactioncontext.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_trans__id__37C5420D
dbo.ca_ssa_transactionsegment		dbo.ca_ssa_ci_detail	dbo.ca_ssa_transactionsegment.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_trans__id__1A69E950
dbo.ca_ssa_transactionserver		dbo.ca_ssa_ci_detail	dbo.ca_ssa_transactionserver.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_trans__id__1B29035F
dbo.ca_ssa_virtualizationmanager		dbo.ca_ssa_ci_detail	dbo.ca_ssa_virtualizationmanager.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_virtu__id__75C27486
dbo.ca_ssa_virtualsystem		dbo.ca_ssa_ci_detail	dbo.ca_ssa_virtualsystem.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_virtu__id__5DEAEAF5
dbo.ca_ssa_vmdatastore		dbo.ca_ssa_ci_detail	dbo.ca_ssa_vmdatastore.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_vmdat__id__38845C1C

	Key name	Columns	Description
	PK_ca_ssa_ci_detail	id	
	mdr_id_idx	c_mdrelementid, c_mdrprodinstance, c_mdrproduct	
	mdr_id_idx2	c_mdrproduct, c_mdrprodinstance, c_mdrelementid	

Name	
dbo.ca_ssa_ci_detail	
dbo.PurgeClearedAlerts	
dbo.ca_ssa_alert	
dbo.ca_ssa_application	
dbo.ca_ssa_applicationserver	
dbo.ca_ssa_applicationsystem	
dbo.ca_ssa_asset	
dbo.ca_ssa_backgroundprocess	
dbo.ca_ssa_binaryrelationship	
dbo.ca_ssa_bootsoftware	
dbo.ca_ssa_businessprocessserver	
dbo.ca_ssa_businesstransaction	
dbo.ca_ssa_changeorder	
dbo.ca_ssa_changepackage	
dbo.ca_ssa_cluster	
dbo.ca_ssa_comment	
dbo.ca_ssa_communicationserver	
dbo.ca_ssa_compliancestatus	
dbo.ca_ssa_computersystem	
dbo.ca_ssa_database	
dbo.ca_ssa_databaseinstance	
dbo.ca_ssa_directoryserver	
dbo.ca_ssa_entity	
dbo.ca_ssa_environmentsensor	
dbo.ca_ssa_file	
dbo.ca_ssa_genericipdevice	
dbo.ca_ssa_group	
dbo.ca_ssa_hypervisormanager	

Name	
dbo.ca_ssa_incident	
dbo.ca_ssa_interfacecard	
dbo.ca_ssa_itactivity	
dbo.ca_ssa_itactivityprofile	
dbo.ca_ssa_itactivitytemplate	
dbo.ca_ssa_location	
dbo.ca_ssa_mailserver	
dbo.ca_ssa_managedaccess	
dbo.ca_ssa_managementagent	
dbo.ca_ssa_mediadrive	
dbo.ca_ssa_memory	
dbo.ca_ssa_multifunctionentity	
dbo.ca_ssa_network	
dbo.ca_ssa_networkserver	
dbo.ca_ssa_notebooks	
dbo.ca_ssa_operatingsystem	
dbo.ca_ssa_organizationalentity	
dbo.ca_ssa_person	
dbo.ca_ssa_port	
dbo.ca_ssa_powersupply	
dbo.ca_ssa_printer	
dbo.ca_ssa_printserver	
dbo.ca_ssa_problem	
dbo.ca_ssa_processor	
dbo.ca_ssa_project	
dbo.ca_ssa_provisionedsoftware	
dbo.ca_ssa_request	
dbo.ca_ssa_resourceserver	
dbo.ca_ssa_router	
dbo.ca_ssa_runninghardware	
dbo.ca_ssa_runningsoftware	
dbo.ca_ssa_securityserver	
dbo.ca_ssa_service	
dbo.ca_ssa_servicespecification	
dbo.ca_ssa_snmpv1access	
dbo.ca_ssa_snmpv3access	
dbo.ca_ssa_softwarecomponent	
dbo.ca_ssa_storagearray	
dbo.ca_ssa_switch	
dbo.ca_ssa_tablespace	

Name	
dbo.ca_ssa_transactioncontext	
dbo.ca_ssa_transactionsegment	
dbo.ca_ssa_transactionserver	
dbo.ca_ssa_virtualizationmanager	
dbo.ca_ssa_virtualsystem	
dbo.ca_ssa_vmdatastore	

dbo.ca_ssa_ci_timestamp

Schema	dbo
Name	ca_ssa_ci_timestamp

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_creationtime	datetime	✓		
3		c_lastmodtime	datetime	✓		
4		c_deletedtime	datetime	✓		

	Key name	Columns	Description
	PK_ca_ssa_ci_timestamp	id	

Name	
dbo.PurgeClearedAlerts	
dbo.PurgeOutageAlert	

dbo.ca_ssa_cluster

Schema	dbo
Name	ca_ssa_cluster

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_administrativestatus	nvarchar(64)	✓		
4		c_groupname	nvarchar(256)	✓		
5		c_grouptype	nvarchar(64)	✓		
6		c_instancename	nvarchar(750)	✓		
7		c_isinmaintenance	bit	✓		
8		c_label	nvarchar(256)	✓		
9		c_lastmodtimestamp	datetime	✓		
10		c_lastmodusername	nvarchar(256)	✓		
11		c_mdrelementid	nvarchar(256)	✓		
12		c_mdrprodinstance	nvarchar(128)	✓		
13		c_mdrproduct	nvarchar(64)	✓		
14		c_primarydnsname	nvarchar(256)	✓		
15		c_primaryipv4address	nvarchar(16)	✓		
16		c_primaryipv4addresswithdomain	nvarchar(256)	✓		
17		c_primaryipv6address	nvarchar(40)	✓		
18		c_primaryipv6addresswithdomain	nvarchar(256)	✓		
19		c_tags	nvarchar(750)	✓		
20		c_tenantid	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
21		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_cluster		dbo.ca_ssa_ci_detail	dbo.ca_ssa_cluster.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_clust__id__2A363CC5

	Key name	Columns	Description
	PK__ca_ssa_c__3213E83F284DF453	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_comment

Schema	dbo
Name	ca_ssa_comment

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_commentid	nvarchar(128)	✓		
4		c_instancename	nvarchar(750)	✓		
5		c_lastmodtimestamp	datetime	✓		
6		c_lastmodusername	nvarchar(256)	✓		
7		c_mdrelementid	nvarchar(256)	✓		
8		c_mdrprodinstance	nvarchar(128)	✓		
9		c_mdrproduct	nvarchar(64)	✓		
10		c_tags	nvarchar(750)	✓		

	PK	Name	Data type	Null	Attributes	Description
11		c_tenantid	nvarchar(256)	✓		
12		c_text	nvarchar(750)	✓		
13		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_comment		dbo.ca_ssa_ci_detail	dbo.ca_ssa_comment.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_comme__id__17236851

	Key name	Columns	Description
	PK__ca_ssa_c__3213E83F153B1FDF	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_communicationserver

Schema	dbo
Name	ca_ssa_communicationserver

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_accessedviatcpport	int	✓		
4		c_administrativestatus	nvarchar(64)	✓		
5		c_communicationcapabilities	nvarchar(256)	✓		
6		c_deviceassetnumber	nvarchar(64)	✓		
7		c_devicebiossystemid	nvarchar(256)	✓		
8		c_devicednsname	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
9		c_deviceipv4address	nvarchar(16)	✓		
10		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
11		c_deviceipv6address	nvarchar(40)	✓		
12		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
13		c_devicemacaddress	nvarchar(18)	✓		
14		c_devicephysserialnumber	nvarchar(64)	✓		
15		c_devicesysname	nvarchar(256)	✓		
16		c_instancename	nvarchar(750)	✓		
17		c_isinmaintenance	bit	✓		
18		c_label	nvarchar(256)	✓		
19		c_lastmodtimestamp	datetime	✓		
20		c_lastmodusername	nvarchar(256)	✓		
21		c_mdrelementid	nvarchar(256)	✓		
22		c_mdrprodinstance	nvarchar(128)	✓		
23		c_mdrproduct	nvarchar(64)	✓		
24		c_processdistinguishingid	nvarchar(256)	✓		
25		c_processid	int	✓		
26		c_productname	nvarchar(128)	✓		
27		c_tags	nvarchar(750)	✓		
28		c_tenantid	nvarchar(256)	✓		
29		c_typename	nvarchar(256)	✓		
30		c_vendor	nvarchar(256)	✓		
31		c_version	nvarchar(64)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_communicationserver		dbo.ca_ssa_ci_detail	dbo.ca_ssa_communicationserver.c = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_commu__id__0D99FE17

	Key name	Columns	Description
	PK__ca_ssa_c__3213E83F0BB1B5A5	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_compliancestatus

Schema	dbo
Name	ca_ssa_compliancestatus

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_compliancecontext	nvarchar(64)	✓		
4		c_complianceid	nvarchar(256)	✓		
5		c_compliancestatus	nvarchar(64)	✓		
6		c_instancename	nvarchar(750)	✓		
7		c_label	nvarchar(256)	✓		
8		c_lastmodtimestamp	datetime	✓		
9		c_lastmodusername	nvarchar(256)	✓		
10		c_mdrelementid	nvarchar(256)	✓		
11		c_mdrprodinstance	nvarchar(128)	✓		

	PK	Name	Data type	Null	Attributes	Description
12		c_mdrproduct	nvarchar(64)	✓		
13		c_tags	nvarchar(750)	✓		
14		c_tenantid	nvarchar(256)	✓		
15		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_compliance status		dbo.ca_ssa_ci_detail	dbo.ca_ssa_compliancestatus.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_compl__id__79C80F94

	Key name	Columns	Description
	PK__ca_ssa_c__3213E83F77DFC722	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_computersystem

Schema	dbo
Name	ca_ssa_computersystem

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_administrativestatus	nvarchar(64)	✓		
4		c_assetnumber	nvarchar(64)	✓		
5		c_biossystemid	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
6		c_computername	nvarchar(256)	✓		
7		c_instancename	nvarchar(750)	✓		
8		c_isinmaintenance	bit	✓		
9		c_label	nvarchar(256)	✓		
10		c_lastmodtimestamp	datetime	✓		
11		c_lastmodusername	nvarchar(256)	✓		
12		c_mdrelementid	nvarchar(256)	✓		
13		c_mdrprodinstance	nvarchar(128)	✓		
14		c_mdrproduct	nvarchar(64)	✓		
15		c_memoryingb	float	✓		
16		c_model	nvarchar(128)	✓		
17		c_numberofcores	int	✓		
18		c_physserialnumber	nvarchar(64)	✓		
19		c_primarydnsname	nvarchar(256)	✓		
20		c_primaryipv4address	nvarchar(16)	✓		
21		c_primaryipv4addresswithdomain	nvarchar(256)	✓		
22		c_primaryipv6address	nvarchar(40)	✓		
23		c_primaryipv6addresswithdomain	nvarchar(256)	✓		
24		c_primarymacaddress	nvarchar(18)	✓		
25		c_primaryostype	nvarchar(64)	✓		
26		c_primaryosversion	nvarchar(64)	✓		
27		c_processortype	nvarchar(64)	✓		
28		c_storageingb	float	✓		

	PK	Name	Data type	Null	Attributes	Description
29		c_sysname	nvarchar(256)	✓		
30		c_systemtype	nvarchar(64)	✓		
31		c_tags	nvarchar(750)	✓		
32		c_tenantid	nvarchar(256)	✓		
33		c_typename	nvarchar(256)	✓		
34		c_vendor	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_computersystem		dbo.ca_ssa_ci_detail	dbo.ca_ssa_computersystem.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_compu__id__2E3BD7D3

	Key name	Columns	Description
	PK__ca_ssa_c__3213E83F2C538F61	id	

Name
dbo.ca_ssa_ci_detail

✓

dbo.ca_ssa_database

Schema	dbo
Name	ca_ssa_database

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		

	PK	Name	Data type	Null	Attributes	Description
3		c_administrativestatus	nvarchar(64)	✓		
4		c_dbinstancename	nvarchar(256)	✓		
5		c_databasename	nvarchar(256)	✓		
6		c_deviceassetnumber	nvarchar(64)	✓		
7		c_devicebiossystemid	nvarchar(256)	✓		
8		c_devicednsname	nvarchar(256)	✓		
9		c_deviceipv4address	nvarchar(16)	✓		
10		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
11		c_deviceipv6address	nvarchar(40)	✓		
12		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
13		c_devicemacaddress	nvarchar(18)	✓		
14		c_devicephysserialnumber	nvarchar(64)	✓		
15		c_devicesysname	nvarchar(256)	✓		
16		c_instancename	nvarchar(750)	✓		
17		c_isinmaintenance	bit	✓		
18		c_label	nvarchar(256)	✓		
19		c_lastmodtimestamp	datetime	✓		
20		c_lastmodusername	nvarchar(256)	✓		
21		c_mdrelementid	nvarchar(256)	✓		
22		c_mdrprodinstance	nvarchar(128)	✓		
23		c_mdrproduct	nvarchar(64)	✓		
24		c_tags	nvarchar(750)	✓		
25		c_tenantid	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
26		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_database		dbo.ca_ssa_ci_detail	dbo.ca_ssa_database.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_datab__id_ _53A266AC

	Key name	Columns	Description
	PK__ca_ssa_d__3213E83F51BA1E3A	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_databaseinstance

Schema	dbo
Name	ca_ssa_databaseinstance

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_accessedviatcpport	int	✓		
4		c_administrativestatus	nvarchar(64)	✓		
5		c_dbinstancename	nvarchar(256)	✓		
6		c_dbservertype	nvarchar(64)	✓		
7		c_deviceassetnumber	nvarchar(64)	✓		
8		c_devicebiossystemid	nvarchar(256)	✓		
9		c_devicednsname	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
10		c_deviceipv4address	nvarchar(16)	✓		
11		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
12		c_deviceipv6address	nvarchar(40)	✓		
13		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
14		c_devicemacaddress	nvarchar(18)	✓		
15		c_devicephysserialnumber	nvarchar(64)	✓		
16		c_devicesysname	nvarchar(256)	✓		
17		c_instancename	nvarchar(750)	✓		
18		c_isinmaintenance	bit	✓		
19		c_label	nvarchar(256)	✓		
20		c_lastmodtimestamp	datetime	✓		
21		c_lastmodusername	nvarchar(256)	✓		
22		c_mdrelementid	nvarchar(256)	✓		
23		c_mdrprodinstance	nvarchar(128)	✓		
24		c_mdrproduct	nvarchar(64)	✓		
25		c_processdistinguishingid	nvarchar(256)	✓		
26		c_processid	int	✓		
27		c_productname	nvarchar(128)	✓		
28		c_tags	nvarchar(750)	✓		
29		c_tenantid	nvarchar(256)	✓		
30		c_typename	nvarchar(256)	✓		
31		c_vendor	nvarchar(256)	✓		
32		c_version	nvarchar(64)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_databaseinstance		dbo.ca_ssa_ci_detail	dbo.ca_ssa_databaseinstance.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_datab__id__7A8729A3

	Key name	Columns	Description
	PK__ca_ssa_d__3213E83F789EE131	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_directoryserver

Schema	dbo
Name	ca_ssa_directoryserver

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_accessedviatcpport	int	✓		
4		c_administrativestatus	nvarchar(64)	✓		
5		c_deviceassetnumber	nvarchar(64)	✓		
6		c_devicebiossystemid	nvarchar(256)	✓		
7		c_devicednsname	nvarchar(256)	✓		
8		c_deviceipv4address	nvarchar(16)	✓		
9		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
10		c_deviceipv6address	nvarchar(40)	✓		
11		c_deviceipv6addresswithdomain	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
12		c_devicemacaddress	nvarchar(18)	✓		
13		c_devicephysserialnumber	nvarchar(64)	✓		
14		c_devicesysname	nvarchar(256)	✓		
15		c_instancename	nvarchar(750)	✓		
16		c_isinmaintenance	bit	✓		
17		c_label	nvarchar(256)	✓		
18		c_lastmodtimestamp	datetime	✓		
19		c_lastmodusername	nvarchar(256)	✓		
20		c_mdrelementid	nvarchar(256)	✓		
21		c_mdrprodinstance	nvarchar(128)	✓		
22		c_mdrproduct	nvarchar(64)	✓		
23		c_processdistinguishingid	nvarchar(256)	✓		
24		c_processid	int	✓		
25		c_tags	nvarchar(750)	✓		
26		c_tenantid	nvarchar(256)	✓		
27		c_typename	nvarchar(256)	✓		
28		c_vendor	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_directoryserver		dbo.ca_ssa_ci_detail	dbo.ca_ssa_directoryserver.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_direc__id__1BE81D6E

	Key name	Columns	Description
	PK__ca_ssa_d__3213E83F19FFD4FC	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_entity

Schema	dbo
Name	ca_ssa_entity

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_derived	bit	✓		
3		c_identifyingname	nvarchar(256)	✓		
4		c_instancename	nvarchar(750)	✓		
5		c_label	nvarchar(256)	✓		
6		c_lastmodtimestamp	datetime	✓		
7		c_lastmodusername	nvarchar(256)	✓		
8		c_mdrelementid	nvarchar(256)	✓		
9		c_mdrprodinstance	nvarchar(128)	✓		
10		c_mdrproduct	nvarchar(64)	✓		
11		c_tags	nvarchar(750)	✓		
12		c_tenantid	nvarchar(256)	✓		
13		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_entity	✓	dbo.ca_ssa_ci_detail	dbo.ca_ssa_entity.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_entit__id__15A53433

	Key name	Columns	Description
	PK__ca_ssa_e__3213E83F13BCEBC1	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_environmentsensor

Schema	dbo
Name	ca_ssa_environmentsensor

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_administrativestatus	nvarchar(64)	✓		
4		c_containingindex	int	✓		
5		c_deviceassetnumber	nvarchar(64)	✓		
6		c_devicebiossystemid	nvarchar(256)	✓		
7		c_devicednsname	nvarchar(256)	✓		
8		c_deviceipv4address	nvarchar(16)	✓		
9		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
10		c_deviceipv6address	nvarchar(40)	✓		
11		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
12		c_devicemacaddress	nvarchar(18)	✓		
13		c_devicephysserialnumber	nvarchar(64)	✓		
14		c_devicesysname	nvarchar(256)	✓		
15		c_instancename	nvarchar(750)	✓		

	PK	Name	Data type	Null	Attributes	Description
16		c_isinmaintenance	bit	✓		
17		c_label	nvarchar(256)	✓		
18		c_lastmodtimestamp	datetime	✓		
19		c_lastmodusername	nvarchar(256)	✓		
20		c_mdrelementid	nvarchar(256)	✓		
21		c_mdrprodinstance	nvarchar(128)	✓		
22		c_mdrproduct	nvarchar(64)	✓		
23		c_osnumeric	int	✓		
24		c_sensortype	nvarchar(64)	✓		
25		c_tags	nvarchar(750)	✓		
26		c_tenantid	nvarchar(256)	✓		
27		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_environmentsensor		dbo.ca_ssa_ci_detail	dbo.ca_ssa_environmentsensor.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_envir__id__59E54FE7

	Key name	Columns	Description
	PK__ca_ssa_e__3213E83F57FD0775	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_events

Schema	dbo
Name	ca_ssa_events

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	nvarchar(50)			
2		topic	nvarchar(50)	✓		
3		payload	image	✓		
4		timestamp	bigint	✓		

	Key name	Columns	Description
✓	PK__ca_ssa_e__3213E83F664B26CC	id	

dbo.ca_ssa_file

Schema	dbo
Name	ca_ssa_file

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_derived	bit	✓		
3		c_administrativestatus	nvarchar(64)	✓		
4		c_deviceassetnumber	nvarchar(64)	✓		
5		c_devicebiossystemid	nvarchar(256)	✓		
6		c_devicednsname	nvarchar(256)	✓		
7		c_deviceipv4address	nvarchar(16)	✓		
8		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
9		c_deviceipv6address	nvarchar(40)	✓		
10		c_deviceipv6addresswithdomain	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
11		c_devicemacaddress	nvarchar(18)	✓		
12		c_devicephysserialnumber	nvarchar(64)	✓		
13		c_devicesysname	nvarchar(256)	✓		
14		c_filepathurl	nvarchar(MAX)	✓		
15		c_filetype	nvarchar(64)	✓		
16		c_instancename	nvarchar(750)	✓		
17		c_isinmaintenance	bit	✓		
18		c_label	nvarchar(256)	✓		
19		c_lastmodtimestamp	datetime	✓		
20		c_lastmodusername	nvarchar(256)	✓		
21		c_mdrelementid	nvarchar(256)	✓		
22		c_mdrprodinstance	nvarchar(128)	✓		
23		c_mdrproduct	nvarchar(64)	✓		
24		c_tags	nvarchar(750)	✓		
25		c_tenantid	nvarchar(256)	✓		
26		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_file		dbo.ca_ssa_ci_detail	dbo.ca_ssa_file.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_file__id__4AD81681

	Key name	Columns	Description
	PK__ca_ssa_f__3213E83F48EFCE0F	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_genericipdevice

Schema	dbo
Name	ca_ssa_genericipdevice

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_derived	bit	✓		
3		c_administrativestatus	nvarchar(64)	✓		
4		c_instancename	nvarchar(750)	✓		
5		c_isinmaintenance	bit	✓		
6		c_label	nvarchar(256)	✓		
7		c_lastmodtimestamp	datetime	✓		
8		c_lastmodusername	nvarchar(256)	✓		
9		c_mdrelementid	nvarchar(256)	✓		
10		c_mdrprodinstance	nvarchar(128)	✓		
11		c_mdrproduct	nvarchar(64)	✓		
12		c_primarydnsname	nvarchar(256)	✓		
13		c_primaryipv4address	nvarchar(16)	✓		
14		c_primaryipv4addresswithdomain	nvarchar(256)	✓		
15		c_primaryipv6address	nvarchar(40)	✓		
16		c_primaryipv6addresswithdomain	nvarchar(256)	✓		
17		c_primarymacaddress	nvarchar(18)	✓		

	PK	Name	Data type	Null	Attributes	Description
18		c_sysname	nvarchar(256)	✓		
19		c_tags	nvarchar(750)	✓		
20		c_tenantid	nvarchar(256)	✓		
21		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_genericipdevice	✈	dbo.ca_ssa_ci_detail	dbo.ca_ssa_genericipdevice.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_gener__id__1FEDB87C

	Key name	Columns	Description
🔑	PK__ca_ssa_g__3213E83F1E05700A	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_group

Schema	dbo
Name	ca_ssa_group

Columns

	PK	Name	Data type	Null	Attributes	Description
1	🔑	id	binary(16)			
2		c_derived	bit	✓		
3		c_administrativestatus	nvarchar(64)	✓		
4		c_groupname	nvarchar(256)	✓		
5		c_grouptype	nvarchar(64)	✓		

	PK	Name	Data type	Null	Attributes	Description
6		c_instancename	nvarchar(750)	✓		
7		c_isinmaintenance	bit	✓		
8		c_label	nvarchar(256)	✓		
9		c_lastmodtimestamp	datetime	✓		
10		c_lastmodusername	nvarchar(256)	✓		
11		c_mdrelementid	nvarchar(256)	✓		
12		c_mdrprodinstance	nvarchar(128)	✓		
13		c_mdrproduct	nvarchar(64)	✓		
14		c_membercriteria	nvarchar(512)	✓		
15		c_tags	nvarchar(750)	✓		
16		c_tenantid	nvarchar(256)	✓		
17		c_typename	nvarchar(256)	✓		

Relations

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_group	✈	dbo.ca_ssa_ci_detail	dbo.ca_ssa_group.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_group__id__62AFA012

Unique keys

	Key name	Columns	Description
🔑	PK__ca_ssa_g__3213E83F60C757A0	id	

Uses

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_hypervisormanager

Schema	dbo
Name	ca_ssa_hypervisormanager

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_derived	bit	✓		
3		c_accessedviatcpport	int	✓		
4		c_administrativestatus	nvarchar(64)	✓		
5		c_deviceassetnumber	nvarchar(64)	✓		
6		c_devicebiossystemid	nvarchar(256)	✓		
7		c_devicednsname	nvarchar(256)	✓		
8		c_deviceipv4address	nvarchar(16)	✓		
9		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
10		c_deviceipv6address	nvarchar(40)	✓		
11		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
12		c_devicemacaddress	nvarchar(18)	✓		
13		c_devicephysserialnumber	nvarchar(64)	✓		
14		c_devicesysname	nvarchar(256)	✓		
15		c_instancename	nvarchar(750)	✓		
16		c_isinmaintenance	bit	✓		
17		c_ismigrationenabled	bit	✓		
18		c_label	nvarchar(256)	✓		
19		c_lastmodtimestamp	datetime	✓		
20		c_lastmodusername	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
21		c_mdrelementid	nvarchar(256)	✓		
22		c_mdrprodinstance	nvarchar(128)	✓		
23		c_mdrproduct	nvarchar(64)	✓		
24		c_processdistinguishingid	nvarchar(256)	✓		
25		c_processid	int	✓		
26		c_productname	nvarchar(128)	✓		
27		c_tags	nvarchar(750)	✓		
28		c_tenantid	nvarchar(256)	✓		
29		c_typename	nvarchar(256)	✓		
30		c_vendor	nvarchar(256)	✓		
31		c_version	nvarchar(64)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_hypervisormanager		dbo.ca_ssa_ci_detail	dbo.ca_ssa_hypervisormanager.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_hyper__id__70FDBF69

	Key name	Columns	Description
	PK__ca_ssa_h__3213E83F6F1576F7	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_incident

Schema	dbo
Name	ca_ssa_incident

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_derived	bit	✓		
3		c_assignedid	nvarchar(128)	✓		
4		c_impact	nvarchar(64)	✓		
5		c_incidentcategory	nvarchar(64)	✓		
6		c_incidentstatus	nvarchar(64)	✓		
7		c_instancename	nvarchar(750)	✓		
8		c_ismajor	bit	✓		
9		c_isreturnedtoservice	bit	✓		
10		c_istemplate	bit	✓		
11		c_label	nvarchar(256)	✓		
12		c_lastmodtimestamp	datetime	✓		
13		c_lastmodusername	nvarchar(256)	✓		
14		c_mdrelementid	nvarchar(256)	✓		
15		c_mdrprodinstance	nvarchar(128)	✓		
16		c_mdrproduct	nvarchar(64)	✓		
17		c_outageendtimestamp	datetime	✓		
18		c_outagestarttimestamp	datetime	✓		
19		c_outagetype	nvarchar(64)	✓		
20		c_priority	nvarchar(64)	✓		

	PK	Name	Data type	Null	Attributes	Description
21		c_rootcause	nvarchar(64)	✓		
22		c_severity	nvarchar(64)	✓		
23		c_symptomcodes	nvarchar(512)	✓		
24		c_tags	nvarchar(750)	✓		
25		c_tenantid	nvarchar(256)	✓		
26		c_typename	nvarchar(256)	✓		
27		c_urgency	nvarchar(64)	✓		
28		c_useraspect	nvarchar(64)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_incident		dbo.ca_ssa_ci_detail	dbo.ca_ssa_incident.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_incid__id__55209ACA

	Key name	Columns	Description
	PK__ca_ssa_i__3213E83F53385258	id	

Name	
dbo.ca_ssa_ci_detail	

dbo.ca_ssa_interfacecard

Schema	dbo
Name	ca_ssa_interfacecard

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		

	PK	Name	Data type	Null	Attributes	Description
3		c_administrativestatus	nvarchar(64)	✓		
4		c_containingindex	int	✓		
5		c_deviceassetnumber	nvarchar(64)	✓		
6		c_devicebiossystemid	nvarchar(256)	✓		
7		c_devicednsname	nvarchar(256)	✓		
8		c_deviceipv4address	nvarchar(16)	✓		
9		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
10		c_deviceipv6address	nvarchar(40)	✓		
11		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
12		c_devicemacaddress	nvarchar(18)	✓		
13		c_devicephysserialnumber	nvarchar(64)	✓		
14		c_devicesysname	nvarchar(256)	✓		
15		c_instancename	nvarchar(750)	✓		
16		c_isinmaintenance	bit	✓		
17		c_label	nvarchar(256)	✓		
18		c_lastmodtimestamp	datetime	✓		
19		c_lastmodusername	nvarchar(256)	✓		
20		c_mdrelementid	nvarchar(256)	✓		
21		c_mdrprodinstance	nvarchar(128)	✓		
22		c_mdrproduct	nvarchar(64)	✓		
23		c_model	nvarchar(128)	✓		
24		c_osnumeric	int	✓		
25		c_tags	nvarchar(750)	✓		

	PK	Name	Data type	Null	Attributes	Description
26		c_tenantid	nvarchar(256)	✓		
27		c_typename	nvarchar(256)	✓		
28		c_vendor	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_interfacecard		dbo.ca_ssa_ci_detail	dbo.ca_ssa_interfacecard.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_inter__id__408F9238

	Key name	Columns	Description
	PK__ca_ssa_i__3213E83F3EA749C6	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_itactivity

Schema	dbo
Name	ca_ssa_itactivity

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_activityid	nvarchar(256)	✓		
4		c_activitystate	nvarchar(64)	✓		
5		c_activitytypes	nvarchar(256)	✓		
6		c_actualendtimestamp	datetime	✓		
7		c_anticipatedendtimestamp	datetime	✓		

	PK	Name	Data type	Null	Attributes	Description
8		c_clusterdnsname	nvarchar(256)	✓		
9		c_clustername	nvarchar(256)	✓		
10		c_definitionname	nvarchar(256)	✓		
11		c_definitionversion	nvarchar(64)	✓		
12		c_deviceassetnumber	nvarchar(64)	✓		
13		c_devicebiossystemid	nvarchar(256)	✓		
14		c_devicednsname	nvarchar(256)	✓		
15		c_deviceipv4address	nvarchar(16)	✓		
16		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
17		c_deviceipv6address	nvarchar(40)	✓		
18		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
19		c_devicemacaddress	nvarchar(18)	✓		
20		c_devicephysserialnumber	nvarchar(64)	✓		
21		c_devicesysname	nvarchar(256)	✓		
22		c_externalrefname	nvarchar(256)	✓		
23		c_instancename	nvarchar(750)	✓		
24		c_isdynamicallyallocated	bit	✓		
25		c_label	nvarchar(256)	✓		
26		c_lastmodtimestamp	datetime	✓		
27		c_lastmodusername	nvarchar(256)	✓		
28		c_mdrelementid	nvarchar(256)	✓		
29		c_mdrprodinstance	nvarchar(128)	✓		
30		c_mdrproduct	nvarchar(64)	✓		

	PK	Name	Data type	Null	Attributes	Description
31		c_runtimediscriminator	nvarchar(64)	✓		
32		c_runtimename	nvarchar(512)	✓		
33		c_starttimestamp	datetime	✓		
34		c_tags	nvarchar(750)	✓		
35		c_tenantid	nvarchar(256)	✓		
36		c_typename	nvarchar(256)	✓		

Relations

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_itactivity		dbo.ca_ssa_ci_detail	dbo.ca_ssa_itactivity.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_itact__id__08D548FA

	Key name	Columns	Description
	PK__ca_ssa_i__3213E83F06ED0088	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_itactivityprofile

Schema	dbo
Name	ca_ssa_itactivityprofile

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_activitytypes	nvarchar(256)	✓		
4		c_administrativestatus	nvarchar(64)	✓		

	PK	Name	Data type	Null	Attributes	Description
5		c_groupname	nvarchar(256)	✓		
6		c_grouptype	nvarchar(64)	✓		
7		c_instancename	nvarchar(750)	✓		
8		c_isinmaintenance	bit	✓		
9		c_label	nvarchar(256)	✓		
10		c_lastmodtimestamp	datetime	✓		
11		c_lastmodusername	nvarchar(256)	✓		
12		c_mdrelementid	nvarchar(256)	✓		
13		c_mdrprodinstance	nvarchar(128)	✓		
14		c_mdrproduct	nvarchar(64)	✓		
15		c_tags	nvarchar(750)	✓		
16		c_tenantid	nvarchar(256)	✓		
17		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_itactivityprofile		dbo.ca_ssa_ci_detail	dbo.ca_ssa_itactivityprofile.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_itact__id__08162EEB

	Key name	Columns	Description
	PK__ca_ssa_i__3213E83F062DE679	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_itactivitytemplate

Schema	dbo
Name	ca_ssa_itactivitytemplate

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_derived	bit	✓		
3		c_activitytypes	nvarchar(256)	✓		
4		c_definitionname	nvarchar(256)	✓		
5		c_definitionversion	nvarchar(64)	✓		
6		c_instancename	nvarchar(750)	✓		
7		c_label	nvarchar(256)	✓		
8		c_lastmodtimestamp	datetime	✓		
9		c_lastmodusername	nvarchar(256)	✓		
10		c_mdrelementid	nvarchar(256)	✓		
11		c_mdrprodinstance	nvarchar(128)	✓		
12		c_mdrproduct	nvarchar(64)	✓		
13		c_tags	nvarchar(750)	✓		
14		c_tenantid	nvarchar(256)	✓		
15		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_itactivitytemplate	✈	dbo.ca_ssa_ci_detail	dbo.ca_ssa_itactivitytemplate.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_itact__id__119F9925

	Key name	Columns	Description
	PK__ca_ssa_i__3213E83F0FB750B3	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_location

Schema	dbo
Name	ca_ssa_location

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_building	nvarchar(64)	✓		
4		c_city	nvarchar(64)	✓		
5		c_countrycode	nvarchar(64)	✓		
6		c_countryname	nvarchar(64)	✓		
7		c_instancename	nvarchar(750)	✓		
8		c_label	nvarchar(256)	✓		
9		c_lastmodtimestamp	datetime	✓		
10		c_lastmodusername	nvarchar(256)	✓		
11		c_locationname	nvarchar(256)	✓		
12		c_mdrelementid	nvarchar(256)	✓		
13		c_mdrprodinstance	nvarchar(128)	✓		
14		c_mdrproduct	nvarchar(64)	✓		
15		c_postalcode	nvarchar(64)	✓		

	PK	Name	Data type	Null	Attributes	Description
16		c_stateorprovince	nvarchar(64)	✓		
17		c_tags	nvarchar(750)	✓		
18		c_tenantid	nvarchar(256)	✓		
19		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_location	🔑	dbo.ca_ssa_ci_detail	dbo.ca_ssa_location.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_locat__id__636EBA21

	Key name	Columns	Description
✈	PK__ca_ssa_l__3213E83F618671AF	id	

Name
dbo.ca_ssa_ci_detail

✓

dbo.ca_ssa_mailserver

Schema	dbo
Name	ca_ssa_mailserver

	PK	Name	Data type	Null	Attributes	Description
1	🔑	id	binary(16)			
2		c_derived	bit	✓		
3		c_accessedviatcpport	int	✓		
4		c_administrativestatus	nvarchar(64)	✓		

	PK	Name	Data type	Null	Attributes	Description
5		c_deviceassetnumber	nvarchar(64)	✓		
6		c_devicebiossystemid	nvarchar(256)	✓		
7		c_devicednsname	nvarchar(256)	✓		
8		c_deviceipv4address	nvarchar(16)	✓		
9		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
10		c_deviceipv6address	nvarchar(40)	✓		
11		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
12		c_devicemacaddress	nvarchar(18)	✓		
13		c_devicephysserialnumber	nvarchar(64)	✓		
14		c_devicesysname	nvarchar(256)	✓		
15		c_instancename	nvarchar(750)	✓		
16		c_isinmaintenance	bit	✓		
17		c_label	nvarchar(256)	✓		
18		c_lastmodtimestamp	datetime	✓		
19		c_lastmodusername	nvarchar(256)	✓		
20		c_mdrelementid	nvarchar(256)	✓		
21		c_mdrprodinstance	nvarchar(128)	✓		
22		c_mdrproduct	nvarchar(64)	✓		
23		c_processdistinguishingid	nvarchar(256)	✓		
24		c_processid	int	✓		
25		c_productname	nvarchar(128)	✓		
26		c_tags	nvarchar(750)	✓		
27		c_tenantid	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
28		c_typename	nvarchar(256)	✓		
29		c_vendor	nvarchar(256)	✓		
30		c_version	nvarchar(64)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_mailserver		dbo.ca_ssa_ci_detail	dbo.ca_ssa_mailserver.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_mails__id__45544755

	Key name	Columns	Description
	PK__ca_ssa_m__3213E83F436BFEE3	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_managedaccess

Schema	dbo
Name	ca_ssa_managedaccess

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_accessid	nvarchar(256)	✓		
4		c_instancename	nvarchar(750)	✓		
5		c_label	nvarchar(256)	✓		
6		c_lastmodtimestamp	datetime	✓		
7		c_lastmodusername	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
8		c_managementipv4address	nvarchar(16)	✓		
9		c_managementipv6address	nvarchar(40)	✓		
10		c_managementport	int	✓		
11		c_mdrelementid	nvarchar(256)	✓		
12		c_mdrprodinstance	nvarchar(128)	✓		
13		c_mdrproduct	nvarchar(64)	✓		
14		c_tags	nvarchar(750)	✓		
15		c_tenantid	nvarchar(256)	✓		
16		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_managedaccess		dbo.ca_ssa_ci_detail	dbo.ca_ssa_managedaccess.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_manag__id__3D491139

	Key name	Columns	Description
	PK__ca_ssa_m__3213E83F3B60C8C7	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_managementagent

Schema	dbo
Name	ca_ssa_managementagent

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_derived	bit	✓		
3		c_accessedviatcpport	int	✓		
4		c_administrativestatus	nvarchar(64)	✓		
5		c_deviceassetnumber	nvarchar(64)	✓		
6		c_devicebiossystemid	nvarchar(256)	✓		
7		c_devicednsname	nvarchar(256)	✓		
8		c_deviceipv4address	nvarchar(16)	✓		
9		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
10		c_deviceipv6address	nvarchar(40)	✓		
11		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
12		c_devicemacaddress	nvarchar(18)	✓		
13		c_devicephysserialnumber	nvarchar(64)	✓		
14		c_devicesysname	nvarchar(256)	✓		
15		c_instancename	nvarchar(750)	✓		
16		c_isinmaintenance	bit	✓		
17		c_label	nvarchar(256)	✓		
18		c_lastmodtimestamp	datetime	✓		
19		c_lastmodusername	nvarchar(256)	✓		
20		c_mdrelementid	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
21		c_mdrprodinstance	nvarchar(128)	✓		
22		c_mdrproduct	nvarchar(64)	✓		
23		c_processdistinguishingid	nvarchar(256)	✓		
24		c_processid	int	✓		
25		c_productname	nvarchar(128)	✓		
26		c_tags	nvarchar(750)	✓		
27		c_tenantid	nvarchar(256)	✓		
28		c_typename	nvarchar(256)	✓		
29		c_vendor	nvarchar(256)	✓		
30		c_version	nvarchar(64)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_managementagent		dbo.ca_ssa_ci_detail	dbo.ca_ssa_managementagent.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_manag__id__703EA55A

	Key name	Columns	Description
	PK__ca_ssa_m__3213E83F6E565CE8	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_mediadrive

Schema	dbo
Name	ca_ssa_mediadrive

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_derived	bit	✓		
3		c_administrativestatus	nvarchar(64)	✓		
4		c_capacityinmb	float	✓		
5		c_containingindex	int	✓		
6		c_deviceassetnumber	nvarchar(64)	✓		
7		c_devicebiossystemid	nvarchar(256)	✓		
8		c_devicednsname	nvarchar(256)	✓		
9		c_deviceipv4address	nvarchar(16)	✓		
10		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
11		c_deviceipv6address	nvarchar(40)	✓		
12		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
13		c_devicemacaddress	nvarchar(18)	✓		
14		c_devicephysserialnumber	nvarchar(64)	✓		
15		c_devicesysname	nvarchar(256)	✓		
16		c_drivetype	nvarchar(64)	✓		
17		c_instancename	nvarchar(750)	✓		
18		c_isinmaintenance	bit	✓		
19		c_isphysical	bit	✓		
20		c_label	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
21		c_lastmodtimestamp	datetime	✓		
22		c_lastmodusername	nvarchar(256)	✓		
23		c_mdrelementid	nvarchar(256)	✓		
24		c_mdrprodinstance	nvarchar(128)	✓		
25		c_mdrproduct	nvarchar(64)	✓		
26		c_model	nvarchar(128)	✓		
27		c_osnumeric	int	✓		
28		c_supportsremovablemedia	bit	✓		
29		c_supportswrite	bit	✓		
30		c_tags	nvarchar(750)	✓		
31		c_tenantid	nvarchar(256)	✓		
32		c_typename	nvarchar(256)	✓		
33		c_vendor	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_mediadrive		dbo.ca_ssa_ci_detail	dbo.ca_ssa_mediadrive.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_media__id__041093DD

	Key name	Columns	Description
	PK__ca_ssa_m__3213E83F02284B6B	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_memory

Schema	dbo
Name	ca_ssa_memory

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_derived	bit	✓		
3		c_administrativestatus	nvarchar(64)	✓		
4		c_containingindex	int	✓		
5		c_deviceassetnumber	nvarchar(64)	✓		
6		c_devicebiossystemid	nvarchar(256)	✓		
7		c_devicednsname	nvarchar(256)	✓		
8		c_deviceipv4address	nvarchar(16)	✓		
9		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
10		c_deviceipv6address	nvarchar(40)	✓		
11		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
12		c_devicemacaddress	nvarchar(18)	✓		
13		c_devicephysserialnumber	nvarchar(64)	✓		
14		c_devicesysname	nvarchar(256)	✓		
15		c_instancename	nvarchar(750)	✓		
16		c_isinmaintenance	bit	✓		
17		c_isphysical	bit	✓		
18		c_label	nvarchar(256)	✓		
19		c_lastmodtimestamp	datetime	✓		
20		c_lastmodusername	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
21		c_mdrelementid	nvarchar(256)	✓		
22		c_mdrprodinstance	nvarchar(128)	✓		
23		c_mdrproduct	nvarchar(64)	✓		
24		c_memorytype	nvarchar(64)	✓		
25		c_osnumeric	int	✓		
26		c_sizeinmb	float	✓		
27		c_tags	nvarchar(750)	✓		
28		c_tenantid	nvarchar(256)	✓		
29		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_memory		dbo.ca_ssa_ci_detail	dbo.ca_ssa_memory.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_memor__id__23F3538A

	Key name	Columns	Description
	PK__ca_ssa_m__3213E83F220B0B18	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_multifunctionentity

Schema	dbo
Name	ca_ssa_multifunctionentity

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_derived	bit	✓		
3		c_groupname	nvarchar(256)	✓		
4		c_instancename	nvarchar(750)	✓		
5		c_label	nvarchar(256)	✓		
6		c_lastmodtimestamp	datetime	✓		
7		c_lastmodusername	nvarchar(256)	✓		
8		c_mdrelementid	nvarchar(256)	✓		
9		c_mdrprodinstance	nvarchar(128)	✓		
10		c_mdrproduct	nvarchar(64)	✓		
11		c_tags	nvarchar(750)	✓		
12		c_tenantid	nvarchar(256)	✓		
13		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_multifunctionentity	✓	dbo.ca_ssa_ci_detail	dbo.ca_ssa_multifunctionentity.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_multi__id__24B26D99

	Key name	Columns	Description
✓	PK__ca_ssa_m__3213E83F22CA2527	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_network

Schema	dbo
Name	ca_ssa_network

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_administrativestatus	nvarchar(64)	✓		
4		c_groupname	nvarchar(256)	✓		
5		c_grouptype	nvarchar(64)	✓		
6		c_instancename	nvarchar(750)	✓		
7		c_isinmaintenance	bit	✓		
8		c_label	nvarchar(256)	✓		
9		c_lastmodtimestamp	datetime	✓		
10		c_lastmodusername	nvarchar(256)	✓		
11		c_mdrelementid	nvarchar(256)	✓		
12		c_mdrprodinstance	nvarchar(128)	✓		
13		c_mdrproduct	nvarchar(64)	✓		
14		c_tags	nvarchar(750)	✓		
15		c_tenantid	nvarchar(256)	✓		
16		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_network		dbo.ca_ssa_ci_detail	dbo.ca_ssa_network.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_netwo__id__4EDDB18F

	Key name	Columns	Description
	PK__ca_ssa_n__3213E83F4CF5691D	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_networkserver

Schema	dbo
Name	ca_ssa_networkserver

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_accessedviatcpport	int	✓		
4		c_administrativestatus	nvarchar(64)	✓		
5		c_deviceassetnumber	nvarchar(64)	✓		
6		c_devicebiossystemid	nvarchar(256)	✓		
7		c_devicednsname	nvarchar(256)	✓		
8		c_deviceipv4address	nvarchar(16)	✓		
9		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
10		c_deviceipv6address	nvarchar(40)	✓		
11		c_deviceipv6addresswithdomain	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
12		c_devicemacaddress	nvarchar(18)	✓		
13		c_devicephysserialnumber	nvarchar(64)	✓		
14		c_devicesysname	nvarchar(256)	✓		
15		c_instancename	nvarchar(750)	✓		
16		c_isinmaintenance	bit	✓		
17		c_label	nvarchar(256)	✓		
18		c_lastmodtimestamp	datetime	✓		
19		c_lastmodusername	nvarchar(256)	✓		
20		c_mdrelementid	nvarchar(256)	✓		
21		c_mdrprodinstance	nvarchar(128)	✓		
22		c_mdrproduct	nvarchar(64)	✓		
23		c_processdistinguishingid	nvarchar(256)	✓		
24		c_processid	int	✓		
25		c_productname	nvarchar(128)	✓		
26		c_protocol	nvarchar(64)	✓		
27		c_tags	nvarchar(750)	✓		
28		c_tenantid	nvarchar(256)	✓		
29		c_typename	nvarchar(256)	✓		
30		c_vendor	nvarchar(256)	✓		
31		c_version	nvarchar(64)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_networkserver	⊱	dbo.ca_ssa_ci_detail	dbo.ca_ssa_networkserver.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_netwo__id__0CDAE408

	Key name	Columns	Description
	PK__ca_ssa_n__3213E83F0AF29B96	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_notebooks

Schema	dbo
Name	ca_ssa_notebooks

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		sheetname	nvarchar(100)			
3		sheetid	binary(16)			

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_notebooks		dbo.ca_ssa_ci_detail	dbo.ca_ssa_notebooks.sheetid = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_no__sheet__0EF836A4

	Key name	Columns	Description
	PK_ca_ssa_notebooks	sheetid, id	

Name
dbo.ca_ssa_ci_detail

Name
dbo.PurgeClearedAlerts44

dbo.ca_ssa_notebooks_timestamp

Schema	dbo
Name	ca_ssa_notebooks_timestamp

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_creationtime	datetime	✓		
3		c_lastmodtime	datetime	✓		
4		c_deletedtime	datetime	✓		

	Key name	Columns	Description
✓	PK_ca_ssa_notebooks_timestamp	id	

Name
dbo.PurgeClearedAlerts

dbo.ca_ssa_operatingsystem

Schema	dbo
Name	ca_ssa_operatingsystem

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_derived	bit	✓		
3		c_administrativestatus	nvarchar(64)	✓		
4		c_deviceassetnumber	nvarchar(64)	✓		
5		c_devicebiossystemid	nvarchar(256)	✓		
6		c_devicednsname	nvarchar(256)	✓		
7		c_deviceipv4address	nvarchar(16)	✓		

	PK	Name	Data type	Null	Attributes	Description
8		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
9		c_deviceipv6address	nvarchar(40)	✓		
10		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
11		c_devicemacaddress	nvarchar(18)	✓		
12		c_devicephysserialnumber	nvarchar(64)	✓		
13		c_devicesysname	nvarchar(256)	✓		
14		c_instancename	nvarchar(750)	✓		
15		c_isinmaintenance	bit	✓		
16		c_label	nvarchar(256)	✓		
17		c_lastmodtimestamp	datetime	✓		
18		c_lastmodusername	nvarchar(256)	✓		
19		c_mdrelementid	nvarchar(256)	✓		
20		c_mdrprodinstance	nvarchar(128)	✓		
21		c_mdrproduct	nvarchar(64)	✓		
22		c_ostype	nvarchar(64)	✓		
23		c_productname	nvarchar(128)	✓		
24		c_tags	nvarchar(750)	✓		
25		c_tenantid	nvarchar(256)	✓		
26		c_typename	nvarchar(256)	✓		
27		c_vendor	nvarchar(256)	✓		
28		c_version	nvarchar(64)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_operatingsystem		dbo.ca_ssa_ci_detail	dbo.ca_ssa_operatingsystem.id = dbo.ca_ssa_ci_detail.id	FK_id_ca_ssa_opera__id__592635D8

	Key name	Columns	Description
	PK__ca_ssa_o__3213E83F573DED66	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_organizationalentity

Schema	dbo
Name	ca_ssa_organizationalentity

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_emailaddresses	nvarchar(750)	✓		
4		c_groupname	nvarchar(256)	✓		
5		c_instancename	nvarchar(750)	✓		
6		c_label	nvarchar(256)	✓		
7		c_lastmodtimestamp	datetime	✓		
8		c_lastmodusername	nvarchar(256)	✓		
9		c_mdrelementid	nvarchar(256)	✓		
10		c_mdrprodinstance	nvarchar(128)	✓		
11		c_mdrproduct	nvarchar(64)	✓		

	PK	Name	Data type	Null	Attributes	Description
12		c_orgtype	nvarchar(64)	✓		
13		c_otherphonenumbers	nvarchar(750)	✓		
14		c_primaryphonenumber	nvarchar(64)	✓		
15		c_tags	nvarchar(750)	✓		
16		c_tenantid	nvarchar(256)	✓		
17		c_typename	nvarchar(256)	✓		
18		c_websites	nvarchar(750)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_organizationalentity		dbo.ca_ssa_ci_detail	dbo.ca_ssa_organizationalentity.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_organ__id__75035A77

	Key name	Columns	Description
	PK__ca_ssa_o__3213E83F731B1205	id	

Name
dbo.ca_ssa_ci_detail

✓

dbo.ca_ssa_person

Schema	dbo
Name	ca_ssa_person

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_companyname	nvarchar(128)	✓		
4		c_emailaddresses	nvarchar(750)	✓		
5		c_employeeid	nvarchar(64)	✓		
6		c_familyname	nvarchar(64)	✓		
7		c_firstname	nvarchar(64)	✓		
8		c_instancename	nvarchar(750)	✓		
9		c_isactive	bit	✓		
10		c_jobtitle	nvarchar(128)	✓		
11		c_label	nvarchar(256)	✓		
12		c_lastmodtimestamp	datetime	✓		
13		c_lastmodusername	nvarchar(256)	✓		
14		c_mdrelementid	nvarchar(256)	✓		
15		c_mdrprodinstance	nvarchar(128)	✓		
16		c_mdrproduct	nvarchar(64)	✓		
17		c_middlenames	nvarchar(128)	✓		
18		c_namedaliases	nvarchar(750)	✓		
19		c_otherphonenumbers	nvarchar(750)	✓		
20		c_primaryphonenumber	nvarchar(64)	✓		

	PK	Name	Data type	Null	Attributes	Description
21		c_tags	nvarchar(750)	✓		
22		c_tenantid	nvarchar(256)	✓		
23		c_typename	nvarchar(256)	✓		
24		c_username	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_person		dbo.ca_ssa_ci_detail	dbo.ca_ssa_person.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_perso__id__4F9CCB9E

	Key name	Columns	Description
	PK__ca_ssa_p__3213E83F4DB4832C	id	

Name	
dbo.ca_ssa_ci_detail	

✓

dbo.ca_ssa_port

Schema	dbo
Name	ca_ssa_port

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_administrativestatus	nvarchar(64)	✓		
4		c_cardcontainingindex	int	✓		
5		c_cardosnumeric	int	✓		

	PK	Name	Data type	Null	Attributes	Description
6		c_containingindex	int	✓		
7		c_deviceassetnumber	nvarchar(64)	✓		
8		c_devicebiossystemid	nvarchar(256)	✓		
9		c_devicednsname	nvarchar(256)	✓		
10		c_deviceipv4address	nvarchar(16)	✓		
11		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
12		c_deviceipv6address	nvarchar(40)	✓		
13		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
14		c_devicemacaddress	nvarchar(18)	✓		
15		c_devicephysserialnumber	nvarchar(64)	✓		
16		c_devicesysname	nvarchar(256)	✓		
17		c_ifindex	nvarchar(128)	✓		
18		c_iftype	nvarchar(64)	✓		
19		c_iftypeextension	nvarchar(64)	✓		
20		c_instancename	nvarchar(750)	✓		
21		c_isinmaintenance	bit	✓		
22		c_isphysical	bit	✓		
23		c_label	nvarchar(256)	✓		
24		c_lastmodtimestamp	datetime	✓		
25		c_lastmodusername	nvarchar(256)	✓		
26		c_mdrelementid	nvarchar(256)	✓		
27		c_mdrprodinstance	nvarchar(128)	✓		
28		c_mdrproduct	nvarchar(64)	✓		

	PK	Name	Data type	Null	Attributes	Description
29		c_model	nvarchar(128)	✓		
30		c_nomspeedinbitspersec	float	✓		
31		c_osnumeric	int	✓		
32		c_portid	nvarchar(256)	✓		
33		c_primaryipv4address	nvarchar(16)	✓		
34		c_primaryipv4addresswithdomain	nvarchar(256)	✓		
35		c_primaryipv6address	nvarchar(40)	✓		
36		c_primaryipv6addresswithdomain	nvarchar(256)	✓		
37		c_primarymacaddress	nvarchar(18)	✓		
38		c_tags	nvarchar(750)	✓		
39		c_tenantid	nvarchar(256)	✓		
40		c_typename	nvarchar(256)	✓		
41		c_vendor	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_port		dbo.ca_ssa_ci_detail	dbo.ca_ssa_port.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_port__id__28B808A7

	Key name	Columns	Description
	PK__ca_ssa_p__3213E83F26CFC035	id	

Name
dbo.ca_ssa_ci_detail

✓

dbo.ca_ssa_powersupply

Schema	dbo
Name	ca_ssa_powersupply

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_derived	bit	✓		
3		c_administrativestatus	nvarchar(64)	✓		
4		c_containingindex	int	✓		
5		c_deviceassetnumber	nvarchar(64)	✓		
6		c_devicebiossystemid	nvarchar(256)	✓		
7		c_devicednsname	nvarchar(256)	✓		
8		c_deviceipv4address	nvarchar(16)	✓		
9		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
10		c_deviceipv6address	nvarchar(40)	✓		
11		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
12		c_devicemacaddress	nvarchar(18)	✓		
13		c_devicephysserialnumber	nvarchar(64)	✓		
14		c_devicesysname	nvarchar(256)	✓		
15		c_instancename	nvarchar(750)	✓		
16		c_isinmaintenance	bit	✓		
17		c_label	nvarchar(256)	✓		
18		c_lastmodtimestamp	datetime	✓		
19		c_lastmodusername	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
20		c_mdrelementid	nvarchar(256)	✓		
21		c_mdrprodinstance	nvarchar(128)	✓		
22		c_mdrproduct	nvarchar(64)	✓		
23		c_model	nvarchar(128)	✓		
24		c_osnumeric	int	✓		
25		c_tags	nvarchar(750)	✓		
26		c_tenantid	nvarchar(256)	✓		
27		c_typename	nvarchar(256)	✓		
28		c_vendor	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_powersupply		dbo.ca_ssa_ci_detail	dbo.ca_ssa_powersupply.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_power__id__420DC656

	Key name	Columns	Description
	PK__ca_ssa_p__3213E83F40257DE4	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_printer

Schema	dbo
Name	ca_ssa_printer

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		

	PK	Name	Data type	Null	Attributes	Description
3		c_administrativestatus	nvarchar(64)	✓		
4		c_assetnumber	nvarchar(64)	✓		
5		c_deviceassetnumber	nvarchar(64)	✓		
6		c_devicebiossystemid	nvarchar(256)	✓		
7		c_devicednsname	nvarchar(256)	✓		
8		c_deviceipv4address	nvarchar(16)	✓		
9		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
10		c_deviceipv6address	nvarchar(40)	✓		
11		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
12		c_devicemacaddress	nvarchar(18)	✓		
13		c_devicephysserialnumber	nvarchar(64)	✓		
14		c_devicesysname	nvarchar(256)	✓		
15		c_faxnumber	nvarchar(64)	✓		
16		c_instancename	nvarchar(750)	✓		
17		c_iscopier	bit	✓		
18		c_isfax	bit	✓		
19		c_isinmaintenance	bit	✓		
20		c_isscanner	bit	✓		
21		c_label	nvarchar(256)	✓		
22		c_lastmodtimestamp	datetime	✓		
23		c_lastmodusername	nvarchar(256)	✓		
24		c_localprintername	nvarchar(256)	✓		
25		c_mdrelementid	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
26		c_mdrprodinstance	nvarchar(128)	✓		
27		c_mdrproduct	nvarchar(64)	✓		
28		c_model	nvarchar(128)	✓		
29		c_physserialnumber	nvarchar(64)	✓		
30		c_primarydnsname	nvarchar(256)	✓		
31		c_primaryipv4address	nvarchar(16)	✓		
32		c_primaryipv4addresswithdomain	nvarchar(256)	✓		
33		c_primaryipv6address	nvarchar(40)	✓		
34		c_primaryipv6addresswithdomain	nvarchar(256)	✓		
35		c_primarymacaddress	nvarchar(18)	✓		
36		c_supportscolor	bit	✓		
37		c_sysname	nvarchar(256)	✓		
38		c_tags	nvarchar(750)	✓		
39		c_tenantid	nvarchar(256)	✓		
40		c_typename	nvarchar(256)	✓		
41		c_vendor	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_printer		dbo.ca_ssa_ci_detail	dbo.ca_ssa_printer.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_print__id__370627FE

	Key name	Columns	Description
	PK__ca_ssa_p__3213E83F351DDF8C	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_printserver

Schema	dbo
Name	ca_ssa_printserver

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_accessedviatcpport	int	✓		
4		c_administrativestatus	nvarchar(64)	✓		
5		c_deviceassetnumber	nvarchar(64)	✓		
6		c_devicebiossystemid	nvarchar(256)	✓		
7		c_devicednsname	nvarchar(256)	✓		
8		c_deviceipv4address	nvarchar(16)	✓		
9		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
10		c_deviceipv6address	nvarchar(40)	✓		
11		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
12		c_devicemacaddress	nvarchar(18)	✓		
13		c_devicephysserialnumber	nvarchar(64)	✓		
14		c_devicesysname	nvarchar(256)	✓		
15		c_instancename	nvarchar(750)	✓		
16		c_isinmaintenance	bit	✓		
17		c_label	nvarchar(256)	✓		
18		c_lastmodtimestamp	datetime	✓		

	PK	Name	Data type	Null	Attributes	Description
19		c_lastmodusernaame	nvarchar(256)	✓		
20		c_mdrelementid	nvarchar(256)	✓		
21		c_mdrprodinstance	nvarchar(128)	✓		
22		c_mdrproduct	nvarchar(64)	✓		
23		c_processdistinguishingid	nvarchar(256)	✓		
24		c_processid	int	✓		
25		c_productname	nvarchar(128)	✓		
26		c_tags	nvarchar(750)	✓		
27		c_tenantid	nvarchar(256)	✓		
28		c_typename	nvarchar(256)	✓		
29		c_vendor	nvarchar(256)	✓		
30		c_version	nvarchar(64)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_printserver		dbo.ca_ssa_ci_detail	dbo.ca_ssa_printserver.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_print__id__324172E1

	Key name	Columns	Description
	PK__ca_ssa_p__3213E83F30592A6F	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_problem

Schema	dbo
Name	ca_ssa_problem

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_derived	bit	✓		
3		c_assignedid	nvarchar(128)	✓		
4		c_currency	nvarchar(64)	✓		
5		c_dollarcost	float	✓		
6		c_impact	nvarchar(64)	✓		
7		c_instancename	nvarchar(750)	✓		
8		c_isescalated	bit	✓		
9		c_istemplate	bit	✓		
10		c_label	nvarchar(256)	✓		
11		c_lastmodtimestamp	datetime	✓		
12		c_lastmodusername	nvarchar(256)	✓		
13		c_mdrelementid	nvarchar(256)	✓		
14		c_mdrprodinstance	nvarchar(128)	✓		
15		c_mdrproduct	nvarchar(64)	✓		
16		c_priority	nvarchar(64)	✓		
17		c_problemcategory	nvarchar(64)	✓		
18		c_problemstatus	nvarchar(64)	✓		
19		c_rootcause	nvarchar(64)	✓		
20		c_severity	nvarchar(64)	✓		

	PK	Name	Data type	Null	Attributes	Description
21		c_tags	nvarchar(750)	✓		
22		c_tenantid	nvarchar(256)	✓		
23		c_typename	nvarchar(256)	✓		
24		c_urgency	nvarchar(64)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_problem	✈	dbo.ca_ssa_ci_detail	dbo.ca_ssa_problem.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_probl__id__414EAC47

	Key name	Columns	Description
🔑	PK__ca_ssa_p__3213E83F3F6663D5	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_processor

Schema	dbo
Name	ca_ssa_processor

	PK	Name	Data type	Null	Attributes	Description
1	🔑	id	binary(16)			
2		c_derived	bit	✓		
3		c_administrativestatus	nvarchar(64)	✓		
4		c_containingindex	int	✓		
5		c_deviceassetnumber	nvarchar(64)	✓		
6		c_devicebiossystemid	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
7		c_devicednsname	nvarchar(256)	✓		
8		c_deviceipv4address	nvarchar(16)	✓		
9		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
10		c_deviceipv6address	nvarchar(40)	✓		
11		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
12		c_devicemacaddress	nvarchar(18)	✓		
13		c_devicephysserialnumber	nvarchar(64)	✓		
14		c_devicesysname	nvarchar(256)	✓		
15		c_instancename	nvarchar(750)	✓		
16		c_isinmaintenance	bit	✓		
17		c_isphysical	bit	✓		
18		c_label	nvarchar(256)	✓		
19		c_lastmodtimestamp	datetime	✓		
20		c_lastmodusername	nvarchar(256)	✓		
21		c_mdrelementid	nvarchar(256)	✓		
22		c_mdrprodinstance	nvarchar(128)	✓		
23		c_mdrproduct	nvarchar(64)	✓		
24		c_model	nvarchar(128)	✓		
25		c_osnumeric	int	✓		
26		c_processortype	nvarchar(64)	✓		
27		c_speedinghz	float	✓		
28		c_tags	nvarchar(750)	✓		
29		c_tenantid	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
30		c_typename	nvarchar(256)	✓		
31		c_vendor	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_processor		dbo.ca_ssa_ci_detail	dbo.ca_ssa_processor.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_proce__id__6C390A4C

	Key name	Columns	Description
	PK__ca_ssa_p__3213E83F6A50C1DA	id	

Name
dbo.ca_ssa_ci_detail

✓

dbo.ca_ssa_project

Schema	dbo
Name	ca_ssa_project

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_actualtodatecost	float	✓		
4		c_approvalstatus	nvarchar(64)	✓		
5		c_businessalignmentdegree	nvarchar(64)	✓		
6		c_currency	nvarchar(64)	✓		
7		c_instancename	nvarchar(750)	✓		

	PK	Name	Data type	Null	Attributes	Description
8		c_isactive	bit	✓		
9		c_label	nvarchar(256)	✓		
10		c_lastmodtimestamp	datetime	✓		
11		c_lastmodusername	nvarchar(256)	✓		
12		c_mdrelementid	nvarchar(256)	✓		
13		c_mdrprodinstance	nvarchar(128)	✓		
14		c_mdrproduct	nvarchar(64)	✓		
15		c_plannedcost	float	✓		
16		c_progress	nvarchar(64)	✓		
17		c_projectid	nvarchar(256)	✓		
18		c_projectpriority	tinyint	✓		
19		c_remainingcost	float	✓		
20		c_riskdegree	nvarchar(64)	✓		
21		c_tags	nvarchar(750)	✓		
22		c_tenantid	nvarchar(256)	✓		
23		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_project		dbo.ca_ssa_ci_detail	dbo.ca_ssa_project.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_proje__id__297722B6

	Key name	Columns	Description
	PK__ca_ssa_p__3213E83F278EDA44	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_provisionedsoftware

Schema	dbo
Name	ca_ssa_provisionedsoftware

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_derived	bit	✓		
3		c_administrativestatus	nvarchar(64)	✓		
4		c_deviceassetnumber	nvarchar(64)	✓		
5		c_devicebiossystemid	nvarchar(256)	✓		
6		c_devicednsname	nvarchar(256)	✓		
7		c_deviceipv4address	nvarchar(16)	✓		
8		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
9		c_deviceipv6address	nvarchar(40)	✓		
10		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
11		c_devicemacaddress	nvarchar(18)	✓		
12		c_devicephysserialnumber	nvarchar(64)	✓		
13		c_devicesysname	nvarchar(256)	✓		
14		c_instancename	nvarchar(750)	✓		
15		c_isinmaintenance	bit	✓		
16		c_islocal	bit	✓		
17		c_label	nvarchar(256)	✓		
18		c_lastmodtimestamp	datetime	✓		

	PK	Name	Data type	Null	Attributes	Description
19		c_lastmodusernaame	nvarchar(256)	✓		
20		c_locales	nvarchar(750)	✓		
21		c_mdrelementid	nvarchar(256)	✓		
22		c_mdrprodinstance	nvarchar(128)	✓		
23		c_mdrproduct	nvarchar(64)	✓		
24		c_osenvironments	nvarchar(750)	✓		
25		c_processorenvironments	nvarchar(750)	✓		
26		c_productname	nvarchar(128)	✓		
27		c_provisionedforitem	nvarchar(128)	✓		
28		c_provisioningmethod	nvarchar(64)	✓		
29		c_releasetype	nvarchar(64)	✓		
30		c_softwarecategories	nvarchar(750)	✓		
31		c_softwarepathurl	nvarchar(MAX)	✓		
32		c_tags	nvarchar(750)	✓		
33		c_tenantid	nvarchar(256)	✓		
34		c_typename	nvarchar(256)	✓		
35		c_vendor	nvarchar(256)	✓		
36		c_version	nvarchar(64)	✓		
37		c_virtualizationenvironments	nvarchar(750)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_provisionedsoftware		dbo.ca_ssa_ci_detail	dbo.ca_ssa_provisionedsoftware.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_provi__id__16644E42

	Key name	Columns	Description
	PK__ca_ssa_p__3213E83F147C05D0	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_request

Schema	dbo
Name	ca_ssa_request

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit			
3		c_assignedid	nvarchar(128)			
4		c_impact	nvarchar(64)			
5		c_instancename	nvarchar(750)			
6		c_isreturnedtoservice	bit			
7		c_istemplate	bit			
8		c_label	nvarchar(256)			
9		c_lastmodtimestamp	datetime			
10		c_lastmodusername	nvarchar(256)			
11		c_mdrelementid	nvarchar(256)			
12		c_mdrprodinstance	nvarchar(128)			
13		c_mdrproduct	nvarchar(64)			

	PK	Name	Data type	Null	Attributes	Description
14		c_priority	nvarchar(64)	✓		
15		c_requestcategory	nvarchar(64)	✓		
16		c_requeststatus	nvarchar(64)	✓		
17		c_severity	nvarchar(64)	✓		
18		c_tags	nvarchar(750)	✓		
19		c_tenantid	nvarchar(256)	✓		
20		c_typename	nvarchar(256)	✓		
21		c_urgency	nvarchar(64)	✓		
22		c_useraspect	nvarchar(64)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_request		dbo.ca_ssa_ci_detail	dbo.ca_ssa_request.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_reque__id__257187A8

	Key name	Columns	Description
	PK__ca_ssa_r__3213E83F23893F36	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_resourceserver

Schema	dbo
Name	ca_ssa_resourceserver

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_derived	bit	✓		
3		c_accessedviatcpport	int	✓		
4		c_administrativestatus	nvarchar(64)	✓		
5		c_deviceassetnumber	nvarchar(64)	✓		
6		c_devicebiossystemid	nvarchar(256)	✓		
7		c_devicednsname	nvarchar(256)	✓		
8		c_deviceipv4address	nvarchar(16)	✓		
9		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
10		c_deviceipv6address	nvarchar(40)	✓		
11		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
12		c_devicemacaddress	nvarchar(18)	✓		
13		c_devicephysserialnumber	nvarchar(64)	✓		
14		c_devicesysname	nvarchar(256)	✓		
15		c_instancename	nvarchar(750)	✓		
16		c_isinmaintenance	bit	✓		
17		c_label	nvarchar(256)	✓		
18		c_lastmodtimestamp	datetime	✓		
19		c_lastmodusername	nvarchar(256)	✓		
20		c_mdrelementid	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
21		c_mdrprodinstance	nvarchar(128)	✓		
22		c_mdrproduct	nvarchar(64)	✓		
23		c_processdistinguishingid	nvarchar(256)	✓		
24		c_processid	int	✓		
25		c_productname	nvarchar(128)	✓		
26		c_tags	nvarchar(750)	✓		
27		c_tenantid	nvarchar(256)	✓		
28		c_typename	nvarchar(256)	✓		
29		c_vendor	nvarchar(256)	✓		
30		c_version	nvarchar(64)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_resourceserver		dbo.ca_ssa_ci_detail	dbo.ca_ssa_resourceserver.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_resou__id__61F08603

	Key name	Columns	Description
	PK__ca_ssa_r__3213E83F60083D91	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_router

Schema	dbo
Name	ca_ssa_router

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_administrativestatus	nvarchar(64)	✓		
4		c_assetnumber	nvarchar(64)	✓		
5		c_firmwareversion	nvarchar(64)	✓		
6		c_instancename	nvarchar(750)	✓		
7		c_isinmaintenance	bit	✓		
8		c_label	nvarchar(256)	✓		
9		c_lastmodtimestamp	datetime	✓		
10		c_lastmodusername	nvarchar(256)	✓		
11		c_mdrelementid	nvarchar(256)	✓		
12		c_mdrprodinstance	nvarchar(128)	✓		
13		c_mdrproduct	nvarchar(64)	✓		
14		c_model	nvarchar(128)	✓		
15		c_physserialnumber	nvarchar(64)	✓		
16		c_primarydnsname	nvarchar(256)	✓		
17		c_primaryipv4address	nvarchar(16)	✓		
18		c_primaryipv4addresswithdomain	nvarchar(256)	✓		
19		c_primaryipv6address	nvarchar(40)	✓		
20		c_primaryipv6addresswithdomain	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
21		c_primarymacaddress	nvarchar(18)	✓		
22		c_routingprotocoltypes	nvarchar(128)	✓		
23		c_routingredundancytype	nvarchar(64)	✓		
24		c_sysname	nvarchar(256)	✓		
25		c_tags	nvarchar(750)	✓		
26		c_tenantid	nvarchar(256)	✓		
27		c_typename	nvarchar(256)	✓		
28		c_vendor	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_router		dbo.ca_ssa_ci_detail	dbo.ca_ssa_router.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_route__id__035179CE

	Key name	Columns	Description
	PK__ca_ssa_r__3213E83F0169315C	id	

Name
dbo.ca_ssa_ci_detail

✓

dbo.ca_ssa_runninghardware

Schema	dbo
Name	ca_ssa_runninghardware

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_derived	bit	✓		
3		c_administrativestatus	nvarchar(64)	✓		
4		c_containingindex	int	✓		
5		c_deviceassetnumber	nvarchar(64)	✓		
6		c_devicebiossystemid	nvarchar(256)	✓		
7		c_devicednsname	nvarchar(256)	✓		
8		c_deviceipv4address	nvarchar(16)	✓		
9		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
10		c_deviceipv6address	nvarchar(40)	✓		
11		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
12		c_devicemacaddress	nvarchar(18)	✓		
13		c_devicephysserialnumber	nvarchar(64)	✓		
14		c_devicesysname	nvarchar(256)	✓		
15		c_instancename	nvarchar(750)	✓		
16		c_isinmaintenance	bit	✓		
17		c_isphysical	bit	✓		
18		c_label	nvarchar(256)	✓		
19		c_lastmodtimestamp	datetime	✓		
20		c_lastmodusername	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
21		c_mdrelementid	nvarchar(256)	✓		
22		c_mdrprodinstance	nvarchar(128)	✓		
23		c_mdrproduct	nvarchar(64)	✓		
24		c_model	nvarchar(128)	✓		
25		c_osnumeric	int	✓		
26		c_tags	nvarchar(750)	✓		
27		c_tenantid	nvarchar(256)	✓		
28		c_typename	nvarchar(256)	✓		
29		c_vendor	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_runninghardware		dbo.ca_ssa_ci_detail	dbo.ca_ssa_runninghardware.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_runni__id__125EB334

	Key name	Columns	Description
	PK__ca_ssa_r__3213E83F10766AC2	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_runningsoftware

Schema	dbo
Name	ca_ssa_runningsoftware

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_accessedviatcpport	int	✓		

	PK	Name	Data type	Null	Attributes	Description
4		c_administrativestatus	nvarchar(64)	✓		
5		c_deviceassetnumber	nvarchar(64)	✓		
6		c_devicebiossystemid	nvarchar(256)	✓		
7		c_devicednsname	nvarchar(256)	✓		
8		c_deviceipv4address	nvarchar(16)	✓		
9		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
10		c_deviceipv6address	nvarchar(40)	✓		
11		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
12		c_devicemacaddress	nvarchar(18)	✓		
13		c_devicephysserialnumber	nvarchar(64)	✓		
14		c_devicesysname	nvarchar(256)	✓		
15		c_instancename	nvarchar(750)	✓		
16		c_isinmaintenance	bit	✓		
17		c_label	nvarchar(256)	✓		
18		c_lastmodtimestamp	datetime	✓		
19		c_lastmodusername	nvarchar(256)	✓		
20		c_mdrelementid	nvarchar(256)	✓		
21		c_mdrprodinstance	nvarchar(128)	✓		
22		c_mdrproduct	nvarchar(64)	✓		
23		c_processdistinguishingid	nvarchar(256)	✓		
24		c_processid	int	✓		
25		c_productname	nvarchar(128)	✓		
26		c_tags	nvarchar(750)	✓		

	PK	Name	Data type	Null	Attributes	Description
27		c_tenantid	nvarchar(256)	✓		
28		c_typename	nvarchar(256)	✓		
29		c_vendor	nvarchar(256)	✓		
30		c_version	nvarchar(64)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_runningsoftware		dbo.ca_ssa_ci_detail	dbo.ca_ssa_runningsoftware.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_runni__id__7F4BDEC0

	Key name	Columns	Description
	PK__ca_ssa_r__3213E83F7D63964E	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_securityserver

Schema	dbo
Name	ca_ssa_securityserver

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_accessedviatcpport	int	✓		
4		c_administrativestatus	nvarchar(64)	✓		
5		c_deviceassetnumber	nvarchar(64)	✓		
6		c_devicebiossystemid	nvarchar(256)	✓		
7		c_devicednsname	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
8		c_deviceipv4address	nvarchar(16)	✓		
9		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
10		c_deviceipv6address	nvarchar(40)	✓		
11		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
12		c_devicemacaddress	nvarchar(18)	✓		
13		c_devicephysserialnumber	nvarchar(64)	✓		
14		c_devicesysname	nvarchar(256)	✓		
15		c_instancename	nvarchar(750)	✓		
16		c_isinmaintenance	bit	✓		
17		c_label	nvarchar(256)	✓		
18		c_lastmodtimestamp	datetime	✓		
19		c_lastmodusername	nvarchar(256)	✓		
20		c_mdrelementid	nvarchar(256)	✓		
21		c_mdrprodinstance	nvarchar(128)	✓		
22		c_mdrproduct	nvarchar(64)	✓		
23		c_processdistinguishingid	nvarchar(256)	✓		
24		c_processid	int	✓		
25		c_productname	nvarchar(128)	✓		
26		c_securitycapabilities	nvarchar(256)	✓		
27		c_tags	nvarchar(750)	✓		
28		c_tenantid	nvarchar(256)	✓		
29		c_typename	nvarchar(256)	✓		
30		c_vendor	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
31		c_version	nvarchar(64)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_securityserver		dbo.ca_ssa_ci_detail	dbo.ca_ssa_securityserver.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_secur__id__5D2BD0E6

	Key name	Columns	Description
	PK__ca_ssa_s__3213E83F5B438874	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_service

Schema	dbo
Name	ca_ssa_service

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_accessedviatcpport	int	✓		
4		c_administrativestatus	nvarchar(64)	✓		
5		c_availabilityend	nvarchar(MAX)	✓		
6		c_availabilitystart	nvarchar(MAX)	✓		
7		c_businessimpact	tinyint	✓		
8		c_businessrisk	tinyint	✓		
9		c_instancename	nvarchar(750)	✓		
10		c_isinmaintenance	bit	✓		

	PK	Name	Data type	Null	Attributes	Description
11		c_label	nvarchar(256)	✓		
12		c_lastmodtimestamp	datetime	✓		
13		c_lastmodusername	nvarchar(256)	✓		
14		c_mdrelementid	nvarchar(256)	✓		
15		c_mdrprodinstance	nvarchar(128)	✓		
16		c_mdrproduct	nvarchar(64)	✓		
17		c_servicecapabilities	nvarchar(750)	✓		
18		c_servicelifecycle state	nvarchar(64)	✓		
19		c_servicename	nvarchar(256)	✓		
20		c_serviceversion	nvarchar(64)	✓		
21		c_tags	nvarchar(750)	✓		
22		c_tenantid	nvarchar(256)	✓		
23		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_service		dbo.ca_ssa_ci_detail	dbo.ca_ssa_service.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_servi__id__2EFAF1E2

	Key name	Columns	Description
	PK__ca_ssa_s__3213E83F2D12A970	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_servicespecification

Schema	dbo
Name	ca_ssa_servicespecification

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_derived	bit	✓		
3		c_administrativestatus	nvarchar(64)	✓		
4		c_availabilityend	nvarchar(MAX)	✓		
5		c_availabilitystart	nvarchar(MAX)	✓		
6		c_businessimpact	tinyint	✓		
7		c_businessrisk	tinyint	✓		
8		c_instancename	nvarchar(750)	✓		
9		c_isinmaintenance	bit	✓		
10		c_label	nvarchar(256)	✓		
11		c_lastmodtimestamp	datetime	✓		
12		c_lastmodusername	nvarchar(256)	✓		
13		c_mdrelementid	nvarchar(256)	✓		
14		c_mdrprodinstance	nvarchar(128)	✓		
15		c_mdrproduct	nvarchar(64)	✓		
16		c_servicecapabilities	nvarchar(750)	✓		
17		c_specificationlifecyclestate	nvarchar(64)	✓		
18		c_specificationname	nvarchar(256)	✓		
19		c_specificationversion	nvarchar(64)	✓		
20		c_tags	nvarchar(750)	✓		
21		c_tenantid	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
22		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_servicespecification		dbo.ca_ssa_ci_detail	dbo.ca_ssa_servicespecification.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_servi__id__6774552F

	Key name	Columns	Description
	PK__ca_ssa_s__3213E83F658C0CBD	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_snmpv1access

Schema	dbo
Name	ca_ssa_snmpv1access

Columns

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_accessid	nvarchar(256)	✓		
4		c_getcommunitystring	nvarchar(750)	✓		
5		c_instancename	nvarchar(750)	✓		
6		c_label	nvarchar(256)	✓		
7		c_lastmodtimestamp	datetime	✓		
8		c_lastmodusername	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
9		c_managementipv4address	nvarchar(16)	✓		
10		c_managementipv6address	nvarchar(40)	✓		
11		c_managementport	int	✓		
12		c_mdrelementid	nvarchar(256)	✓		
13		c_mdrprodinstance	nvarchar(128)	✓		
14		c_mdrproduct	nvarchar(64)	✓		
15		c_setcommunitystring	nvarchar(750)	✓		
16		c_tags	nvarchar(750)	✓		
17		c_tenantid	nvarchar(256)	✓		
18		c_typename	nvarchar(256)	✓		

Relations

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_snmpv1access		dbo.ca_ssa_ci_detail	dbo.ca_ssa_snmpv1access.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_snmpv__id__33BFA6FF

Unique keys

	Key name	Columns	Description
	PK__ca_ssa_s__3213E83F31D75E8D	id	

Uses

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_snmpv3access

Schema	dbo
Name	ca_ssa_snmpv3access

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)	✓		
2		c_derived	bit	✓		
3		c_accessid	nvarchar(256)	✓		
4		c_authentication key	nvarchar(64)	✓		
5		c_authenticationprotocol	nvarchar(64)	✓		
6		c_instancename	nvarchar(750)	✓		
7		c_label	nvarchar(256)	✓		
8		c_lastmodtimestamp	datetime	✓		
9		c_lastmodusername	nvarchar(256)	✓		
10		c_managementipv4address	nvarchar(16)	✓		
11		c_managementipv6address	nvarchar(40)	✓		
12		c_managementport	int	✓		
13		c_mdrelementid	nvarchar(256)	✓		
14		c_mdrprodinstance	nvarchar(128)	✓		
15		c_mdrproduct	nvarchar(64)	✓		
16		c_privacykey	nvarchar(64)	✓		
17		c_privacyprotocol	nvarchar(64)	✓		
18		c_tags	nvarchar(750)	✓		
19		c_tenantid	nvarchar(256)	✓		
20		c_typename	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
21		c_username	nvarchar(64)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_snmpv3access		dbo.ca_ssa_ci_detail	dbo.ca_ssa_snmpv3access.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_snmpv__id__66B53B20

	Key name	Columns	Description
	PK__ca_ssa_s__3213E83F64CCF2AE	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_softwarecomponent

Schema	dbo
Name	ca_ssa_softwarecomponent

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_componentname	nvarchar(256)	✓		
4		c_componenttype	nvarchar(64)	✓		
5		c_deviceassetnumber	nvarchar(64)	✓		
6		c_devicebiossystemid	nvarchar(256)	✓		
7		c_devicednsname	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
8		c_deviceipv4address	nvarchar(16)	✓		
9		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
10		c_deviceipv6address	nvarchar(40)	✓		
11		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
12		c_devicemacaddress	nvarchar(18)	✓		
13		c_devicephysserialnumber	nvarchar(64)	✓		
14		c_devicesysname	nvarchar(256)	✓		
15		c_instancename	nvarchar(750)	✓		
16		c_islogical	bit	✓		
17		c_label	nvarchar(256)	✓		
18		c_lastmodtimestamp	datetime	✓		
19		c_lastmodusername	nvarchar(256)	✓		
20		c_mdrelementid	nvarchar(256)	✓		
21		c_mdrprodinstance	nvarchar(128)	✓		
22		c_mdrproduct	nvarchar(64)	✓		
23		c_tags	nvarchar(750)	✓		
24		c_tenantid	nvarchar(256)	✓		
25		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_softwarecomponent		dbo.ca_ssa_ci_detail	dbo.ca_ssa_softwarecomponent.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_softw__id__546180BB

	Key name	Columns	Description
	PK__ca_ssa_s__3213E83F52793849	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_storagearray

Schema	dbo
Name	ca_ssa_storagearray

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_administrativestatus	nvarchar(64)	✓		
4		c_assetnumber	nvarchar(64)	✓		
5		c_drivetypes	nvarchar(256)	✓		
6		c_harddrivecapacityingb	float	✓		
7		c_instancename	nvarchar(750)	✓		
8		c_isinmaintenance	bit	✓		
9		c_label	nvarchar(256)	✓		
10		c_lastmodtimestamp	datetime	✓		
11		c_lastmodusername	nvarchar(256)	✓		
12		c_mdrelementid	nvarchar(256)	✓		
13		c_mdrprodinstance	nvarchar(128)	✓		
14		c_mdrproduct	nvarchar(64)	✓		

	PK	Name	Data type	Null	Attributes	Description
15		c_model	nvarchar(128)	✓		
16		c_numberoftapedrives	int	✓		
17		c_physserialnumber	nvarchar(64)	✓		
18		c_primarydnsname	nvarchar(256)	✓		
19		c_primaryipv4address	nvarchar(16)	✓		
20		c_primaryipv4addresswithdomain	nvarchar(256)	✓		
21		c_primaryipv6address	nvarchar(40)	✓		
22		c_primaryipv6addresswithdomain	nvarchar(256)	✓		
23		c_primarymacaddress	nvarchar(18)	✓		
24		c_sysname	nvarchar(256)	✓		
25		c_tags	nvarchar(750)	✓		
26		c_tenantid	nvarchar(256)	✓		
27		c_typename	nvarchar(256)	✓		
28		c_vendor	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_storagearray	>	dbo.ca_ssa_ci_detail	dbo.ca_ssa_storagearray.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_stora__id__6B79F03D

	Key name	Columns	Description
🔑	PK__ca_ssa_s__3213E83F6991A7CB	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_switch

Schema	dbo
Name	ca_ssa_switch

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_derived	bit	✓		
3		c_administrativestatus	nvarchar(64)	✓		
4		c_assetnumber	nvarchar(64)	✓		
5		c_firmwareversion	nvarchar(64)	✓		
6		c_instancename	nvarchar(750)	✓		
7		c_isinmaintenance	bit	✓		
8		c_label	nvarchar(256)	✓		
9		c_lastmodtimestamp	datetime	✓		
10		c_lastmodusername	nvarchar(256)	✓		
11		c_mdrelementid	nvarchar(256)	✓		
12		c_mdrprodinstance	nvarchar(128)	✓		
13		c_mdrproduct	nvarchar(64)	✓		
14		c_model	nvarchar(128)	✓		
15		c_physserialnumber	nvarchar(64)	✓		
16		c_primarydnsname	nvarchar(256)	✓		
17		c_primaryipv4address	nvarchar(16)	✓		
18		c_primaryipv4addresswithdomain	nvarchar(256)	✓		
19		c_primaryipv6address	nvarchar(40)	✓		
20		c_primaryipv6addresswithdomain	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
21		c_primarymacaddress	nvarchar(18)	✓		
22		c_sysname	nvarchar(256)	✓		
23		c_tags	nvarchar(750)	✓		
24		c_tenantid	nvarchar(256)	✓		
25		c_typename	nvarchar(256)	✓		
26		c_vendor	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_switch		dbo.ca_ssa_ci_detail	dbo.ca_ssa_switch.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_switc__id__33008CF0

	Key name	Columns	Description
	PK__ca_ssa_s__3213E83F3118447E	id	

Name	
dbo.ca_ssa_switch	

dbo.ca_ssa_tablespace

Schema	dbo
Name	ca_ssa_tablespace

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_administrativestatus	nvarchar(64)	✓		

	PK	Name	Data type	Null	Attributes	Description
4		c_dbinstancename	nvarchar(256)	✓		
5		c_databasename	nvarchar(256)	✓		
6		c_deviceassetnumber	nvarchar(64)	✓		
7		c_devicebiossystemid	nvarchar(256)	✓		
8		c_devicednsname	nvarchar(256)	✓		
9		c_deviceipv4address	nvarchar(16)	✓		
10		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
11		c_deviceipv6address	nvarchar(40)	✓		
12		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
13		c_devicemacaddress	nvarchar(18)	✓		
14		c_devicephysserialnumber	nvarchar(64)	✓		
15		c_devicesysname	nvarchar(256)	✓		
16		c_instancename	nvarchar(750)	✓		
17		c_isinmaintenance	bit	✓		
18		c_label	nvarchar(256)	✓		
19		c_lastmodtimestamp	datetime	✓		
20		c_lastmodusername	nvarchar(256)	✓		
21		c_mdrelementid	nvarchar(256)	✓		
22		c_mdrprodinstance	nvarchar(128)	✓		
23		c_mdrproduct	nvarchar(64)	✓		
24		c_tablespacename	nvarchar(256)	✓		
25		c_tags	nvarchar(750)	✓		
26		c_tenantid	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
27		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_tablespace		dbo.ca_ssa_ci_detail	dbo.ca_ssa_tablespace.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_table__id__7E8CC4B1

	Key name	Columns	Description
	PK__ca_ssa_t__3213E83F7CA47C3F	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_tags

Schema	dbo
Name	ca_ssa_tags

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		tagname	varchar(100)			
3		tagvalue	nvarchar(MAX)	✓		

	Key name	Columns	Description
	PK_ca_ssa_tags	id, tagname	

Name
dbo.ca_ssa_tags
dbo.PurgeClearedAlerts

dbo.ca_ssa_transactioncontext

Schema	dbo
Name	ca_ssa_transactioncontext

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_derived	bit	✓		
3		c_contextname	nvarchar(256)	✓		
4		c_contexttype	nvarchar(64)	✓		
5		c_instancename	nvarchar(750)	✓		
6		c_label	nvarchar(256)	✓		
7		c_lastmodtimestamp	datetime	✓		
8		c_lastmodusername	nvarchar(256)	✓		
9		c_mdrelementid	nvarchar(256)	✓		
10		c_mdrprodinstance	nvarchar(128)	✓		
11		c_mdrproduct	nvarchar(64)	✓		
12		c_tags	nvarchar(750)	✓		
13		c_tenantid	nvarchar(256)	✓		
14		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_transactioncontext	→	dbo.ca_ssa_ci_detail	dbo.ca_ssa_transactioncontext.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_trans__id__37C5420D

	Key name	Columns	Description
✓	PK__ca_ssa_t__3213E83F35DCF99B	id	

Name	
dbo.ca_ssa_transactioncontext	

dbo.ca_ssa_transactionsegment

Schema	dbo
Name	ca_ssa_transactionsegment

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_instancename	nvarchar(750)	✓		
4		c_label	nvarchar(256)	✓		
5		c_lastmodtimestamp	datetime	✓		
6		c_lastmodusername	nvarchar(256)	✓		
7		c_mdrelementid	nvarchar(256)	✓		
8		c_mdrprodinstance	nvarchar(128)	✓		
9		c_mdrproduct	nvarchar(64)	✓		
10		c_segmentname	nvarchar(256)	✓		
11		c_tags	nvarchar(750)	✓		
12		c_tenantid	nvarchar(256)	✓		
13		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_transactionsegment		dbo.ca_ssa_ci_detail	dbo.ca_ssa_transactionsegment.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_trans__id__1A69E950

	Key name	Columns	Description
	PK__ca_ssa_t__3213E83F1881A0DE	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_transactionserver

Schema	dbo
Name	ca_ssa_transactionserver

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_accessedviatcpport	int	✓		
4		c_administrativestatus	nvarchar(64)	✓		
5		c_deviceassetnumber	nvarchar(64)	✓		
6		c_devicebiossystemid	nvarchar(256)	✓		
7		c_devicednsname	nvarchar(256)	✓		
8		c_deviceipv4address	nvarchar(16)	✓		
9		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
10		c_deviceipv6address	nvarchar(40)	✓		
11		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
12		c_devicemacaddress	nvarchar(18)	✓		
13		c_devicephysserialnumber	nvarchar(64)	✓		
14		c_devicesysname	nvarchar(256)	✓		
15		c_instancename	nvarchar(750)	✓		

	PK	Name	Data type	Null	Attributes	Description
16		c_isinmaintenance	bit	✓		
17		c_label	nvarchar(256)	✓		
18		c_lastmodtimestamp	datetime	✓		
19		c_lastmodusername	nvarchar(256)	✓		
20		c_mdrelementid	nvarchar(256)	✓		
21		c_mdrprodinstance	nvarchar(128)	✓		
22		c_mdrproduct	nvarchar(64)	✓		
23		c_processdistinguishingid	nvarchar(256)	✓		
24		c_processid	int	✓		
25		c_productname	nvarchar(128)	✓		
26		c_tags	nvarchar(750)	✓		
27		c_tenantid	nvarchar(256)	✓		
28		c_typename	nvarchar(256)	✓		
29		c_vendor	nvarchar(256)	✓		
30		c_version	nvarchar(64)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_transactionserver		dbo.ca_ssa_ci_detail	dbo.ca_ssa_transactionserver.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_trans__id__1B29035F

	Key name	Columns	Description
	PK__ca_ssa_t__3213E83F1940BAED	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_virtualizationmanager

Schema	dbo
Name	ca_ssa_virtualizationmanager

	PK	Name	Data type	Null	Attributes	Description
1		id	binary(16)			
2		c_derived	bit	✓		
3		c_accessedviatcpport	int	✓		
4		c_administrativestatus	nvarchar(64)	✓		
5		c_apiversion	nvarchar(64)	✓		
6		c_deviceassetnumber	nvarchar(64)	✓		
7		c_devicebiossystemid	nvarchar(256)	✓		
8		c_devicednsname	nvarchar(256)	✓		
9		c_deviceipv4address	nvarchar(16)	✓		
10		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
11		c_deviceipv6address	nvarchar(40)	✓		
12		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
13		c_devicemacaddress	nvarchar(18)	✓		
14		c_devicephysserialnumber	nvarchar(64)	✓		
15		c_devicesysname	nvarchar(256)	✓		
16		c_instancename	nvarchar(750)	✓		
17		c_isinmaintenance	bit	✓		
18		c_label	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
19		c_lastmodtimestamp	datetime	✓		
20		c_lastmodusername	nvarchar(256)	✓		
21		c_mdrelementid	nvarchar(256)	✓		
22		c_mdrprodinstance	nvarchar(128)	✓		
23		c_mdrproduct	nvarchar(64)	✓		
24		c_processdistinguishingid	nvarchar(256)	✓		
25		c_processid	int	✓		
26		c_productname	nvarchar(128)	✓		
27		c_tags	nvarchar(750)	✓		
28		c_tenantid	nvarchar(256)	✓		
29		c_typename	nvarchar(256)	✓		
30		c_vendor	nvarchar(256)	✓		
31		c_version	nvarchar(64)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_virtualizationmanager		dbo.ca_ssa_ci_detail	dbo.ca_ssa_virtualizationmanager.ci_id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_virtu__id__75C27486

	Key name	Columns	Description
	PK__ca_ssa_v__3213E83F73DA2C14	id	

Name
dbo.ca_ssa_ci_detail

dbo.ca_ssa_virtualsystem

Schema	dbo
Name	ca_ssa_virtualsystem

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_derived	bit	✓		
3		c_administrativestatus	nvarchar(64)	✓		
4		c_biossystemid	nvarchar(256)	✓		
5		c_computername	nvarchar(256)	✓		
6		c_instancename	nvarchar(750)	✓		
7		c_isinmaintenance	bit	✓		
8		c_label	nvarchar(256)	✓		
9		c_lastmodtimestamp	datetime	✓		
10		c_lastmodusername	nvarchar(256)	✓		
11		c_mdrelementid	nvarchar(256)	✓		
12		c_mdrprodinstance	nvarchar(128)	✓		
13		c_mdrproduct	nvarchar(64)	✓		
14		c_memoryingb	float	✓		
15		c_numberofcores	int	✓		
16		c_primarydnsname	nvarchar(256)	✓		
17		c_primaryipv4address	nvarchar(16)	✓		
18		c_primaryipv4addresswithdomain	nvarchar(256)	✓		
19		c_primaryipv6address	nvarchar(40)	✓		
20		c_primaryipv6addresswithdomain	nvarchar(256)	✓		
21		c_primarymacaddress	nvarchar(18)	✓		

	PK	Name	Data type	Null	Attributes	Description
22		c_primaryostype	nvarchar(64)	✓		
23		c_primaryosversion	nvarchar(64)	✓		
24		c_processortype	nvarchar(64)	✓		
25		c_storageingb	float	✓		
26		c_tags	nvarchar(750)	✓		
27		c_tenantid	nvarchar(256)	✓		
28		c_typename	nvarchar(256)	✓		
29		c_virtualid	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_virtualsystem		dbo.ca_ssa_ci_detail	dbo.ca_ssa_virtualsystem.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_virtu__id__5DEAEAF5

	Key name	Columns	Description
	PK__ca_ssa_v__3213E83F5C02A283	id	

Name
dbo.ca_ssa_ci_detail

✓

dbo.ca_ssa_vmdatastore

Schema	dbo
Name	ca_ssa_vmdatastore

	PK	Name	Data type	Null	Attributes	Description
1	✓	id	binary(16)			
2		c_derived	bit	✓		
3		c_administrativestatus	nvarchar(64)	✓		
4		c_capacityinmb	float	✓		
5		c_deviceassetnumber	nvarchar(64)	✓		
6		c_devicebiossystemid	nvarchar(256)	✓		
7		c_devicednsname	nvarchar(256)	✓		
8		c_deviceipv4address	nvarchar(16)	✓		
9		c_deviceipv4addresswithdomain	nvarchar(256)	✓		
10		c_deviceipv6address	nvarchar(40)	✓		
11		c_deviceipv6addresswithdomain	nvarchar(256)	✓		
12		c_devicemacaddress	nvarchar(18)	✓		
13		c_devicephysserialnumber	nvarchar(64)	✓		
14		c_devicesysname	nvarchar(256)	✓		
15		c_filepathurl	nvarchar(MAX)	✓		
16		c_instancename	nvarchar(750)	✓		
17		c_isinmaintenance	bit	✓		
18		c_ismultihost	bit	✓		
19		c_label	nvarchar(256)	✓		
20		c_lastmodtimestamp	datetime	✓		
21		c_lastmodusername	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
22		c_mdrelementid	nvarchar(256)	✓		
23		c_mdrprodinstance	nvarchar(128)	✓		
24		c_mdrproduct	nvarchar(64)	✓		
25		c_storename	nvarchar(256)	✓		
26		c_storetype	nvarchar(64)	✓		
27		c_tags	nvarchar(750)	✓		
28		c_tenantid	nvarchar(256)	✓		
29		c_typename	nvarchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.ca_ssa_vmdatastore		dbo.ca_ssa_ci_detail	dbo.ca_ssa_vmdatastore.id = dbo.ca_ssa_ci_detail.id	FK__ca_ssa_vmdat__id__38845C1C

	Key name	Columns	Description
	PK__ca_ssa_v__3213E83F369C13AA	id	

Name
dbo.ca_ssa_ci_detail

dbo.CI

Schema	dbo
Name	CI
Description	Contains details of the managed CIs.

Columns

	PK	Name	Data type	Null	Attributes	Description
1		CIID	bigint		Identity / Auto increment column	
2		InstanceID	nvarchar(256)			
3		DeviceID	nvarchar(256)			
4		ClassID	int			
5		IPAddress	varchar(256)	✓		
6		Severity	int	✓		
7		Vendor	varchar(128)	✓		
8		LastUpdate	datetime	✓		
9		Label	nvarchar(256)			
10		Description	nvarchar(1024)	✓		
11		Calender	varchar(64)	✓		
12		Location	nvarchar(512)	✓		
13		GELString	nvarchar(512)	✓		
14		Offline	tinyint			
15		EnterpriseFlag	tinyint			
16		OLT	int	✓		
17		Granularity	tinyint			
18		SecurityGroups	varchar(2048)	✓		
19		SecurityGroupFlag	tinyint	✓		
20		BusinessPriority	int	✓	Default: -1	
21		Category	nvarchar(256)	✓		
22		UserDefinedCol1	nvarchar(256)	✓		
23		UserDefinedCol2	nvarchar(256)	✓		
24		UserDefinedCol3	nvarchar(256)	✓		
25		UserDefinedCol4	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
26		UserDefinedCol5	nvarchar(256)	✓		
27		UserDefinedCol6	nvarchar(256)	✓		
28		UserDefinedCol7	nvarchar(256)	✓		
29		UserDefinedCol8	nvarchar(256)	✓		
30		UserDefinedCol9	nvarchar(256)	✓		
31		UserDefinedCol10	nvarchar(256)	✓		
32		Permissions	tinyint	✓		
33		USMNotebookID	varchar(50)	✓		
34		USMSheetID	varchar(50)	✓		

Relations

Foreign table		Primary table	Join	Title / Name / Description
dbo.CIEvent		dbo.CI	dbo.CIEvent.CIID = dbo.CI.CIID	FK_CI_CIEvent
dbo.CIRelationship		dbo.CI	dbo.CIRelationship.ANodeCCID = dbo.CI.CIID	FK_CIRelationship_CI_ANodeCCID
dbo.CIRelationship		dbo.CI	dbo.CIRelationship.BNodeCIID = dbo.CI.CIID	FK_CIRelationship_CI_BNodeCIID
dbo.CISLO		dbo.CI	dbo.CISLO.CIID = dbo.CI.CIID	FK_CISLO_CIID
dbo.CIStaging		dbo.CI	dbo.CIStaging.CIID = dbo.CI.CIID	FK_CI_CIStaging_CIID
dbo.ScheduleRelationship		dbo.CI	dbo.ScheduleRelationship.CIID = dbo.CI.CIID	FK_ScheduleRelationship_CIID

Unique keys

	Key name	Columns	Description
	PK_CI	CIID	
	IX_CIINSTANCEID_DEVICEID	InstanceID	

Uses

Name	
• dbo.CIChangeHistory • dbo.Class	
• dbo.CIChangeHistory • dbo.Class	
• dbo.CIChangeHistory • dbo.Class	

Used by

Name	
dbo.CI	
dbo.SecureCI	
dbo.SecureInfrastructureCI	
dbo.SecureServiceCI	
dbo.SecureServices • dbo.SecureUserServices	
dbo.PurgeOutageAlert	
dbo.CIEvent	
dbo.CIRelationship	
dbo.CIRelationship	
dbo.CISLO	
dbo.CIStaging	
dbo.ScheduleRelationship	

dbo.CIChangeHistory

Schema	dbo
Name	CIChangeHistory

Columns

	PK	Name	Data type	Null	Attributes	Description
1		ChangeID	bigint		Identity / Auto increment column	
2		ChangeType	varchar(12)			
3		ChangeTypeID	bigint			

	PK	Name	Data type	Null	Attributes	Description
4		ClassName	varchar(64)	✓		
5		BNodeCIID	bigint	✓		
6		ServiceCIID	bigint	✓		
7		AssociationID	int	✓		
8		ModificationTime	datetime			
9		ModificationType	varchar(20)			
10		ApplicationName	varchar(128)	✓		
11		UserName	nvarchar(128)	✓		

Unique keys

	Key name	Columns	Description
	XPKCIChangeHistory	ChangeID	

Used by

Name	
dbo.CIChangeHistory	
dbo.cleanHistoryData	
dbo.PurgeOutageAlert	
dbo.CI	
dbo.CIRelationship	

dbo.CIEscalationPolicy

Schema	dbo
Name	CIEscalationPolicy

	PK	Name	Data type	Null	Attributes	Description
1	✓	EscalationID	bigint			
2	✓	CIID	bigint			
3		ApplyGlobalPolicy	tinyint	✓	Default: 0	

	Key name	Columns	Description
✓	PK_CIEscalationPolicy	EscalationID, CIID	

dbo.CIEvent

Schema	dbo
Name	CIEvent

	PK	Name	Data type	Null	Attributes	Description
1		ClassID	int			
2	✓	MDRAlarmID	varchar(256)			
3		Severity	int	✓		
4		SituationMessage	nvarchar(256)	✓		
5		ReportedTime	datetime	✓		
6		LoggedTime	datetime	✓		
7		CINamespaceMapID	varchar(256)	✓		
8		CILiCURL	varchar(1024)	✓		
9	✓	CIID	bigint			
10	✓	ConnectorID	int			

Foreign table		Primary table	Join	Title / Name / Description
dbo.CIEvent		dbo.CI	dbo.CIEvent.CIID = dbo.CI.CIID	FK_CI_CIEvent
dbo.CIEvent		dbo.ConnectorConfiguration	dbo.CIEvent.ConnectorID = dbo.ConnectorConfiguration.ConnectorID	FK_ConnectorConfig_CIEvent

	Key name	Columns	Description
	PK_CIEvent	MDRAlarmID, CIID, ConnectorID	

Name
dbo.CI
dbo.ConnectorConfiguration

dbo.CIHealth

Schema	dbo
Name	CIHealth

	PK	Name	Data type	Null	Attributes	Description
1		PeriodID	bigint		Identity / Auto increment column	
2		CIID	bigint			
3		Severity	int			
4		Impact	int	✓		
5		StartPeriod	datetime			
6		EndPeriod	datetime	✓		
7		Mode	int			

	Key name	Columns	Description
	PK_CIHealth	PeriodID	

Name
dbo.CIHealthDiff
dbo.cleanHistoryData

dbo.CIQuality

Schema	dbo
Name	CIQuality

Columns

	PK	Name	Data type	Null	Attributes	Description
1		PeriodID	bigint		Identity / Auto increment column	
2		CIID	bigint			
3		Quality	int			
4		StartPeriod	datetime			
5		EndPeriod	datetime	✓		
6		Mode	int			

Unique keys

	Key name	Columns	Description
	PK_CIQuality	PeriodID	

Used by

Name
dbo.CIQualityDiff
dbo.cleanHistoryData

dbo.CIRelationship

Schema	dbo
Name	CIRelationship

Columns

	PK	Name	Data type	Null	Attributes	Description
1	✓	ANodeCIID	bigint			
2	✓	BNodeCIID	bigint			
3	✓	ServiceCIID	bigint			
4	✓	AssociationID	int			
5		PolicyID	bigint	✓		
6		ServiceImpact	int	✓		
7		Relevance	int	✓		
8		Significance	int	✓		
9		LastUpdate	datetime	✓		
10		AdminStatus	int			
11		USMAssocMapping	int	✓		
12		USMNotebookID	varchar(50)	✓		
13		USMSheetID	varchar(50)	✓		

Relations

Foreign table		Primary table	Join	Title / Name / Description
dbo.CIRelationship	✓	dbo.AssociationType	dbo.CIRelationship.AssociationID = dbo.AssociationType.AssociationID	FK_CIRelationship_AssociationType
dbo.CIRelationship	✓	dbo.CI	dbo.CIRelationship.ANodeCIID = dbo.CI.CIID	FK_CIRelationship_CI_ANodeCCID

Foreign table		Primary table	Join	Title / Name / Description
dbo.CIRelationship		dbo.CI	dbo.CIRelationship.BNodeCIID = dbo.CI.CIID	FK_CIRelationship_CI_BNodeCIID

Unique keys

	Key name	Columns	Description
	PK_CIRelationship	ANodeCIID, BNodeCIID, ServiceCIID, AssociationID	

Uses

Name	
dbo.CIRelationship • dbo.AssociationType • dbo.CI • dbo.CI	
dbo.CIChangeHistory	
dbo.CIChangeHistory	
dbo.CIChangeHistory	

dbo.CIRisk

Schema	dbo
Name	CIRisk

	PK	Name	Data type	Null	Attributes	Description
1		PeriodID	bigint		Identity / Auto increment column	
2		CIID	bigint			
3		Risk	int			
4		StartPeriod	datetime			
5		EndPeriod	datetime	✓		
6		Mode	int			

	Key name	Columns	Description
	PK_CIRisk	PeriodID	

Name
dbo.CIRiskDiff
dbo.cleanHistoryData

dbo.CISLO

Schema	dbo
Name	CISLO

	PK	Name	Data type	Null	Attributes	Description
1		CIID	bigint			
2		SLOID	bigint			

Foreign table		Primary table	Join	Title / Name / Description
dbo.CISLO		dbo.CI	dbo.CISLO.CIID = dbo.CI.CIID	FK_CISLO_CIID
dbo.CISLO		dbo.ServiceLevelObjective	dbo.CISLO.SLOID = dbo.ServiceLevelObjective.SLOID	FK_CISLO_SLOID

	Key name	Columns	Description
	PK_CISLO_RELATIONSHIP	CIID, SLOID	

Name
dbo.CI
dbo.ServiceLevelObjective

dbo.CIStaging

Schema	dbo
Name	CIStaging
Description	Contains details of unmanaged CIs and managed CIs.

Columns

	PK	Name	Data type	Null	Attributes	Description
1	✓	CINamespaceMapID	nvarchar(256)			
2		InstanceID	nvarchar(256)			
3		DeviceID	nvarchar(256)			
4		ClassID	int			
5		IPAddress	varchar(256)	✓		
6		Vendor	nvarchar(128)	✓		
7		Label	nvarchar(256)			
8		Description	nvarchar(1024)	✓		
9		CILiCURL	varchar(1024)	✓		
10	✓	ConnectorID	int			
11		Category	nvarchar(256)	✓		
12		SiloResourceId	nvarchar(256)			
13		CIID	bigint	✓		
14		Key1	nvarchar(256)	✓		
15		Confidence1	int	✓		
16		Key2	nvarchar(256)	✓		
17		Confidence2	int	✓		
18		Key3	nvarchar(256)	✓		
19		Confidence3	int	✓		
20		Key4	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
21		Confidence4	int	✓		
22		Key5	nvarchar(256)	✓		
23		Confidence5	int	✓		
24		Key6	nvarchar(256)	✓		
25		Confidence6	int	✓		
26		Key7	nvarchar(256)	✓		
27		Confidence7	int	✓		
28		Key8	nvarchar(256)	✓		
29		Confidence8	int	✓		
30		Key9	nvarchar(256)	✓		
31		Confidence9	int	✓		
32		Key10	nvarchar(256)	✓		
33		Confidence10	int	✓		
34		Key11	nvarchar(256)	✓		
35		Confidence11	int	✓		
36		Key12	nvarchar(256)	✓		
37		Confidence12	int	✓		
38		Key13	nvarchar(256)	✓		
39		Confidence13	int	✓		
40		Key14	nvarchar(256)	✓		
41		Confidence14	int	✓		
42		Key15	nvarchar(256)	✓		
43		Confidence15	int	✓		
44		Key16	nvarchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
45		Confidence16	int	✓		
46		AdminStatus	tinyint	✓		
47		ModificationTime	datetime	✓		
48		TenantID	nvarchar(256)	✓		
49		UserDefinedCol1	nvarchar(256)	✓		
50		UserDefinedCol2	nvarchar(256)	✓		
51		UserDefinedCol3	nvarchar(256)	✓		
52		UserDefinedCol4	nvarchar(256)	✓		
53		UserDefinedCol5	nvarchar(256)	✓		
54		UserDefinedCol6	nvarchar(256)	✓		
55		UserDefinedCol7	nvarchar(256)	✓		
56		UserDefinedCol8	nvarchar(256)	✓		
57		UserDefinedCol9	nvarchar(256)	✓		
58		UserDefinedCol10	nvarchar(256)	✓		
59		USMNotebookID	nvarchar(50)	✓		
60		USMSheetID	nvarchar(50)	✓		

Relations

Foreign table		Primary table	Join	Title / Name / Description
dbo.CIStaging	⊱	dbo.CI	dbo.CIStaging.CIID = dbo.CI.CIID	FK_CI_CIStaging_CIID
dbo.CIStaging	⊱	dbo.Class	dbo.CIStaging.ClassID = dbo.Class.ClassID	FK_CI_CIStaging_ClassID
dbo.CIStaging	⊱	dbo.ConnectorConfiguration	dbo.CIStaging.ConnectorID = dbo.ConnectorConfiguration.ConnectorID	FK_ConnectorConfig_CIStaging

Unique keys

	Key name	Columns	Description
	PK_CIStaging	CINamespaceMapID, ConnectorID	

Uses

Name	
dbo.CIStaging • dbo.CI • dbo.Class • dbo.ConnectorConfiguration	
dbo.CIStagingTemp	
dbo.CIStagingTemp	
dbo.CIStagingTemp	

dbo.CIStagingTemp

Schema	dbo
Name	CIStagingTemp

Columns

	PK	Name	Data type	Null	Attributes	Description
1		SqID	bigint		Identity / Auto increment column	
2		CINamespaceMapID	nvarchar(256)			
3		InstanceID	nvarchar(256)			
4		DeviceID	nvarchar(256)			
5		ClassID	int			
6		CIPresent	varchar(10)	✓		
7		ConnectorID	int			
8		SiloResourceId	nvarchar(256)			
9		ModificationTime	datetime	✓		
10		USMNotebookID	nvarchar(50)	✓		

	PK	Name	Data type	Null	Attributes	Description
11		USMSheetID	nvarchar(50)	✓		

Unique keys

	Key name	Columns	Description
	PK_CIStagingTemp	SqID	

Used by

Name
dbo.CIStaging

dbo.Class

Schema	dbo
Name	Class

Columns

	PK	Name	Data type	Null	Attributes	Description
1		ClassID	int			
2		ClassName	varchar(64)			
3		OSILayerID	smallint	✓		
4		OSILayerName	varchar(48)	✓		
5		OSIFamilyName	varchar(256)	✓		
6		FamilyID	int	✓		
7		significance	int	✓		
8		SigDefault	smallint	✓		
9		IsTopLevel	smallint	✓		

	PK	Name	Data type	Null	Attributes	Description
10		IsViewed	smallint		Default: 1	
11		IsModeled	smallint		Default: 1	
12		IsAggregating	smallint		Default: 0	
13		SmallIcon	varchar(64)	✓		
14		LargeIcon	varchar(64)	✓		

Relations

Foreign table		Primary table	Join	Title / Name / Description
dbo.Class		dbo.Family	dbo.Class.FamilyID = dbo.Family.FamilyID	FK_Class_Family_FamilyID
dbo.AlertEscalationPolicy		dbo.Class	dbo.AlertEscalationPolicy.ClassID = dbo.Class.ClassID	FK_AlertEscalationActions_Class_ClassID
dbo.Alerts		dbo.Class	dbo.Alerts.ClassID = dbo.Class.ClassID	FK_Alerts_Class_ClassID
dbo.CIStaging		dbo.Class	dbo.CIStaging.ClassID = dbo.Class.ClassID	FK_CI_CIStaging_ClassID

Unique keys

	Key name	Columns	Description
	PK_Class	ClassID	

Uses

Name	
dbo.Class • dbo.Family	

Used by

Name
dbo.CIChange
dbo.AlertEscalationPolicy
dbo.Alerts

Name
dbo.CIStaging

dbo.ClassProperty

Schema	dbo
Name	ClassProperty

	PK	Name	Data type	Null	Attributes	Description
1	✓	QName	varchar(128)			
2		ID	int			
3		Name	varchar(64)			
4		Type	varchar(16)			
5		IsInherited	tinyint			
6		IsExtension	tinyint			
7		DefinedBy	varchar(64)	✓		
8		IsQueriable	tinyint			
9		IsMany	tinyint			
10		IsNullable	tinyint			
11		IsEnum	tinyint			
12		EnumType	varchar(64)	✓		
13		I18nTag	varchar(100)			

	Key name	Columns	Description
✓	PK_ClassProperty	QName	

dbo.ClassRelationship

Schema	dbo
Name	ClassRelationship

	PK	Name	Data type	Null	Attributes	Description
1	✓	ClassRelID	int			
2		ParentClassID	int	✓		

	PK	Name	Data type	Null	Attributes	Description
3		ChildClassID	int	✓		

	Key name	Columns	Description
	PK_ClassRelationship	ClassRelID	

dbo.ConnectorConfiguration

Schema	dbo
b	ConnectorConfiguration

	PK	Name	Data type	Null	Attributes	Description
1		ConnectorID	int		Identity / Auto increment column	
2		ConnectorName	varchar(256)			
3		MDRName	varchar(256)			
4		ConnectorDesc	nvarchar(256)	✓		
5		ConnectorStatus Desc	nvarchar(512)	✓		
6		MDRDesc	nvarchar(256)	✓		
7		Status	int	✓		
8		ConnectorPermis sions	int	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.CIEvent		dbo.ConnectorConfigura tion	dbo.CIEvent.ConnectorID = dbo.ConnectorConfigurationEvent.ConnectorID	FK_ConnectorConfig_CI
dbo.CIStaging		dbo.ConnectorConfigura tion	dbo.CIStaging.ConnectorID = dbo.ConnectorConfigurationStaging.ConnectorID	FK_ConnectorConfig_CI

	Key name	Columns	Description
	XPKConnectorConfiguration	ConnectorID	
	IX_ConnectorConfiguration	ConnectorName, MDRName	

Name
dbo.cleanAlertsFromRemovedConnectors
dbo.PurgeClearedAlerts
dbo.CIEvent
dbo.CIStaging

dbo.ConnectorPopupLauncher

Schema	dbo
Name	ConnectorPopupLauncher

	PK	Name	Data type	Null	Attributes	Description
1		ConnectorID	int			
2		PopupLabelname	varchar(128)			
3		PopupSequence	int	✓		
4		ServerName	nvarchar(128)	✓		
5		PortNumber	int	✓		
6		Protocol	char(6)	✓		
7		UserID	varchar(128)	✓		
8		Password	nvarchar(128)	✓		
9		URL_Launch	varchar(2048)	✓		
10		URL_LaunchContextType	smallint			

	Key name	Columns	Description
	PK_ConnectorPopupLauncher	ConnectorID, URL_LaunchContextType, PopupLabelname	

dbo.Customer

Schema	dbo
Name	Customer

	PK	Name	Data type	Null	Attributes	Description
1		CustomerID	bigint		Identity / Auto increment column	
2		TenantID	nvarchar(256)	✓		
3		Name	nvarchar(256)			
4		Priority	int			
5		Tier	int			
6		Description	nvarchar(1024)			
7		CustomerIdentifier	nvarchar(1024)	✓		
8		CreatedTime	datetime			
9		ConnectorID	int			
10		Deleted	smallint	✓		
11		DeletedTime	datetime	✓		

	Key name	Columns	Description
	PK_Customer	CustomerID	

Name
dbo.SecureCustomers

dbo.CustomerImpact

Schema	dbo
Name	CustomerImpact

	PK	Name	Data type	Null	Attributes	Description
1	✓	ID	bigint		Identity / Auto increment column	
2		AlertID	bigint			
3		CustomerID	bigint			
4		Impact	int			
5		CreatedTime	datetime			
6		Deleted	smallint			
7		DeletedTime	datetime	✓		

	Key name	Columns	Description
✓	PK_CUSTOMERIMPACT	ID	

dbo.CustomerRelationship

Schema	dbo
Name	CustomerRelationship

	PK	Name	Data type	Null	Attributes	Description
1	✓	CustomerID	bigint			
2	✓	ChildID	bigint			
3	✓	ChildType	tinyint			
4	✓	UsmAssociationType	int			
5		Inherited	tinyint			
6	✓	CreatedTime	datetime			
7		Deleted	smallint			
8		DeletedTime	datetime	✓	Default: 0	

	Key name	Columns	Description
	PK_CustomerRelationship	CustomerID, ChildID, ChildType, UsmAssociationType, CreatedTime	

Name
dbo.CustomerServices

dbo.DbAvailHistory

Schema	dbo
Name	DbAvailHistory

	PK	Name	Data type	Null	Attributes	Description
1		DbAvailID	bigint		Identity / Auto increment column	
2		ServiceID	bigint			
3		Availability	int			
4		StartPeriod	datetime			
5		EndPeriod	datetime	✓		
6		Normal	int			
7		Minor	int			
8		Major	int			
9		Critical	int			
10		Down	int			
11		Unknown	int			
12		Maintenance	int			
13		Other	int			

	Key name	Columns	Description
	PK_DbAvailHistory	DbAvailID	
	IX_DbAvailHistory_ServiceID_Start_Endingperiod	ServiceID, StartPeriod, EndPeriod	

Name
dbo.DbAvailDiff
dbo.cleanHistoryData

dbo.DbQualityHistory

Schema	dbo
Name	DbQualityHistory

	PK	Name	Data type	Null	Attributes	Description
1		DbQualityID	bigint		Identity / Auto increment column	
2		ServiceID	bigint			
3		Quality	int			
4		StartPeriod	datetime			
5		EndPeriod	datetime	✔		
6		Normal	int			
7		Minor	int			
8		Major	int			
9		Critical	int			
10		Down	int			
11		Unknown	int			
12		Maintenance	int			
13		Other	int			

	Key name	Columns	Description
	PK_DbQualityHistory	DbQualityID	
	IX_DbQualityHistory_ServiceID_Start_Endingperiod	ServiceID, StartPeriod, EndPeriod	

Name
dbo.DbQualityDiff
dbo.cleanHistoryData

dbo.DbRiskHistory

Schema	dbo
Name	DbRiskHistory

	PK	Name	Data type	Null	Attributes	Description
1		DbRiskID	bigint		Identity / Auto increment column	
2		ServiceID	bigint			
3		Risk	int			
4		StartPeriod	datetime			
5		EndPeriod	datetime	✓		
6		Normal	int			
7		Minor	int			
8		Major	int			
9		Critical	int			
10		Down	int			
11		Unknown	int			
12		Maintenance	int			
13		Other	int			

	Key name	Columns	Description
	PK_DbRiskHistory	DbRiskID	
	IX_DbRiskHistory_ServiceID_Start_Endingperiod	ServiceID, StartPeriod, EndPeriod	

Name
dbo.DbRiskDiff
dbo.cleanHistoryData

dbo.DbSchemaVersion

Schema	dbo
Name	DbSchemaVersion

	PK	Name	Data type	Null	Attributes	Description
1		Version	varchar(1024)			
2		Major_rev	int			
3		Minor_rev	int			

dbo.DbUserPreference

Schema	dbo
Name	DbUserPreference

	PK	Name	Data type	Null	Attributes	Description
1		DbPrefID	bigint		Identity / Auto increment column	
2		Name	nvarchar(128)			
3		Panels	varchar(512)			
4		Preferences	varchar(512)	✓		

	Key name	Columns	Description
	PK_DbUserPreference	DbPrefID	

dbo.EscalationPolicyRelationship

Schema	dbo
Name	EscalationPolicyRelationship

	PK	Name	Data type	Null	Attributes	Description
1		EscalationID	bigint			
2		TypeID	bigint			
3		Type	tinyint			
4		Sequence	int			

	PK	Name	Data type	Null	Attributes	Description
5		ApplyGlobalPolicy	tinyint	✓	Default: 0	

	Key name	Columns	Description
	PK_EscalationPolicyRelationship	EscalationID, TypeID, Type	

dbo.EscalationScheduleRelationship

Schema	dbo
Name	EscalationScheduleRelationship

	PK	Name	Data type	Null	Attributes	Description
1		PolicyID	bigint			
2		ScheduleID	bigint			

	Key name	Columns	Description
	PK_ESCALATIONSCHEDULERELATIONSHIP	PolicyID, ScheduleID	

dbo.Family

Schema	dbo
Name	Family

	PK	Name	Data type	Null	Attributes	Description
1		FamilyID	int		Identity / Auto increment column	
2		FamilyName	varchar(256)			
3		FamilyDesc	varchar(256)	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.Class		dbo.Family	dbo.Class.FamilyID = dbo.Family.FamilyID	FK_Class_Family_FamilyID

	Key name	Columns	Description
	XPKFamily	FamilyID	

Name
dbo.Class

dbo.GEIntegration

Schema	dbo
Name	GEIntegration

	PK	Name	Data type	Null	Attributes	Description
1		CIID	bigint			
2		Latitude	decimal(4, 2)			
3		Longitude	decimal(4, 2)			
4		Offset	int	✓		
5		Altitude	decimal(4, 2)	✓		
6		GELString	nvarchar(255)	✓		
7		Location	nvarchar(512)	✓		

	Key name	Columns	Description
	PK_GEIntegration	CIID	

dbo.GlobalDefaults

Schema	dbo
Name	GlobalDefaults

	PK	Name	Data type	Null	Attributes	Description
1		Type	varchar(64)			
2		value	bigint			

dbo.HelpDeskConfiguration

Schema	dbo
Name	HelpDeskConfiguration

	PK	Name	Data type	Null	Attributes	Description
1	✓	HDConfigID	bigint		Identity / Auto increment column	
2		HDType	int			
3		ConfigType	int			
4		ConfigXML	nvarchar(MAX)			

	Key name	Columns	Description
✓	PK_HelpDeskConfiguration	HDConfigID	

dbo.InactiveAlerts

Schema	dbo
Name	InactiveAlerts

	PK	Name	Data type	Null	Attributes	Description
1		id	bigint	✓		

Name
dbo.PurgeClearedAlerts

dbo.KeyDefinition

Schema	dbo
Name	KeyDefinition

	PK	Name	Data type	Null	Attributes	Description
1	✓	ClassID	int			
2		Key1	varchar(256)	✓		
3		Key2	varchar(256)	✓		

	PK	Name	Data type	Null	Attributes	Description
4		Key3	varchar(256)	✓		
5		Key4	varchar(256)	✓		
6		Key5	varchar(256)	✓		
7		Key6	varchar(256)	✓		
8		Key7	varchar(256)	✓		
9		Key8	varchar(256)	✓		

	Key name	Columns	Description
	PK_KEY	ClassID	

dbo.OutageAlerts

Schema	dbo
Name	OutageAlerts

	PK	Name	Data type	Null	Attributes	Description
1		PeriodID	bigint			
2		AlertID	bigint			

	Key name	Columns	Description
	PK_OutageAlerts	PeriodID, AlertID	

Name
dbo.AllOutageAlerts
dbo.PurgeOutageAlert

dbo.OutageQualityAlerts

Schema	dbo
Name	OutageQualityAlerts

	PK	Name	Data type	Null	Attributes	Description
1		PeriodID	bigint			
2		AlertID	bigint			

	Key name	Columns	Description
	PK_OutageQualityAlerts	PeriodID, AlertID	

Name
dbo.AllQualityAlerts
dbo.PurgeOutageAlert

dbo.OutageRiskAlerts

Schema	dbo
Name	OutageRiskAlerts

	PK	Name	Data type	Null	Attributes	Description
1		PeriodID	bigint			
2		AlertID	bigint			

	Key name	Columns	Description
	PK_OutageRiskAlerts	PeriodID, AlertID	

Name
dbo.AllRiskAlerts
dbo.PurgeOutageAlert

dbo.PolicyGroup

Schema	dbo
Name	PolicyGroup

	PK	Name	Data type	Null	Attributes	Description
1	✓	PolicyGroupID	bigint		Identity / Auto increment column	
2		PolicyGroupName	nvarchar(64)			
3		PolicyTypeID	int			
4		PolicyGroupDescription	nvarchar(512)	✓		
5		Threshold1	int			
6		Action1	int			
7		AlertType1	int		Default: 0	
8		Threshold2	int	✓		
9		Action2	int	✓		
10		AlertType2	int		Default: 0	
11		Threshold3	int	✓		
12		Action3	int	✓		
13		AlertType3	int		Default: 0	
14		Threshold4	int	✓		
15		Action4	int	✓		
16		AlertType4	int		Default: 0	
17		UsmRelTypeID	int			
18		AnodeCIID	bigint			
19		ServiceCIID	bigint			
20		IsAutoMaint	tinyint			

Foreign table		Primary table	Join	Title / Name / Description
dbo.PolicyGroup		dbo.PolicyType	dbo.PolicyGroup.PolicyTypeID = dbo.PolicyType.PolicyTypeID	FK_PolicyGroup_PolicyTypeID

	Key name	Columns	Description
	PK_PolicyGroup	PolicyGroupID	

Name
dbo.PolicyType

dbo.PolicyGroupCI

Schema	dbo
Name	PolicyGroupCI

	PK	Name	Data type	Null	Attributes	Description
1		PolicyGroupID	bigint			
2		BnodeCIID	bigint			

	Key name	Columns	Description
	PK_PolicyGroupCI	PolicyGroupID, BnodeCIID	

dbo.PolicyType

Schema	dbo
Name	PolicyType

	PK	Name	Data type	Null	Attributes	Description
1		PolicyTypeID	int			
2		PolicyTypeName	nvarchar(64)			

	PK	Name	Data type	Null	Attributes	Description
3		PolicyTypedesc	nvarchar(256)	✓		
4		IsDefault	tinyint			
5		IsAutoMaint	tinyint			
6		Threshold1	int			
7		Action1	int			
8		Threshold2	int	✓		
9		Action2	int	✓		
10		Threshold3	int	✓		
11		Action3	int	✓		
12		Threshold4	int	✓		
13		Action4	int	✓		

Foreign table		Primary table	Join	Title / Name / Description
dbo.PolicyGroup		dbo.PolicyType	dbo.PolicyGroup.PolicyTypeID = dbo.PolicyType.PolicyTypeID	FK_PolicyGroup_PolicyTypeID

	Key name	Columns	Description
	PK_PolicyType	PolicyTypeID	

Name	
dbo.PolicyGroup	

dbo.ProductInfo

Schema	dbo
Name	ProductInfo

	PK	Name	Data type	Null	Attributes	Description
1	✓	ProductName	nvarchar(255)			
2		ProductDescription	text	✓		
3		Majorver	int	✓		
4		Minorver	int	✓		
5		Releasever	int	✓		
6		Buildver	int	✓		
7		Licenceinfo	nvarchar(255)	✓		
8		installeddate	datetime	✓		
9		DbSchemaVersion	varchar(16)	✓		

	Key name	Columns	Description
✓	PK_ProductInfo	ProductName	

Name
dbo.PurgeClearedAlerts

dbo.RootCauseRules

Schema	dbo
Name	RootCauseRules

	PK	Name	Data type	Null	Attributes	Description
1	✓	ID	int		Identity / Auto increment column	
2		RuleString	ntext			

	PK	Name	Data type	Null	Attributes	Description
3		Enabled	tinyint			

	Key name	Columns	Description
	PK_RootCauseRules	ID	

dbo.ScheduleRelationship

Schema	dbo
Name	ScheduleRelationship

	PK	Name	Data type	Null	Attributes	Description
1		CIID	bigint			
2		ScheduleID	bigint			

Foreign table		Primary table	Join	Title / Name / Description
dbo.ScheduleRelationship		dbo.CI	dbo.ScheduleRelationship.CIID = dbo.CI.CIID	FK_ScheduleRelationship_CIID
dbo.ScheduleRelationship		dbo.Schedule	dbo.ScheduleRelationship.ScheduleID = dbo.Schedule.ScheduleID	FK_ScheduleRelationship_ScheduleID

	Key name	Columns	Description
	PK_SCHEDULE_RELATIONSHIP	CIID, ScheduleID	

Name
dbo.CI
dbo.Schedule

dbo.service_discovery_rules

Schema	dbo
Name	service_discovery_rules

	PK	Name	Data type	Null	Attributes	Description
1		rules_id	int		Identity / Auto increment column	
2		tenant	varchar(30)			
3		rules	ntext			
4		revision	int			
5		production	char(1)			
6		date_of_change	datetime			
7		user_of_change	varchar(30)			

	Key name	Columns	Description
	PK_service_discovery_rules	rules_id	

dbo.ServiceLevelObjective

Schema	dbo
Name	ServiceLevelObjective

	PK	Name	Data type	Null	Attributes	Description
1		SLOID	bigint		Identity / Auto increment column	
2		Name	nvarchar(256)	✓		
3		Enabled	int			
4		ObjectiveType	int			
5		OutageType	int			
6		SLOPeriod	bigint	✓		
7		Description	nvarchar(256)	✓		
8		ViolationThresh	decimal(11, 5)			
9		ViolationType	int			
10		AlarmThresh	decimal(8, 5)			
11		CreatedBy	nvarchar(50)	✓		

	PK	Name	Data type	Null	Attributes	Description
12		CreateTime	datetime	✓		
13		Active	tinyint		Default: 0	
14		Met	tinyint		Default: 1	
15		FailureTime	bigint		Default: 0	
16		UpdateTime	bigint		Default: 0	

Foreign table		Primary table	Join	Title / Name / Description
dbo.CISLO		dbo.ServiceLevelObjective	dbo.CISLO.SLOID = dbo.ServiceLevelObjective.SLOID	FK_CISLO_SLOID
dbo.SLOSchedule		dbo.ServiceLevelObjective	dbo.SLOSchedule.SLOID = dbo.ServiceLevelObjective.SLOID	FK_SLOSchedule_SLOID

	Key name	Columns	Description
	PK_SLO_ID	SLOID	

Name
dbo.SLAReport
dbo.CISLO
dbo.SLOSchedule

dbo.SLARecord

Schema	dbo
Name	SLARecord

	PK	Name	Data type	Null	Attributes	Description
1		PeriodID	bigint		Identity / Auto increment column	
2		SLAID	bigint			
3		ServiceID	bigint			
4		SLOID	bigint			
5		SLAName	nvarchar(256)			
6		ServiceName	nvarchar(256)			
7		StartPeriod	datetime			
8		EndPeriod	datetime			

	PK	Name	Data type	Null	Attributes	Description
9		Threshold	bigint			
10		Up	bigint			
11		Outage	bigint			
12		Violated	bigint			
13		Unknown	bigint			
14		Maintenance	bigint			
15		Active	tinyint	✓	Default: 0	
16		Met	tinyint	✓	Default: 0	
17		FailureTime	datetime	✓		
18		UpdateTime	datetime	✓		

	Key name	Columns	Description
	PK_SLARecord	PeriodID	

Name
dbo.SLAReport

dbo.SLOSchedule

Schema	dbo
Name	SLOSchedule

	PK	Name	Data type	Null	Attributes	Description
1		SLOID	bigint			
2		ScheduleID	bigint			

Foreign table		Primary table	Join	Title / Name / Description
dbo.SLOSchedule		dbo.Schedule	dbo.SLOSchedule.ScheduleID = dbo.Schedule.ScheduleID	FK_SLOSchedule_ScheduleID
dbo.SLOSchedule		dbo.ServiceLevelObjective	dbo.SLOSchedule.SLOID = dbo.ServiceLevelObjective.SLOID	FK_SLOSchedule_SLOID

	Key name	Columns	Description
	PK_SLOSCHEDULE_RELATIONSHIP	SLOID, ScheduleID	

Name
dbo.Schedule
dbo.ServiceLevelObjective

Name
dbo.SLAReport

dbo.TopoLayout

Schema	dbo
Name	TopoLayout

	PK	Name	Data type	Null	Attributes	Description
1		CIID	bigint			
2		Layout	ntext			
3		UpdateTime	bigint			

dbo.UM_CLAIM

Schema	dbo
Name	UM_CLAIM

	PK	Name	Data type	Null	Attributes	Description
1	✓	UM_ID	int		Identity / Auto increment column	
2		UM_DIALECT_ID	int	✓		
3		UM_CLAIM_URI	varchar(255)	✓		
4		UM_DISPLAY_TAG	varchar(255)	✓		
5		UM_DESCRIPTION	varchar(255)	✓		
6		UM_MAPPED_ATTRIBUTE	varchar(255)	✓		
7		UM_REG_EX	varchar(255)	✓		
8		UM_SUPPORTED	smallint	✓		
9		UM_REQUIRED	smallint	✓		
10		UM_DISPLAY_ORDER	int	✓		
11		UM_TENANT_ID	int		Default: 0	

Foreign table		Primary table	Join	Title / Name / Description
dbo.UM_CLAIM		dbo.UM_DIALECT	dbo.UM_CLAIM.UM_DIALECT_ID = dbo.UM_DIALECT.UM_ID dbo.UM_CLAIM.UM_TENANT_ID = dbo.UM_DIALECT.UM_TENANT_ID	FK__UM_CLAIM__6319B466
dbo.UM_CLAIM_BEHAVIOR		dbo.UM_CLAIM	dbo.UM_CLAIM_BEHAVIOR.UM_UM_CLAIM_ID = dbo.UM_CLAIM.UM_ID dbo.UM_CLAIM_BEHAVIOR.UM_TENANT_ID = dbo.UM_CLAIM.UM_TENANT_ID	FK__UM_CLAIM_BEHAVIO__6F7F8B4B

	Key name	Columns	Description
	PK__UM_CLAIM__FD75BEDE603D47BB	UM_ID, UM_TENANT_ID	

Name
dbo.UM_DIALECT

Name
dbo.UM_CLAIM_BEHAVIOR

dbo.UM_CLAIM_BEHAVIOR

Schema	dbo
Name	UM_CLAIM_BEHAVIOR

	PK	Name	Data type	Null	Attributes	Description
1		UM_ID	int		Identity / Auto increment column	
2		UM_PROFILE_ID	int	✓		
3		UM_CLAIM_ID	int	✓		
4		UM_BEHAVIOUR	smallint	✓		
5		UM_TENANT_ID	int		Default: 0	

Foreign table		Primary table	Join	Title / Name / Description
dbo.UM_CLAIM_BEHAVIOR		dbo.UM_CLAIM	dbo.UM_CLAIM_BEHAVIOR.UM_ID = dbo.UM_CLAIM.UM_ID dbo.UM_CLAIM_BEHAVIOR.UM_TENANT_ID = dbo.UM_CLAIM.UM_TENANT_ID	FK__UM_CLAIM__UM_CLAIM_BEHAVIO__6F7F8B4B
dbo.UM_CLAIM_BEHAVIOR		dbo.UM_PROFILE_CONFIG	dbo.UM_CLAIM_BEHAVIOR.UM_ID = dbo.UM_PROFILE_CONFIG.UM_ID dbo.UM_CLAIM_BEHAVIOR.UM_TENANT_ID = dbo.UM_PROFILE_CONFIG.UM_TENANT_ID	FK__UM_PROFILE__UM_PROFILE_BEHAVIO__6E8B6712

Unique keys

	Key name	Columns	Description
	PK__UM_CLAIM__FD75BEDE6BAEFA67	UM_ID, UM_TENANT_ID	

Name
dbo.UM_CLAIM
dbo.UM_PROFILE_CONFIG

dbo.UM_CUSTOM_USERSTORE

Schema	dbo
Name	UM_CUSTOM_USERSTORE

	PK	Name	Data type	Null	Attributes	Description
1		UM_ID	int		Identity / Auto increment column	
2		UM_USERSTORE_PROPERTY	varchar(255)			
3		UM_USERSTORE_VALUE	varchar(500)			

	PK	Name	Data type	Null	Attributes	Description
4		UM_TENANT_ID	int		Default: 0	

	Key name	Columns	Description
	PK__UM_CUSTO__FD75BEDE047AA831	UM_ID, UM_TENANT_ID	

dbo.UM_DIALECT

Schema	dbo
Name	UM_DIALECT

	PK	Name	Data type	Null	Attributes	Description
1		UM_ID	int		Identity / Auto increment column	
2		UM_DIALECT_URI	varchar(255)	✓		
3		UM_TENANT_ID	int		Default: 0	

Foreign table		Primary table	Join	Title / Name / Description
dbo.UM_CLAIM		dbo.UM_DIALECT	dbo.UM_CLAIM.UM_DIALECT_ID = dbo.UM_DIALECT.UM_ID dbo.UM_CLAIM.UM_TENANT_ID = dbo.UM_DIALECT.UM_TENANT_ID	FK__UM_CLAIM__6319B466
dbo.UM_PROFILE_CONFIG		dbo.UM_DIALECT	dbo.UM_PROFILE_CONFIG.UM_DIALECT_ID = dbo.UM_DIALECT.UM_ID dbo.UM_PROFILE_CONFIG.UM_TENANT_ID = dbo.UM_DIALECT.UM_TENANT_ID	FK__UM_PROFILE_CONFI__68D28DBC

	Key name	Columns	Description
	PK__UM_DIALE__FD75BEDE589C25F3	UM_ID, UM_TENANT_ID	
	UQ__UM_DIALE__B64D10075B78929E	UM_DIALECT_URI, UM_TENANT_ID	

Name
dbo.UM_CLAIM
dbo.UM_PROFILE_CONFIG

dbo.UM_HYBRID_REMEMBER_ME

Schema	dbo
Name	UM_HYBRID_REMEMBER_ME

	PK	Name	Data type	Null	Attributes	Description
1		UM_ID	int		Identity / Auto increment column	
2		UM_USER_NAME	varchar(255)			
3		UM_COOKIE_VALUE	varchar(1024)	✓		
4		UM_CREATED_TIME	timestamp			
5		UM_TENANT_ID	int		Default: 0	

	Key name	Columns	Description
	PK__UM_HYBRI__FD75BEDE7FB5F314	UM_ID, UM_TENANT_ID	

dbo.UM_HYBRID_ROLE

Schema	dbo
Name	UM_HYBRID_ROLE

	PK	Name	Data type	Null	Attributes	Description
1		UM_ID	int		Identity / Auto increment column	
2		UM_ROLE_NAME	varchar(255)	✓		
3		UM_TENANT_ID	int		Default: 0	

Foreign table		Primary table	Join	Title / Name / Description
dbo.UM_HYBRID_USER_ROLE		dbo.UM_HYBRID_ROLE	dbo.UM_HYBRID_USER_ROLE.UM_ROLE_ID = dbo.UM_HYBRID_ROLE.UM_ID dbo.UM_HYBRID_USER_ROLE.UM_TENANT_ID = dbo.UM_HYBRID_ROLE.UM_TENANT_ID	FK__UM_HYBRID_USE__FD75BEDE7CD98669

	Key name	Columns	Description
	PK__UM_HYBRI__FD75BEDE725BF7F6	UM_ID, UM_TENANT_ID	

Name
dbo.UM_HYBRID_USER_ROLE

Schema	dbo
Name	UM_HYBRID_ROLE

	PK	Name	Data type	Null	Attributes	Description
1		UM_ID	int		Identity / Auto increment column	
2		UM_ROLE_NAME	varchar(255)			
3		UM_TENANT_ID	int		Default: 0	

Foreign table		Primary table	Join	Title / Name / Description
dbo.UM_HYBRID_USER_ROLE		dbo.UM_HYBRID_ROLE	dbo.UM_HYBRID_USER_ROLE.UM_ROLE_ID = dbo.UM_HYBRID_ROLE.UM_ID dbo.UM_HYBRID_USER_ROLE.UM_TENANT_ID = dbo.UM_HYBRID_ROLE.UM_TENANT_ID	FK__UM_HYBRID_USE__FD75BEDE7CD98669

	Key name	Columns	Description
	PK__UM_HYBRI__FD75BEDE725BF7F6	UM_ID, UM_TENANT_ID	

Name
dbo.UM_HYBRID_ROLE
dbo.UM_HYBRID_USER_ROLE

dbo.UM_HYBRID_USER_ROLE

Schema	dbo
Name	UM_HYBRID_USER_ROLE

	PK	Name	Data type	Null	Attributes	Description
1		UM_ID	int		Identity / Auto increment column	
2		UM_USER_NAME	varchar(255)	✓		
3		UM_ROLE_ID	int			
4		UM_TENANT_ID	int		Default: 0	

Foreign table		Primary table	Join	Title / Name / Description
dbo.UM_HYBRID_USER_ROLE		dbo.UM_HYBRID_ROLE	dbo.UM_HYBRID_USER_ROLE.UM_ROLE_ID = dbo.UM_HYBRID_ROLE.UM_ID dbo.UM_HYBRID_USER_ROLE.UM_TENANT_ID = dbo.UM_HYBRID_ROLE.UM_TENANT_ID	FK__UM_HYBRI__UM_HYBRID_USER_R__7CD98669

	Key name	Columns	Description
	PK__UM_HYBRI__FD75BEDE7720AD13	UM_ID, UM_TENANT_ID	
	UQ__UM_HYBRI__32A75FE679FD19BE	UM_USER_NAME, UM_ROLE_ID, UM_TENANT_ID	

Name
dbo.UM_HYBRID_ROLE

dbo.UM_PERMISSION

Schema	dbo
Name	UM_PERMISSION

	PK	Name	Data type	Null	Attributes	Description
1		UM_ID	int		Identity / Auto increment column	
2		UM_RESOURCE_ID	varchar(255)			
3		UM_ACTION	varchar(255)			
4		UM_TENANT_ID	int		Default: 0	

Foreign table		Primary table	Join	Title / Name / Description
dbo.UM_ROLE_PERMISSION		dbo.UM_PERMISSION	dbo.UM_ROLE_PERMISSION.UM_PERMISSION_ID = dbo.UM_PERMISSION.UM_ID dbo.UM_ROLE_PERMISSION.UM_TENANT_ID = dbo.UM_PERMISSION.UM_TENANT_ID	FK__UM_ROLE_PERM__UM_ID__43A1090D
dbo.UM_USER_PERMISSION		dbo.UM_PERMISSION	dbo.UM_USER_PERMISSION.UM_PERMISSION_ID = dbo.UM_PERMISSION.UM_ID dbo.UM_USER_PERMISSION.UM_TENANT_ID = dbo.UM_PERMISSION.UM_TENANT_ID	FK__UM_USER_PERM__UM_ID__4C364F0E

	Key name	Columns	Description
	PK__UM_PERMI__FD75BEDE39237A9A	UM_ID, UM_TENANT_ID	

Name
dbo.UM_ROLE_PERMISSION
dbo.UM_USER_PERMISSION

dbo.UM_PROFILE_CONFIG

Schema	dbo
Name	UM_PROFILE_CONFIG

	PK	Name	Data type	Null	Attributes	Description
1	✓	UM_ID	int		Identity / Auto increment column	
2		UM_DIALECT_ID	int	✓		
3		UM_PROFILE_NAME	varchar(255)	✓		
4	✓	UM_TENANT_ID	int		Default: 0	

Foreign table		Primary table	Join	Title / Name / Description
dbo.UM_PROFILE_CONFIG	✓	dbo.UM_DIALECT	dbo.UM_PROFILE_CONFIG.UM_DIALECT_ID = dbo.UM_DIALECT.UM_ID dbo.UM_PROFILE_CONFIG.UM_TENANT_ID = dbo.UM_DIALECT.UM_TENANT_ID	FK__UM_PROFILE_CONFI__68D28DBC
dbo.UM_CLAIM_BEHAVIOR	✓	dbo.UM_PROFILE_CONFIG	dbo.UM_CLAIM_BEHAVIOR.UM_PROFILE_ID = dbo.UM_PROFILE_CONFIG.UM_ID dbo.UM_CLAIM_BEHAVIOR.UM_TENANT_ID = dbo.UM_PROFILE_CONFIG.UM_TENANT_ID	FK__UM_CLAIM_BEHAVI__0E8E6712

	Key name	Columns	Description
✓	PK__UM_PROFI__FD75BEDE65F62111	UM_ID, UM_TENANT_ID	

Name
dbo.UM_DIALECT

Name
dbo.UM_CLAIM_BEHAVIOR

dbo.UM_ROLE

Schema	dbo
Name	UM_ROLE

	PK	Name	Data type	Null	Attributes	Description
1		UM_ID	int		Identity / Auto increment column	
2		UM_ROLE_NAME	varchar(255)			
3		UM_TENANT_ID	int		Default: 0	

Foreign table		Primary table	Join	Title / Name / Description
dbo.UM_USER_ROLE		dbo.UM_ROLE	dbo.UM_USER_ROLE.UM_ROLE_ID = dbo.UM_ROLE.UM_ID dbo.UM_USER_ROLE.UM_TENANT_ID = dbo.UM_ROLE.UM_TENANT_ID	FK__UM_USER_ROLE__54CB950F

	Key name	Columns	Description
	PK__UM_ROLE__FD75BEDE318258D2	UM_ID, UM_TENANT_ID	
	UQ__UM_ROLE__66544A21345EC57D	UM_ROLE_NAME, UM_TENANT_ID	

Name
dbo.UM_USER_ROLE

dbo.UM_ROLE_PERMISSION

Schema	dbo
Name	UM_ROLE_PERMISSION

	PK	Name	Data type	Null	Attributes	Description
1		UM_ID	int		Identity / Auto increment column	
2		UM_PERMISSION_ID	int			
3		UM_ROLE_NAME	varchar(255)			

	PK	Name	Data type	Null	Attributes	Description
4		UM_IS_ALLOWED	smallint			
5		UM_TENANT_ID	int		Default: 0	

Foreign table		Primary table	Join	Title / Name / Description
dbo.UM_ROLE_PERMISSION		dbo.UM_PERMISSION	dbo.UM_ROLE_PERMISSION.UM_ID = dbo.UM_PERMISSION.UM_ID dbo.UM_ROLE_PERMISSION.UM_TENANT_ID = dbo.UM_PERMISSION.UM_TENANT_ID	FK__UM_ROLE_PERMISSI__43A1090D

	Key name	Columns	Description
	PK__UM_ROLE___FD75BEDE3DE82FB7	UM_ID, UM_TENANT_ID	
	UQ__UM_ROLE___223F71C740C49C62	UM_PERMISSION_ID, UM_ROLE_NAME, UM_TENANT_ID	

Name	
dbo.UM_PERMISSION	

dbo.UM_TENANT

Schema	dbo
Name	UM_TENANT

	PK	Name	Data type	Null	Attributes	Description
1		UM_ID	int		Identity / Auto increment column	
2		UM_DOMAIN_NAME	varchar(255)			
3		UM_EMAIL	varchar(255)	✓		
4		UM_ACTIVE	bit	✓	Default: 0	
5		UM_CREATED_DATE	timestamp			

	PK	Name	Data type	Null	Attributes	Description
6		UM_USER_CONFIG	varbinary(MAX)	✓		

	Key name	Columns	Description
	PK__UM_TENAN__53B61AD11B9317B3	UM_ID	
	UQ__UM_TENAN__2BFE64861E6F845E	UM_DOMAIN_NAME	

dbo.UM_USER

Schema	dbo
Name	UM_USER

	PK	Name	Data type	Null	Attributes	Description
1		UM_ID	int		Identity / Auto increment column	
2		UM_USER_NAME	varchar(255)			
3		UM_USER_PASSWORD	varchar(255)			
4		UM_SALT_VALUE	varchar(31)			
5		UM_REQUIRE_CHANGE	bit	✓	Default: 0	
6		UM_CHANGED_TIME	datetime	✓		
7		UM_TENANT_ID	int		Default: 0	

Foreign table		Primary table	Join	Title / Name / Description
dbo.UM_USER_ATTRIBUTE		dbo.UM_USER	dbo.UM_USER_ATTRIBUTE.UM_USER_ID = dbo.UM_USER.UM_ID dbo.UM_USER_ATTRIBUTE.UM_TENANT_ID = dbo.UM_USER.UM_TENANT_ID	FK__UM_USER_ATTRIBUT__2EA5EC27
dbo.UM_USER_ROLE		dbo.UM_USER	dbo.UM_USER_ROLE.UM_USER_ID = dbo.UM_USER.UM_ID dbo.UM_USER_ROLE.UM_TENANT_ID = dbo.UM_USER.UM_TENANT_ID	FK__UM_USER_ROLE__55BFB948

	Key name	Columns	Description
	PK__UM_USER__FD75BEDE2334397B	UM_ID, UM_TENANT_ID	
	UQ__UM_USER__858D3E6B2610A626	UM_USER_NAME, UM_TENANT_ID	

Name
dbo.UM_USER_ATTRIBUTE
dbo.UM_USER_ROLE

dbo.UM_USER_ATTRIBUTE

Schema	dbo
Name	UM_USER_ATTRIBUTE

	PK	Name	Data type	Null	Attributes	Description
1		UM_ID	int		Identity / Auto increment column	
2		UM_ATTR_NAME	varchar(255)			
3		UM_ATTR_VALUE	varchar(1024)	✓		
4		UM_PROFILE_ID	varchar(255)	✓		
5		UM_USER_ID	int	✓		
6		UM_TENANT_ID	int		Default: 0	

Foreign table		Primary table	Join	Title / Name / Description
dbo.UM_USER_ATTRIBUTE		dbo.UM_USER	dbo.UM_USER_ATTRIBUTE.UM_USER_ID = dbo.UM_USER.UM_ID dbo.UM_USER_ATTRIBUTE.UM_TENANT_ID = dbo.UM_USER.UM_TENANT_ID	FK__UM_USER_ATTRIBUT__2EA5EC27

	Key name	Columns	Description
	PK__UM_USER___FD75BEDE2BC97F7C	UM_ID, UM_TENANT_ID	

Name	
dbo.UM_USER	

dbo.UM_USER_PERMISSION

Schema	dbo
Name	UM_USER_PERMISSION

	PK	Name	Data type	Null	Attributes	Description
1		UM_ID	int		Identity / Auto increment column	
2		UM_PERMISSION_ID	int			
3		UM_USER_NAME	varchar(255)			
4		UM_IS_ALLOWED	smallint			
5		UM_TENANT_ID	int		Default: 0	

Foreign table		Primary table	Join	Title / Name / Description
dbo.UM_USER_PERMISSION		dbo.UM_PERMISSION	dbo.UM_USER_PERMISSION.UM_PERMISSION_ID = dbo.UM_PERMISSION.UM_ID dbo.UM_USER_PERMISSION.UM_TENANT_ID = dbo.UM_PERMISSION.UM_TENANT_ID	FK__UM_USER_PERMISSI__4C364F0E

	Key name	Columns	Description
	PK__UM_USER___FD75BEDE467D75B8	UM_ID, UM_TENANT_ID	
	UQ__UM_USER___8C02E6834959E263	UM_PERMISSION_ID, UM_USER_NAME, UM_TENANT_ID	

Name
dbo.UM_PERMISSION

dbo.UM_USER_ROLE

Schema	dbo
Name	UM_USER_ROLE

	PK	Name	Data type	Null	Attributes	Description
1		UM_ID	int		Identity / Auto increment column	
2		UM_ROLE_ID	int			
3		UM_USER_ID	int			
4		UM_TENANT_ID	int		Default: 0	

Foreign table		Primary table	Join	Title / Name / Description
dbo.UM_USER_ROLE		dbo.UM_ROLE	dbo.UM_USER_ROLE.UM_ID = dbo.UM_ROLE.UM_ID dbo.UM_USER_ROLE.UM_TENANT_ID = dbo.UM_ROLE.UM_TENANT_ID	FK__UM_USER_ROLE__54CB950F
dbo.UM_USER_ROLE		dbo.UM_USER	dbo.UM_USER_ROLE.UM_ID = dbo.UM_USER.UM_ID dbo.UM_USER_ROLE.UM_TENANT_ID = dbo.UM_USER.UM_TENANT_ID	FK__UM_USER_ROLE__55BFB948

	Key name	Columns	Description
	PK__UM_USER___FD75BEDE4F12BBB9	UM_ID, UM_TENANT_ID	
	UQ__UM_USER___BBB8EC7E51EF2864	UM_USER_ID, UM_ROLE_ID, UM_TENANT_ID	

Name
dbo.UM_ROLE
dbo.UM_USER

dbo.UserGroupSecurity

Schema	dbo
Name	UserGroupSecurity

	PK	Name	Data type	Null	Attributes	Description
1	✓	UserGroupID	bigint		Identity / Auto increment column	
2	✓	Name	nvarchar(255)			
3		ServiceAdminFlag	tinyint	✓		
4		CustomerAdminFlag	tinyint	✓		
5		AlertQueueAdminFlag	tinyint	✓		
6	✓	CreateTime	datetime			
7		Deleted	tinyint	✓		
8		DeletedTime	datetime	✓		

	Key name	Columns	Description
✓	PK_USERGROUPSECURITY_RELATIONSHIP	Name, UserGroupID, CreateTime	

Used by

Name	

dbo.UserGroupSecurity
• dbo.SecureAlertQueues
  – dbo.SecureUserAlertQueues
• dbo.SecureCustomers
  – dbo.SecureUserCustomers
  • dbo.SecureUserCustomerServices
• dbo.SecureServices
  – dbo.SecureUserServices	

dbo.UserGroupSecurityAssignment

Schema	dbo
Name	UserGroupSecurityAssignment

	PK	Name	Data type	Null	Attributes	Description
1	✓	UserGroupID	bigint			
2	✓	SecurityTypeID	bigint			
3	✓	InternalID	bigint			
4	✓	CreateTime	datetime			
5		Deleted	tinyint	✓		
6		DeletedTime	datetime	✓		

	Key name	Columns	Description
✓	PK_USERGROUPSECURITYASSIGNMENT_RELATIONSHIP	UserGroupID, SecurityTypeID, InternalID, CreateTime	

Name	

dbo.UserGroupSecurityAssignment
• dbo.SecureAlertQueues
  – dbo.SecureUserAlertQueues
• dbo.SecureCustomers
  – dbo.SecureUserCustomers
  • dbo.SecureUserCustomerServices
• dbo.SecureServices
  – dbo.SecureUserServices	

dbo.UserSecurityAssignment

Schema	dbo
Name	UserSecurityAssignment

	PK	Name	Data type	Null	Attributes	Description
1	✓	UserID	nvarchar(255)			
2	✓	UserGroupID	bigint			
3	✓	CreateTime	datetime			
4		Deleted	tinyint	✓		
5		DeletedTime	datetime	✓		

	Key name	Columns	Description
✓	PK_USERSECURITYASSIGNMENT_RELATIONSHIP	UserID, UserGroupID, CreateTime	

Name	
dbo.UserSecurityAssignment • dbo.SecureUserAlertQueues • dbo.SecureUserCustomers – dbo.SecureUserCustomerServices • dbo.SecureUserServices	

dbo.UserSecurityGroups

Schema	dbo
Name	UserSecurityGroups

	PK	Name	Data type	Null	Attributes	Description
1		UserID	varchar(128)			
2		SecurityGroups	varchar(772)			

	Key name	Columns	Description
	PK_UserSecurityStrings	UserID, SecurityGroups	

Name	
dbo.UserSecurityGroups • dbo.SecureCI • dbo.SecureInfrastructureCI • dbo.SecureServiceCI	

dbo.UserSecurityTypes

Schema	dbo
Name	UserSecurityTypes

	PK	Name	Data type	Null	Attributes	Description
1		SecurityTypeID	bigint			
2		Label	nvarchar(128)			

	Key name	Columns	Description
	PK_USERSECURITYTYPES_RELATIONSHIP	Label, SecurityTypeID	

Frequently Asked Questions - CA SOI

Google Maps

Q: Which usage model I must consider for the Google Maps API (default or premium)?

A: The created API key can be used for all the available user accounts. If the number of API calls does not exceed 15000/day (~1000/hr), use Standard usage limit plan, or you must buy a premium key. In the default configuration, the map shows five distinct addresses per Google Maps API call. If there are more locations, it needs to make more calls.

For example, A service with 28 different locations requires 6 Google Maps API calls. As the map refreshes every 5 seconds, the API calls are repeated every 5 seconds. Each separate instance of the loaded page (all users) makes their separate calls and these calls are available in the same API key. You can change the configuration to add more addresses that are rendered per call or change the update interval. This configuration can reduce or increase the usage of the Google Maps API and also affect the performance of the map view.

Q: If I use this integration what information is collected and will the information be public or private?

A: The location and service information is collected, but only the location information is sent.

Q: Which service model or libraries can be used for CA SOI and Google Maps integration.

A: You must use the Google Maps JavaScript API for the integration.

Q: Is the basic key available with CA SOI Installer Package?

A: The basic key is not available with CA SOI installer packager. You must provide an API key before using the Google Map feature.

Q: How to select the API licensing model, if the integration is inside the firewall?

A: Selecting the API licensing model depends on the number of API calls available in an account.

Troubleshooting

This section describes common diagnostic methods that CA SOI administrators can use to diagnose problems and specific remediation steps for potential problems.

Intended Audience

This section is intended for administrators with an advanced understanding of CA SOI and its components. Most operations that are described in this section require administrator-level permissions in the product.

Diagnostic Tools and Methods

This section describes the diagnostic tools and methods available for administrators to diagnose and fix problems in CA SOI.

Debug Consoles

Contents

As an administrator, you use the CA SOI Debug Consoles to test and debug various CA SOI components. Many troubleshooting topics direct you to use these pages to diagnose problems. We recommend that you use these pages only when you are directed by a troubleshooting topic.

Each Debug Console provides procedures for using the page. The troubleshooting topics also provide the specific Debug Console options to select, fields to complete, and so on.

The topics that follow provide an overview of the Debug Console functionality.

Access the Debug Consoles

SA Manager Debug Console

You access the CA SOI Console Debug Console on the SA Manager server using the following URL:

http://manager_host:manager_*port/*sam/debug/

- *manager_host*
  Specifies the server where the SA Manager resides that was specified during the CA SOI installation.
- *manager_port*
  Specifies the Tomcat port that is used for the SA Manager Server HTTP communication and that was specified during the CA SOI installation.
  Default: 7090.

UI Server Debug Console

You access the CA SOI UI Server Debug Console on the UI Server using the following URL:

http://*uiserver_host*:*ui_port/*sam/debug/

- *uiserver_host*
  Specifies the server where the UI Server resides that was specified during the CA SOI installation.
- *ui_port*
  Specifies the Tomcat port that is used for UI Server HTTP communication and that was specified the during CA SOI installation.
  Default: 7070.

SA Manager Debug Console

The following debug pages are available on the SA Manager server:

> **NOTE**
> Click Show Usage on any debug page to view detailed page usage procedures.

- **Troubleshooting**
  - **Triage Tests**
    Use this page to determine the subsystem availability to help you diagnose subsystem failures. This test is generally your first step in troubleshooting many *major* problems. If any Triage Tests fail, you must resolve those failures before any other problems. For more information, see SA Manager Triage Tests.
  - **Debug Controller**
    Use this page to activate the debug features for various modules and log to the soimgr-debug.log file. Troubleshooting topics ask you to turn on the certain logging feature when diagnosing problems. The advanced options let you separate or group the logging into one or more files.
    Note: We recommend that you turn on each logging option only as you need the information. Excessive logging can both slow down your system and make the log file difficult to read.
  - **Server Log Scanner**
    Use this page to search for a string in one or more CA SOI log files that are available on this server. This search is similar to a grep utility. Found search strings are indicated in red in the log file. The search results also include the log file line numbers and character location. The embedded links allow you to inspect the log contents quickly. A summary function provides an outline of possible problems that are found in the log file.
  - **Consolidated Log Scanner**
    Use this page to search the soimgr-error.log file. You can search for connection errors and you can see information from particular sources.
- **Logs**
  - **Log Download**
    Use this page to package and download log files that you want to archive or for working with CA Support technicians.
  - **Web Server Log**
    Use this page to view and search the current segment of the soimgr.log file. This page is useful if you cannot access a server remotely to view a log file.
  - **Consolidated Error Log**
    Use this page to display the soimgr-error.log file. You can optionally show the log starting from a particular search string.
- **Debug Pages**
  - **Memory Usage**
    Use this page to view the server's memory for diagnosing performance issues. You can run the Java Garbage Collection mechanism to manage Java memory. You can set the page to refresh automatically at regular intervals. You can also toggle logging memory information to the soimgr.log file.
  - **Thread Info**
    Use this page to generate a Java Virtual Machine (JVM) thread dump when you are trying to diagnose a problem on the JVM. You can see which threads are most active.
  - **Queue Monitor**
    Use this page to monitor the sizes of thread pools and their associated job queues.
  - **Database Connectivity**
    Use this page to validate the SA Store Database connection and test the access time. The test parameters are configurable. You can also generate a report.
  - **Database Tables**

Use this page to inspect and report on the SA Store database table sizes for troubleshooting database problems and determining when the database requires maintenance. You can also generate, package, and download a report.

- **Service Check**
  Use this page to scan and validate the SA Manager Server services so you can diagnose a service corruption. You can also view service statistics such as the number of CIs.
- **Layout Check**
  Use this page to scan and validate the internal service layout (topology) XML descriptors so you can diagnose layout issues. It is nearly impossible to see these layouts in the database itself.

SA Manager Triage Tests

The SA Manager Triage Tests, help you to diagnose major system problems.

Once you access the SA Manager Debug Console and click Triage Tests, click the Show usage link to view how to run the tests.

The SA Manager Triage Tests page provides the following test results:

- **Base Test**
  The Base Test section is an overall sanity test of the platform. The tests verify the mechanisms to run the Triage Tests. Any failure can indicate a major problem. A failure can indicate possible problems with the embedded CA Catalyst module. Restart the SA Manager server. If the Base Test fails again, contact CA Support.
- **Manager Base**
  The Manager Base section determines if the basic SA Manager components are working. Manager Base failures can indicate an SA Store Database problem.
- **Access Manager**
  The Access Manager section indicates possible problems that are related to CA EEM. Failures indicate a problem related to the CA EEM availability or content.
- **Manager Services**
  The Manager Services section indicates possible problems that are related to the alert repository, the escalation policies, and the help desk connection.
- **Auxiliary**
  Ignore this section. This is for future auxiliary tests.

UI Server Debug Console

> **NOTE**
> Click Show Usage on any debug page to view detailed page usage procedures.

- **Troubleshooting**
  - **Triage Tests**
    Use this page to determine the subsystem availability to help you diagnose subsystem failures. This test is generally your first step in troubleshooting many *major* problems. For more information, see UI Server Triage Tests.
  - **Debug Controller**
    Use this page to activate the debug features for various modules and log to the soimgr-debug.log file. Troubleshooting topics ask you to turn on the certain logging feature when diagnosing problems. The advanced options let you separate or group the logging into one or more files.
    Note: We recommend that you turn on each logging option only as you need the information. Excessive logging can both slow down your system and make the log file difficult to read.
  - **Server Log Scanner**
    Use this page to search for a string in one or more CA SOI log files that are available on this server. This search is similar to a grep utility. Found search strings are indicated in red in the log file. The search results also include the

log file line numbers and character location. The embedded links allow you to inspect the log contents quickly. A summary function provides an outline of possible problems that are found in the log file.

- **Logs**
  - **Log Download**
    Use this page to package and download log files that you want to archive or for working with CA Support technicians.
  - **Web Server Log**
    Use this page to view and search the current segment of the soiuis.log file. This page is useful if you cannot access a server remotely to view the log file.
- **Debug Pages**
  - **Memory Usage**
    Use this page to view the server's memory for diagnosing performance issues. You can run the Java Garbage Collection mechanism to manage Java memory. You can set the page to refresh automatically at regular intervals. You can also toggle logging memory information to the soiuis.log file.
  - **Thread Info**
    Use this page to generate a Java Virtual Machine (JVM) thread dump when you are trying to diagnose a problem on the virtual machine. You can see which threads are most active.
  - **Queue Monitor**
    Use this page to monitor the thread pools and their associated queue sizes.
  - **Database Connectivity**
    Use this page to validate the SA Store Database connection and test the access time. The test parameters are configurable. You can also generate a report.
  - **Synchronization Tests**
    Use this page to test the repository synchronization between the UI Server and the SA Manager.
  - **Start Client Debug Console**
    Use this page to set debugging levels for various CA SOI features such as model security, service modeler tree, and users. You view the debugging output in the Java Console, which you enable separately from the Windows Java Control Panel.

UI Server Triage Tests

The UI Server Triage Tests, help you to diagnose major system problems.

Once you access the SA Manager Debug Console and click Triage Tests, click the Show usage link to view how to run the tests.

The UI Server Triage Tests page provides the following test results:

- **Base Test**
  The Base Test section is an overall sanity test of the platform. The tests verify the mechanisms to run the Triage Tests. Any failure indicate a possible major problem. Restart the UI Server. If the Base Test fails again, contact CA Support.
- **UI Server Base**
  The UI Server Base section determines if the basic UI Server components are working. A failure indicates possible problems with the type repository or a problem on the SA Manager.
- **Access Manager**
  The Access Manager section indicates possible problems that are related to CA EEM. Failures are either related to the CA EEM availability or content.
- **Domain Services**
  The Domain Services section tests the ability of the UI Server to connect to the SA Manager. A failure indicates a possible communication problem between the UI Server and the SA Manager.
- **Auxiliary**
  You can ignore this section. This is for future auxiliary tests.

Manage the Debug Level for Specific Modules

As an administrator, you can use the Debug Controller page to activate, adjust, and manage the debug features of individual CA SOI server modules. You can perform the following tasks on this page:

- Enable (ON) or disable (OFF) the debug state of the module.
- Specify the appropriate granularity (OFF, MIN, MOD, and MAX) for the debug level.
- Specify whether to repeat the debug output of any selected module into a separate log file specific to that module. The separate (*Sep*) option on the page provides this functionality. You can also specify whether to repeat the debug output of all selected modules into the same file. The group (*Grp*) option provides this functionality. You can specify these settings by using the advanced options. Also, consider the following points when using the separate and group options:
  - Separate (Sep) and group (Grp) options are not mutually exclusive.
  - You can activate any number of separate and group options simultaneously.
  - The *Repeater* log output is emitted only if debug for the module is turned on.
  - If the debug level of the module is turned off, the repeater log output is also turned off.

    NOTE
    The debug Console is designed to be used only with help from CA Technical Support.

Follow these steps:

1. Click Start, Programs, CA, Service Operations Insight, CA Service Operations Insight User Interface.
2. Enter the appropriate user credentials.
   The Administration UI opens.
3. Click the Administration tab.
   The administration options appear in the left pane.
4. Click the plus sign (+) next to the CA Service Operations Insight Manager Configuration option.
   The available servers appear.
5. Click the name of the required server.
   The page that opens contains the Debugging And Logs button.
6. Click Debugging And Logs and then click the Debug Controller link.
   The Debug Controller page opens.
7. Set the debug level for the specific module as follows:

   NOTE
   You can use the Info icon tooltip (Info column) to view detailed information about the function that a specific module performs.

   If the entry in the Module Handle column is truncated, the entry tooltip shows the complete debug handle string (used in the web.xml setup) for the module.

   - Select ON or OFF in the Desired State column to set the debug state for the module.
   - Select OFF, MIN, MOD, or MAX in the Desired Level column to set the desired debug level granularity.
   - Click the Apply button to save the settings.
     The Current State and Current Debug Level columns start showing the updated state.
8. Click the Advanced Options button if you want to specify the repeater options: separate (Sep) and group (Grp).
   The Debug Controller page is updated with the Repeater Logs column appearing next to the Module Handle column.
9. Select the appropriate repeater option for the module as follows:
   - **Sep**
     Allows the logging subsystem to repeat the debug output of *any* selected module into a separate log file that is dedicated to that module. The tooltip of the separate option (Sep) shows the name of the associated log file. This

name is derived from the abbreviated name of the module. For example, if the Sep option is checked for the *Action Service* module, the debug output of the module is duplicated and logged into the *dbg-rpt-actionservice.log* file.

– **Grp**
Allows the logging subsystem to repeat the debug output of *all* selected modules into the same file. The name of the log file that is associated with the group option (Grp) is fixed as *dbg-rpt-customlog.log*. For example, if the Grp option is checked for the *Action Service* and *Action Retry Config* modules, their debug output is duplicated and combined into the *dbg-rpt-customlog.log* file.

10. Click Apply.
The settings are saved.

Log Files

Using Log Files for Diagnosis

Contents

As an administrator, you can use log files to diagnose problems. This section includes information about various log files that help you troubleshoot CA SOI. You can search the contents of many log files on the SA Manager Debug Console and the UI Server Debug Console.

- **BiConfig.log**
Helps you troubleshoot reporting import errors. This file is available at SOI_HOME\Reports.
- **catalyst.log**
Contains information specific to CA Catalyst transactions. This file is available at SOI_HOME\tomcat\logs.
- **CA-SSA-LogicInstallDebug.log**
Tracks the Logic Server installation. Use this file to troubleshoot Logic Server installation problems. This CA Catalyst log file is available at SOI_HOME\log.
- **CA-SSA-RegistryInstallDebug.log**
Tracks the Registry installation. Use this file to troubleshoot Registry installation problems. This CA Catalyst log file is available at SOI_HOME\log.
- **CA-SSA_RestUIInstallDebug.log**
Tracks the USM Web View installation. This file is available at SOI_HOME\log.
- **ci-invalid.log**
Tracks the details about CIs with duplicate sheets. If you do not see a CI in CA SOI, check this file to see if it is a duplicate sheet and the limit for duplicates has been reached for that reconciled sheet. This file is available at SOI_HOME\tomcat\logs.
- **client.log**
Contains an entry for every UI Server that is connected to the SA Manager and records for each user connection. This file is available at SOI_HOME\tomcat\webapps\sam\console\logs. The Client Log page lets you view the contents of the client.log file. You can also use this page to clear the log and remove old entries.
- **connmgr.log**
Traces data that the SA Manager receives from the connectors. The Connector Manager option on the Administration UI debug page controls this log file. This file is available at SOI_HOME\tomcat\logs.

  NOTE
  For more information about the improved SA Manager logs reorganization, see the Reorganization of the SA Manager Logs section.
- Domain_Install_*releasenumber*.log
Tracks the CA SOI Domain connector installation. This file is created when you click Done after the CA SOI Domain connector installation finishes. This file is available at SOI_HOME\log.
- **eitransform.log**

Contains information about all policy operations (such as parse, normalize, and format) included in your connector policy. You can find the eitransform.log file at SOI_HOME\log after it is enabled.

- **EventMgmt.log**
  Includes Event Management information. By default, the log level in the file is set to INFO. This file is available at SOI_HOME\log. Use this file to troubleshoot the Event Management-related errors.
  Example: Search on the Universal connector unexpectedly returning no events
  This example considers an example search on the Universal connector that unexpectedly returned no events. After configuring a DEBUG log level and running the search again, the following search information is logged in the file:

```
INFO | jvm 1 | 2011/04/05 10:02:51 | 10:02:51,063 INFO SDOFactory:208 -
com.ca.eventmanager.operations.EventManagerImpl.getEvents: Incoming Map:key=(Connectors)
 Value=(CA:09997_server01.ca.com@server01.ca.com)
INFO | jvm 1 | 2011/04/05 10:02:51 | 10:02:51,063 INFO SDOFactory:208 -
 com.ca.eventmanager.operations.EventManagerImpl.getEvents: Incoming Map:key=(MdrProduct) Value=(CA:09997)
INFO | jvm 1 | 2011/04/05 10:02:51 | 10:02:51,063 INFO SDOFactory:208 -
 com.ca.eventmanager.operations.EventManagerImpl.getEvents: Incoming Map:key=(scope.query.operator)
 Value=(OR)
INFO | jvm 1 | 2011/04/05 10:02:51 | 10:02:51,063 INFO SDOFactory:208 -
 com.ca.eventmanager.operations.EventManagerImpl.getEvents: Incoming Map:key=(scope.query.timelast)
 Value=(1)
...
INFO | jvm 1 | 2011/04/05 10:02:51 | 10:02:51,141 DEBUG XQueryHelper:95 -
 com.ca.eventmanager.common.XQueryHelper.setTimeLast: timelast=1 qp.timestart=1302008571
 qp.timeend=1302012171
INFO | jvm 1 | 2011/04/05 10:02:51 | 10:02:51,141 DEBUG XQueryHelper:96 -
 com.ca.eventmanager.common.XQueryHelper.setTimeLast: qp.timestart.epoch=Apr 5, 2011 9:02:51 AM
 qp.timeend.epoch=Apr 5, 2011 10:02:51 AM
INFO | jvm 1 | 2011/04/05 10:02:51 | total reccount=0
INFO | jvm 1 | 2011/04/05 10:02:51 | timescoped files: 0
INFO | jvm 1 | 2011/04/05 10:02:51 | recordscoped files:
INFO | jvm 1 | 2011/04/05 10:02:51 | final-query: let $a := doc('file:/C:/Program%20Files/CA/SOI/
resources/Core/EventStore/temp/results27890.xml0.xml')/results/normal return if (count($a)>0) then (<group
 id='0'>{$a}</group>) else ()
INFO | jvm 1 | 2011/04/05 10:02:51 | resultsfile: C:\Program Files\CA\SOI\resources\Core\EventStore\temp
\results27890.xml
...
INFO | jvm 1 | 2011/04/05 10:02:52 | 10:02:52,219 INFO SDOFactory:208 -
 com.ca.eventmanager.operations.EventManagerImpl.getEvents: Returning from DataSource:SSA key=(Result)
 Value=(<results scopedeventcount='0' returnedeventcount='0' warning='' error='pattern_not_matched'> set
 log4j level to TRACE to log entire result set.)
```

- **EventManagement_wrapper.log**
  Contains status details of the CA SAM Event Management service. This file is available at SOI_HOME\jsw\logs.
- **generic_client.log**
  Tracks web services transactions that the Universal connector client makes. This file is available at SOI_HOME\log. The SOI_HOME\lib\generic\log4j.xml file controls this log. The default level is ERROR. Set to DEBUG to trace transactions.
- **ifw.log**
  Contains connector processing details. This file is available at SOI_HOME\log. By default, the debug level is set to ERROR. The IFW appender in SOI_HOME\resources\log4j.xml controls the log levels. If you set a log level, you do not need to restart the system. In general, the log levels produce this type of information:

  - INFO: Represents the messages pertaining to the IFW and connector health
  - DEBUG: Represents the data being published and received; also creates log\debugData files
  - TRACE: Represents the specific methods being called (code trace)
  - ERROR: Represents the error messages that are returned from connectors or IFW
  - FATAL: Represents the non-recoverable errors

  When the IFW appender is set to DEBUG, the SOI_HOME\log\debugData folder contains the following types of files:
  - *HEARTBEAT_PUB.txt: Includes status details of IFW or connectors
  - *RAW.txt: Includes CI, Alert, and Relationship details per connector prior to normalization through EI
  - *PUB.txt: Includes CI, Alert, Relationship details per connector after EI normalization
- **indexer-catalyst.log**
  Includes detailed logging for the CA SAM Store Indexer service. This file is available at SOI_HOME\jsw\logs. The SOI_HOME\indexer\log4j.properties file controls this log.
- **install.log**
  Includes Event connector-related pre- or post-installation errors such as database creation failure and connector registration problems. This log file is available at EI_HOME\logs.
- IntegrationServices_Install_*releasenumber*.log
  Includes IFW installation errors. Use this file to troubleshoot the IFW installation issues. This file is available at SOI_HOME\log.
- MidTier_Install_*releasenumber*
  Tracks installation errors that are related to the Mid-tier connector installation. This file is available at SOI_HOME\log.
- **registry.log**
  Tracks the Registry activity. Use this file to troubleshoot the Registry problems. This CA Catalyst log file is available at SOI_HOME\tomcat\logs.
- Sample_Install_*releasenumber*.log
  Tracks installation errors that are related to the Sample connector. This file is created when you click Done after the Sample connector installation finishes. The file is available at SOI_HOME\log.
- **SAM-IntegrationServices_wrapper.log**
  Contains status and general information regarding the Integration Services process and the connectors. This file is available at SOI_HOME\jsw\logs. This log file acts as a primary runtime log file and includes basic runtime informational messages. Anything that is written to *stdout* is captured in this file. You can check the file to determine the following information:
  - Why the Integration Services service is not running.
  - Why a connector is not running or is in a particular state.

  Example: AMQ connection failure
  This example shows the MQ server connection failure information that is logged in the file:
  ```
  INFO   | jvm 1    | 2009/03/20 09:24:57 | AMQPublisher@server1.ca.com: Trying to establish connection with
   AMQ.
  INFO   | jvm 1    | 2009/03/20 09:24:58 | AMQPublisher@server1.ca.com: AMQ Connection failed. User name or
   password is invalid.
  ```
- **SAM_Tomcat_wrapper.log**
  Includes status information and general processing details from the CA SAM Application Server service. This file is available at SOI_HOME\jsw\logs.
- **SAM-UI_Server_wrapper.log**
  Includes status information and general processing details from the CA SAM User Interface Server service. This file is available at SOI_HOME\jsw\logs.
- **SAM-StoreIndexer_wrapper.log**
  Includes status information and general processing details from the CA SAM Store Indexer service. This file is available at SOI_HOME\\jsw\logs.
- **soimgr.log (and soiuis.log)**
  Tracks CA SOI and CA Catalyst activity. This file also contains all lifecycle (or heartbeat) messages information.

The file soimgr.log at SOI_HOME\tomcat\logs contains SA Manager-specific information. The file soiuis.log that is available at SOI_HOME\SamUI\logs contains UI Server-specific information.

> **NOTE**
> For more information about the SA Manager logs reorganization, see the Reorganization of the SA Manager Logs section.

- **soimgr-debug.log (and soiuis-debug.log)**
  Tracks all debug messages. The soimgr-debug.log file that is available at SOI_HOME\tomcat\logs contains SA Manager-specific information. The file soiuis-debug.log at SOI_HOME\SamUI\logs contains UI Server-specific information.

  > **NOTE**
  > For more information about the SA Manager logs reorganization, see the Reorganization of the SA Manager Logs section.
- **soimgr-error.log**
  Includes a consolidated error log that contains errors from several product components. This file is available at SOI_HOME\tomcat\logs. When you encounter a problem with the product, reference this log file first. The file is accessible from the Operations Console Connection Status dialog and the Connector Configuration page of the Administration UI.
- SOI_Install_*releasenumber*.log
  Includes CA SOI installation error information. This file is available at SOI_HOME\log.
- **ssa.log**
  Includes information about whether USM schema validation failures have occurred. This file is available at SOI_HOME\log.
- **ssa-mobile.log**
  Tracks the Mobile Dashboard information. This file is available at SOI_HOME\SAMUI\logs.
- **ssaweb.log**
  Includes information that is associated with CA Catalyst USM Web View. This file is available at SOI_HOME\SamUI\logs.
- **service-discovery.log**
  Tracks the Service Discovery processing. This file is available at SOI_HOME\log. The *SD* appender in the SOI_HOME\resources\log4j.xml file controls this log.
- **ServiceDiscovery_InstallLog.log**
  Tracks the installation of Service Discovery. This file is available at SOI_HOME\log.
- **trace.log**
  Includes the Reconciler activity. This CA Catalyst log file is available at SOI_HOME\tomcat\logs.
- **ucf.log**
  Tracks the UCF Broker activity. Use this file to troubleshoot synchronization problems that are not occurring in the Logic Server. This CA Catalyst log file is available at SOI_HOME\log.
- **UCF-Broker_wrapper.log**
  Includes information about general UCF Broker status and processing. This file is available at SOI_HOME\jsw\logs.

Updated Log File Names

In releases before CA SOI r3.1, the SA Manager Server and UI Server log files had the same names: sam.log, samdebug.log, and samerror.log. In CA SOI r3.1, the log files were renamed to differentiate these logs and were also updated to include "SOI" as the naming prefix.

The log files have the following names:

SA Manager Server Log Files

Old Log File Name	New Log File Name (r3.1 and later)
sam.log	soimgr.log
samdebug.log	soimgr-debug.log
samerror.log	soimgr-error.log

UI Server Log Files

Old Log File Name	New Log File Name (r3.1 and later)
sam.log	soiuis.log
samdebug.log	soiuis-debug.log

Reorganization of the SA Manager Logs

In CA SOI r3.0 SP3, CA SOI improved the organization of the SA Manager logs. The SA Manager logs are now organized into multiple log files instead of the single sam.log file, which was the case in the releases prior to CA SOI r3.0 SP3. Log messages are logically grouped into specific log files depending on the type of log information, which helps you use these files more efficiently. For example, if you are searching for a specific debug message, you can directly go to the soimgr-debug.log file (samdebug.log in CA SOI r3.0 SP3). This log file includes all debug messages. Segregating the log information in this manner helps you quickly identify the related log file and locate the problem message, reducing the time that is taken to fix the issue.

The SA Manager logs are reorganized into three separate log files as follows:

> **NOTE**
> You can find these files at SOI_HOME\tomcat\logs.

- All *lifecycle* (or *heartbeat*) messages are logged in the soimgr.log (sam.log in CA SOI r3.0 SP3) file.
- All debug messages are logged in the soimgr-debug.log (samdebug.log in CA SOI r3.0 SP3) file.
- All messages from the Connector Manager component are logged in the connmgr.log file.

> **NOTE**
> For more information about how to set the debug level for a specific module, see Manage the Debug Level for Specific Modules.

Logging in the CA SOI IFW

The connector framework uses the log4j.xml file for logging the related information. The log4j.xml configuration file is available at SOI_HOME\resources. The logging is enabled by default in this log file at the INFO level. Multiple log4j appenders are defined for ActiveMQ, IFW, and so on.

To enable more verbose logging, set a specific logger to DEBUG or TRACE. Additionally, if you want to increase all logging for the framework, set *Root IFW Logger* to DEBUG or TRACE instead of INFO:

```
<!-- ROOT IFW LOGGER -->
<logger name="com.ca.sam.ifw" additivity="false">
    <level value="INFO" />
<appender-ref ref="IFW" />
</logger>
```

Connector-specific Logging

This section provides information about connector-specific logging:

- Connectors can provide their own log4j appenders.
- Log4j configuration of connectors is stored in separate files; for example, *<DomainManager>*_log4j.xml.
- Log4j configuration files of connectors are located under <SOI_HOME>\resources\Configurations\log4j.
- Configuration files contain a connector-specific log4j appender and logger.

Configure Event Management Logging

The Event Management information is logged to the following file:

<SOI_HOME>\log\EventMgmt.log

By default, the log level in the file is INFO. You can change this level to pinpoint errors if your event searches or policies are not performing as expected.

Follow these steps:

1. Open the following file:
   <SOI_HOME>\resources\eventManager-log4j.xml
2. Set the priority value to DEBUG as follows, and save and close the file:
   ```
   <root>
   <priority value="DEBUG">
   </priority>
   <appender-ref ref="stdout" />
   </root>
   ```
   The EventMgmt.log file now produces more detailed debug messages.
3. Restart the CA SAM Event Management service.
   The logging change is applied.

Consider an example search on the Universal connector that unexpectedly returned no events. After configuring a DEBUG log level and running the search again, you can see the search details:

```
INFO | jvm 1 | 2011/04/05 10:02:51 | 10:02:51,063 INFO SDOFactory:208 -
 com.ca.eventmanager.operations.EventManagerImpl.getEvents: Incoming Map:key=(Connectors)
 Value=(CA:09997_server01.ca.com@server01.ca.com)
INFO | jvm 1 | 2011/04/05 10:02:51 | 10:02:51,063 INFO SDOFactory:208 -
 com.ca.eventmanager.operations.EventManagerImpl.getEvents: Incoming Map:key=(MdrProduct) Value=(CA:09997)
INFO | jvm 1 | 2011/04/05 10:02:51 | 10:02:51,063 INFO SDOFactory:208 -
 com.ca.eventmanager.operations.EventManagerImpl.getEvents: Incoming Map:key=(scope.query.operator) Value=(OR)
INFO | jvm 1 | 2011/04/05 10:02:51 | 10:02:51,063 INFO SDOFactory:208 -
 com.ca.eventmanager.operations.EventManagerImpl.getEvents: Incoming Map:key=(scope.query.timelast) Value=(1)
...
INFO | jvm 1 | 2011/04/05 10:02:51 | 10:02:51,141 DEBUG XQueryHelper:95 -
 com.ca.eventmanager.common.XQueryHelper.setTimeLast: timelast=1 qp.timestart=1302008571 qp.timeend=1302012171
INFO | jvm 1 | 2011/04/05 10:02:51 | 10:02:51,141 DEBUG XQueryHelper:96 -
 com.ca.eventmanager.common.XQueryHelper.setTimeLast: qp.timestart.epoch=Apr 5, 2011 9:02:51 AM
 qp.timeend.epoch=Apr 5, 2011 10:02:51 AM
INFO | jvm 1 | 2011/04/05 10:02:51 | total reccount=0
INFO | jvm 1 | 2011/04/05 10:02:51 | timescoped files: 0
INFO | jvm 1 | 2011/04/05 10:02:51 | recordscoped files:
```

```
INFO | jvm 1 | 2011/04/05 10:02:51 | final-query: let $a := doc('file:/C:/Program%20Files/CA/SOI/resources/
Core/EventStore/temp/results27890.xml0.xml')/results/normal return if (count($a)>0) then (<group id='0'>{$a}</
group>) else ()
INFO | jvm 1 | 2011/04/05 10:02:51 | resultsfile: C:\Program Files\CA\SOI\resources\Core\EventStore\temp
\results27890.xml
...
INFO | jvm 1 | 2011/04/05 10:02:52 | 10:02:52,219 INFO SDOFactory:208 -
 com.ca.eventmanager.operations.EventManagerImpl.getEvents: Returning from DataSource:SSA key=(Result)
 Value=(<results scopedeventcount='0' returnedeventcount='0' warning='' error='pattern_not_matched'> set log4j
 level to TRACE to log entire result set.)
```

The initial lines define the Universal connector MdrProduct value (CA:00097), the server name (server01.ca.com), and a defined scope to search on events for the last hour. The DEBUG message writes the start and end time of the search (qp.timestart.epoch=Apr 5, 2011 9:02:51 AM qp.timeend.epoch=Apr 5, 2011 10:02:51 AM), which is helpful for determining the files that were searched. Finally, the following message shows that no events were found in the scoped time:

```
INFO | jvm 1 | 2011/04/05 10:02:51 | timescoped files: 0
```

Using this information, you can change the scoped time period and rerun the search.

View and Manage the Client Log (client.log)

The CA SOI installation program installs a client.log file on the SA Manager. By default, it is located at <SOI_HOME>\tomcat\webapps\sam\console\logs.

The client.log file contains an entry for every user who logs on to the Operations Console. The Client Log page allows you to view the contents of client.log file. You can also use this page to clear the log and remove old entries.

Follow these steps:

1. Launch the Dashboard.
2. Click the Administration tab.
3. Click the plus sign (+) next to the CA Service Operations Insight UI Server Configuration option.
4. Click the plus sign (+) next to the server you want to configure.
   The configuration options appear.
5. Click Client Log.
6. (Optional) Enter a number in the Purge entries older than # days field, and then click Go.
   The log entries older than the specified number of days are removed from the log.
7. (Optional) Click Clear Log.

View and Manage UI Server Connection Log

The CA SOI provides a client.log file on the SA Manager. By default, it is at <SOI_HOME>\tomcat\webapps\sam\console\logs.

The client.log file contains an entry for every UI Server that is connected to the SA Manager and records for each user connection. The Client Log page allows you to view the contents of the client.log file. You can also use this page to clear the log and remove old entries.

Follow these steps:

1. Click the Administration tab.
   The administration options appear in the left pane.
2. Click the plus sign (+) next to CA Service Operations Insight Manager Configuration.

3. Click the plus sign (+) next to the server you want to configure.
4. Click UI Server Connection Log.
5. (Optional) Enter a number in the 'Purge entries older than # days' field, and click Go.
   The log entries older than the specified number of days disappear from the log.
6. (Optional) Click Clear Log.

   NOTE
   You can perform this procedure on the SA Manager only. The UI Server has a similar feature where you can view and manage the users that are connected to the Operations Console.

Control the Rolling Behavior of the Client Log File

Follow these steps:

1. Depending on which client.log file you are modifying, stop either the CA SAM User Interface Server service or the CA SAM Application Server service.
2. Locate and open the web.xml file, which is typically at the following location:
   <SOI_HOME>\SAMUI\webapps\sam\WEB-INF
3. Add one or more of the following parameters to the web.xml file:
   - **MaxLogs**
     Specifies the number of log file extents that are maintained in a sequence.
     param-name: com.aprisma.spectrum.app.web.util.MaxLogs
     Default param-value: 10
   - **MaxSize**
     Specifies the maximum log file extent size.
     param-name: com.aprisma.spectrum.app.web.util.MaxSize
     Default param-value: 2,000,000 (bytes)

     NOTE

     The UI Server uses an in-memory DOM representation of the current log that takes up to ten times the file size. Therefore, keep the size of each log extent low. The average size of the entry is about 400 bytes. Therefore, the default of 2 Mb is sufficient for approximately 5000 log on and off entries.
   - **FlushFreq**
     Specifies the number of log entries that are written between log file size checks.
     param-name: com.aprisma.spectrum.app.web.util.FlushFreq
     Default param-value: 10

   The following sample shows the parameter tags for setting the MaxLogs parameter to a maximum of five extents.

```
<context-param>
  <param-name>com.aprisma.spectrum.app.web.util.MaxLogs</param-name>
  <param-value>5</param-value>
  <description>
      This parameter determines max number of client.log extents.
  </description>
</context-param>
```

4. Save the file and restart the CA SAM User Interface Server service or the CA SAM Application Server service.

Isolate CA Catalyst Logging Information from soimgr.log

You can isolate CA Catalyst-related logging information from soimgr.log in the <SOI_HOME>\tomcat\lib\log4j.xml file. You add appropriate entries to the log4j.xml file. These entries configure the product to log CA Catalyst-specific information to a separate catalyst.log file. This file would include information from the Persistence Service, Reconciler, Synchronizer, and Notification Manager components.

Follow these steps:

1. Locate and open the <SOI_HOME>\tomcat\lib\log4j.xml file in a text editor.
2. Add a section to the file similar to the following:

```
<appender name="CAT" class="org.apache.log4j.RollingFileAppender">
 <param name="File" value="&logDir;/catalyst.log"/>
 <param name="Append" value="true"/>
 <param name="MaxFileSize" value="20MB"/>
 <param name="MaxBackupIndex" value="10"/>
 <layout class="org.apache.log4j.PatternLayout">
 <param name="ConversionPattern" value="&filePattern;"/>
 </layout>
</appender>
```

3. Add a logger to the appender with the appropriate log level as follows:

```
<logger name="com.ca.ssa.sor" additivity="false">
 <level value="INFO" />
 <appender-ref ref="CAT" />
</logger>
```

4. Save the changes and close the file.

Using Services for Diagnosis

Contents

As an administrator, you use the following component services that control various operations in CA SOI:

- **CA SOI Application Server**
  Controls the operation of the SA Manager. This service is installed on any system that contains the SA Manager component. Some of the examples where you are required to restart this service are as follows:
  - You configure the email server to send CA SOI notifications if a *mailhost* DNS alias for the email server does not exist.
  - You configure the Mobile Dashboard connection information, USM Web View connection settings, and global settings.
  - You configure the eventManagerClientConfig.xml file to decide how you want to control the Event Management data flow to the Operations Console.
  - You reconfigure the Persistence Store connectivity.
  - You change the administrator password.
  - You experience installation errors or have problems initializing CA Catalyst components.
- **CA SOI Event Management**
  Controls communication with the Event Store on connector systems for Event Management. This service is installed on any system that contains the SA Manager or a connector. Depending on the status of the service (running or stopped), data sources on the system appear in green (running) or red (stopped). Some of the examples where you are required to restart this service are as follows:
  - You configure the Event Management logging.
  - You configure the event search settings.
- **CA SOI Integration Services**
  Controls the operation of the IFW, which handles the communication between the connectors and the SA Manager. This service is installed on any system that contains a connector or the MQ Server, which is a component of the SA Manager. Some of the examples where you are required to restart this service are as follows:

  - You configure the IFW configuration file.
  - You enable a connector that was previously disabled.
  - You configure the Event Store parameters.
  - You manually deploy a custom connector.
  - You change the global alert filter setting.
  - You change the global DNS lookup setting.
  - You experience installation errors or have problems initializing CA Catalyst components.
- **CA SOI Store Indexer**
  Controls the indexing of USM data from the Persistence Store for use by the USM Web View. This service is installed on any system that contains the UI Server. Some of the examples where you are required to restart this service are as follows:
  - You change the Persistence Store connectivity while configuring the Registry.
  - You change the Solr connectivity as part of the Registry configuration.
  - You configure the Registry for the standalone SA Manager installation.
  - You experience installation errors or have problems initializing CA Catalyst components.
- **CA SOI User Interface Server**
  Controls the operation of the UI Server, including all user interfaces. This service is installed on any system that contains the UI Server component. Some of the examples where you are required to restart this service are as follows:
  - You add custom links to the Dashboard.
  - You add custom metrics to the Dashboard.
  - You configure the level of services that the Dashboard displays.
  - You change the metric icons on the Mobile Dashboard.
  - You customize the Mobile Dashboard polling intervals.
  - You synchronize the UI Server with the report server.
  - You experience installation errors or have problems initializing CA Catalyst components.
- **CA UCF Broker**
  Controls the UCF broker, which facilitates create, update, and delete operations from connectors to their source domain managers. This service is installed on any system that contains the SA Manager. Some of the examples where you restart this service are as follows:
  - You try to enable the maintenance synchronization in the case no domain manager entry appears in the list.
  - You experience installation errors or have problems initializing CA Catalyst components.

Manage Services

You can view the status of each installed service and can start or stop it as required. For example, you can verify that the product installed successfully by checking that all installed services are in the running state. If any service is not running, you can manually restart it.

Follow these steps:

1. Select Start, Settings, Control Panel, Administrative Tools, Services.
   The Services dialog opens.
2. Check for the status of the following services on the appropriate servers, depending on where the components were installed:
   - CA SOI Application Server
   - CA SOI Event Management
   - CA SOI Integration Services
   - CA SOI Store Indexer
   - CA SOI User Interface Server
   - CA UCF Broker

3. Right-click the service name, and select Start (or Stop) from the context menu.
   The service is started or stopped as applicable.

Using the Status Bar for Diagnosis

Contents

As administrator, you can use the Status bar at the bottom of the Operations Console to view the status information about the current CA SOI connection. The Status bar contains the following icons:

-
  Opens the Connection Status dialog, which displays the current connection status of all CA SOI components (SA Manager, UI Server, connectors, and so on).

  NOTE
  For the User Management Service, even if CA EEM displays without a version number in the Description column, it does not indicate a problem with the product. The service still displays as Online.
-
  Opens the Messages dialog, which displays any new messages from the administrator.

If the SA Manager connection is lost, the Status bar also displays your user name, the login server, and an alert message.

Verify the Connection Status of Components

You can verify that the connection status of each installed component using the Connection Status dialog. The Connection Status dialog also displays a connection log and an Open Consolidated Error Log button. This log button opens the consolidated log file in a web browser. The Connection icon displays a green icon to indicate that all components have a connected status. The icon displays a red icon to indicate that one or more components have lost connection.

Follow these steps:

1. Select Start, Programs, CA, Service Operations Insight, CA Service Operations Insight User Interface.
   An authentication dialog opens.
2. Enter the Administrator user credentials that you specified during the installation, and click OK.
   The CA SOI Dashboard opens.
3. Click Console.
   The Operations Console opens.
4. Click the Connection icon

   at the bottom right of the Console.
   The Connection Status dialog opens and displays the connection status of each installed component.
5. Review the information.

   NOTE
   If you want to view the message, click the Messages icon

   .
   The Messages dialog also displays the message date, time, and sender information.

Using the Administration Tab for Diagnosis

As an administrator, use the Administration tab on the Dashboard to maintain connectors and SA Manager settings. The Administration tab also lets you configure single sign-on using CA EEM, email notifications, and other administrative functions. You can also use the Administration tab with other diagnostic methods to diagnose various issues. Some of the issues that you can diagnose using the Administration tab are as follows:

- You are unable to connect to CA Service Desk or create CA Service Desk in CA SOI. In this case, you can verify that the connection settings are correct on the Help Desk Configuration page of the Administration tab. (Administration, CA Service Operations Insight Manager Configuration, *server_name.*)
- You are not getting any data from your connector. In such situations, you can consider the following points:
  - Verify the status of the connector in the Connector Status field (Administration, Connector Configuration, *server_name*, *connector_name*).
  - Review the associated message in the Status Description field (Administration, Connector Configuration, *server_name*, *connector_name*).
  - Open the consolidated error log file from the Connector Configuration page (Administration, Connector Configuration).
  - Verify the status of the IFW in the Current Integration Framework Status field (Administration, Connector Configuration, *server_name*).
- You are experiencing issues communicating with the CA EEM server. To diagnose this issue, you can verify that the correct CA EEM server-related information is available on the EEM Configuration page on the SA Manager and UI Server. You can also click the Test button to verify whether the connection information is correct.
- You are unable to launch the Mobile Dashboard from the Administration UI. To diagnose this issue, you can review the communication settings for the Mobile Dashboard server on the Mobile Dashboard Server Configuration page (Administration, CA Service Operations Insight Manager Configuration, *server_name*).
- You can view details about the SA Manager to help troubleshoot messaging and database connection issues. Access the Administration, CA Service Operations Insight Manager Configuration, *server_name* page to view the information. You can access the SA Server Debug Console on the SA Manager server using the Debugging and Logs button on the Manager Configure page (Administration, CA Service Operations Insight Manager Configuration, *server_name*).
- You are unable to use the $[USM Web View URL] runtime token to invoke USM Web View as part of an escalation action. In this case, you can review and test the connection settings for USM Web View on the USM Web View Configuration page (Administration, CA Service Operations Insight Manager Configuration, *server_name*).
- You can access the UI Server Debug Console on the UI server using the Debugging and Logs button on the UI Server page (Administration, CA Service Operations Insight UI Server Configuration, *server_name*).
- You are unable to invoke CA Process Automation processes in an alert escalation action. In this scenario, you can verify the communication settings for the CA Process Automation server on the Process Automation Server Configuration page (Administration, CA Service Operations Insight Manager Configuration, *server_name*).
- You can review the Client Log page (Administration, CA Service Operations UI Server Configuration, *server_name*) to view the contents of the client.log file. This file contains an entry for every UI Server that is connected to the SA Manager and records for each user connection. You can also use this page to clear the log and remove old entries.
- You are unable to generate reports in CA SOI. In this case, you can verify that CA SOI is configured to access the BusinessObjects report server. Also, verify that all the connection settings on the Configure Report Server page (Administration, CA Service Operations UI Server Configuration, *server_name*) are correct.

> **NOTE**
> For more information about how to access the Administration UI and perform various operations, see General Administration.

Set Notifications for the OutOfMemory Conditions on the SA Manager

As an administrator, you can enable email notifications for out-of-memory conditions.

Any exceptions (such as OutOfMemory conditions) on the SA Manager can compromise the stability of the SA Manager if they are left unnoticed. To manage such situations, CA SOI provides a detection mechanism that sends notifications whenever any exception stops the JSW wrapper. CA SOI identifies such scenarios and uses the JSW wrapper to send email notifications to the administrator. The email includes detailed information about the failure condition and the reason about why CA SOI was shut down. The administrator can then analyze the scenario, take appropriate measures to fix the issues, and prevent the SA Manager from growing indefinitely and compromising the stability. To set the email notification, update the SOI_HOME\jsw\conf\EmailNotification.conf file.

Follow these steps:

1. Navigate to the SOI_HOME\jsw\conf folder.
2. Locate and open the EmailNotification.conf file in a text editor.
3. Update the parameters available under various sections of the file as appropriate; for example:
   - To enter the sender and recipient email addresses, update the *wrapper.event.default.email.sender* and *wrapper.event.default.email.recipient* parameters under the *# Common Event Email settings.* section. An example is as follows:
     ```
     wrapper.event.default.email.sender=abc@xyz.com
     wrapper.event.default.email.recipient=def@xyz.com
     ```
   - To update the email server, update the *wrapper.event.default.email.smtp.host=<SMTP Server Host>* entry and modify the SMTP server host with the actual email server the client is using. An example is as follows:
     ```
     wrapper.event.default.email.smtp.host=mail.xyz.com
     ```
   - To receive notifications for any option that is mentioned in the *# Enable specific event emails.* section, uncomment the specific option.
   - To customize the body of the email, update the *wrapper.event.jvm_restart.email.body* parameter under the *# Specify custom mail content* section. An example is as follows:
     ```
     wrapper.event.jvm_restart.email.body=The JVM was restarted.\n\nPlease check on its status.\n
     ```

     NOTE
     For more information about various parameters, see http://wrapper.tanukisoftware.com/doc/english/props-event.html#properties.
4. Save the changes in the file.
   The settings are applied.

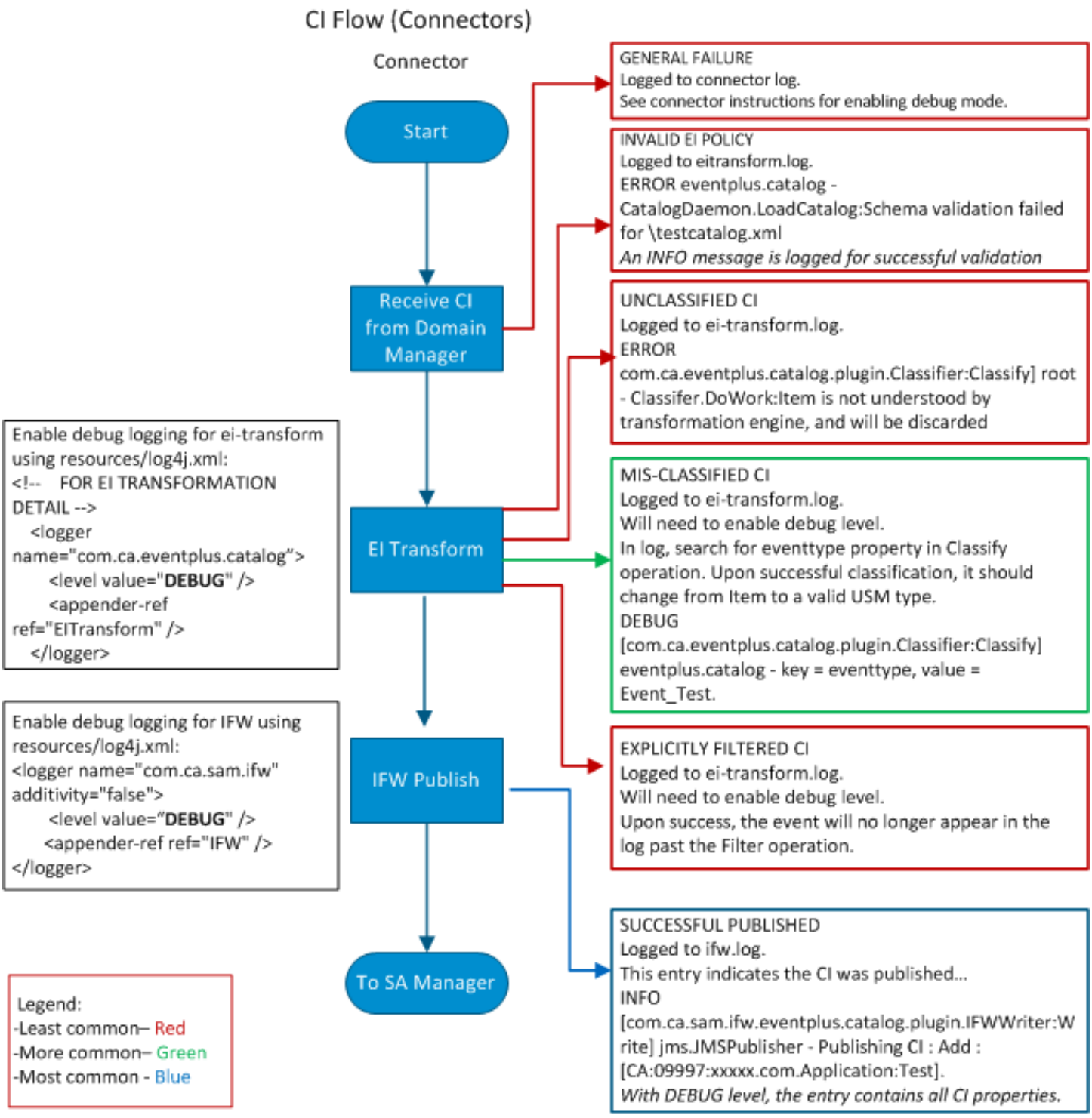
CI Flow in CA SOI and Log File Outputs

Contents

As an administrator, use the following graphics to view the flow of CIs through various components. The graphics in each topic also show the outputs that are generated to the log files during the CI flow so that you can trace CIs through the system when necessary.

CI Flow in CA SOI Connectors

The following graphic shows the CI flow and related log files in connectors:

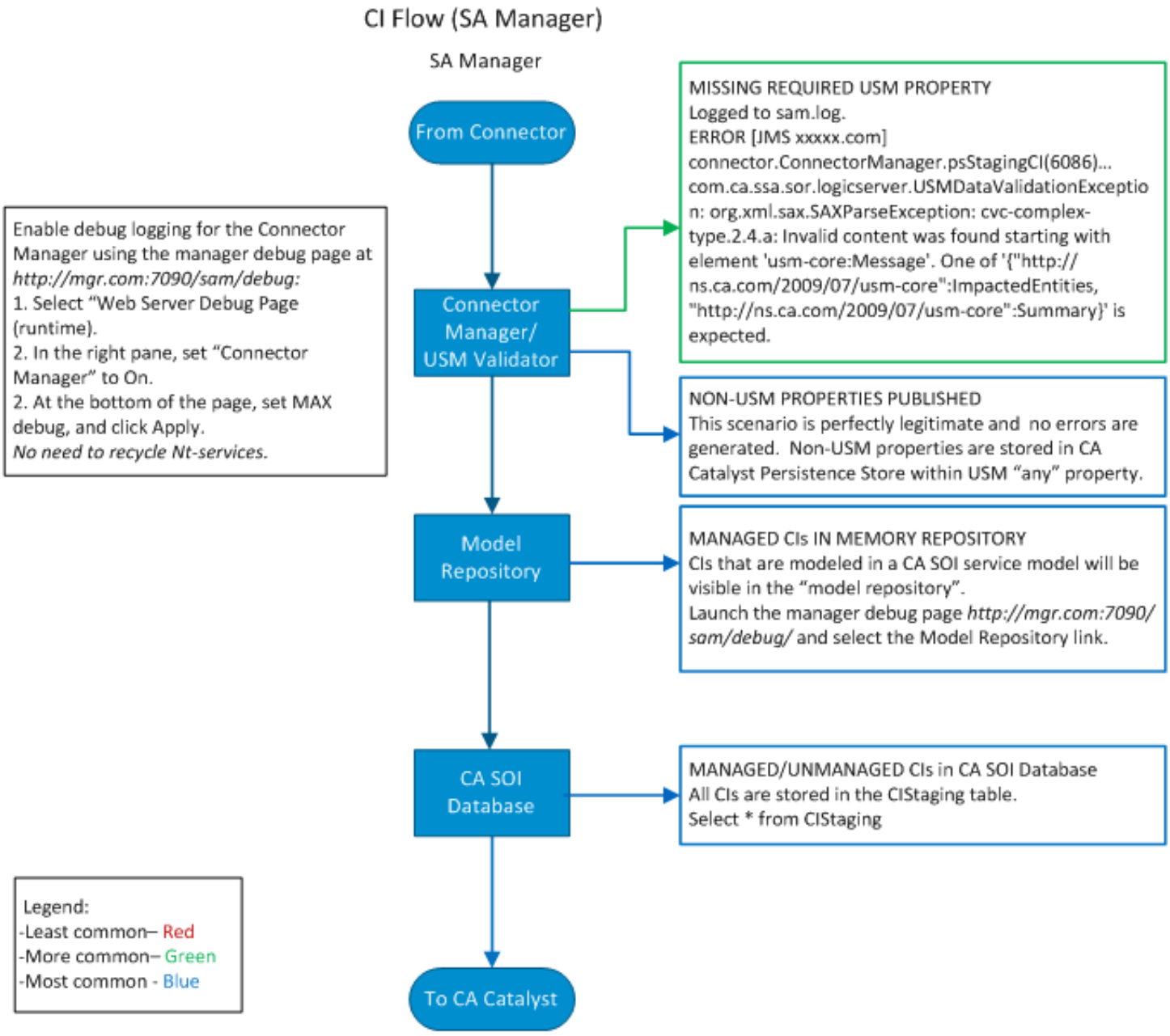


The following points explain the information that is covered in the graphic:

- Connector receives CIs from the domain manager.
- CIs are transformed to the USM format.
- Transformed CIs move to the IFW for publishing.
- The IFW publishes transformed CIs to the SA Manager.

CI Flow in CA SOI SA Manager

The following graphic shows the CI flow and related log files in the SA Manager:

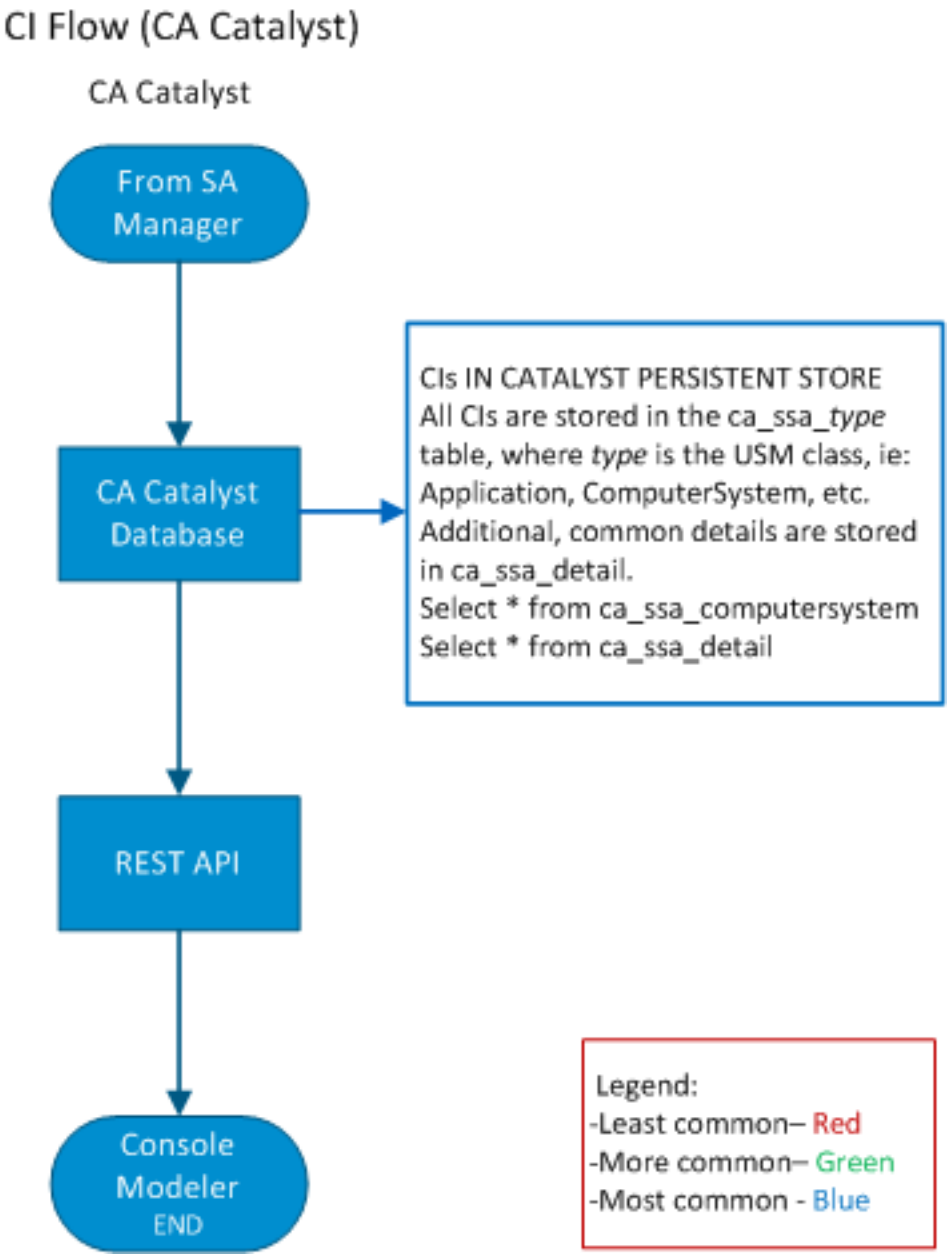


The following points explain the information that is covered in the graphic:

- Connector Manager (or USM Validator) validates the CIs (for example, validates for any missing USM properties) coming from the connector.
- Model Repository maintains and caches the model information. All managed CIs become available in this repository.
- All CIs (managed and unmanaged) are stored in the CA SOI database (SA Store) and become available to CA Catalyst.

CI Flow in CA SOI CA Catalyst

The following graphic shows the CI flow and related log files in CA Catalyst:



The following points explain the information that is covered in the graphic:

- All CIs are stored in the CA Catalyst database.
- All stored CIs become available to USM Web View.
- All CIs become available to the Console and Modeler.

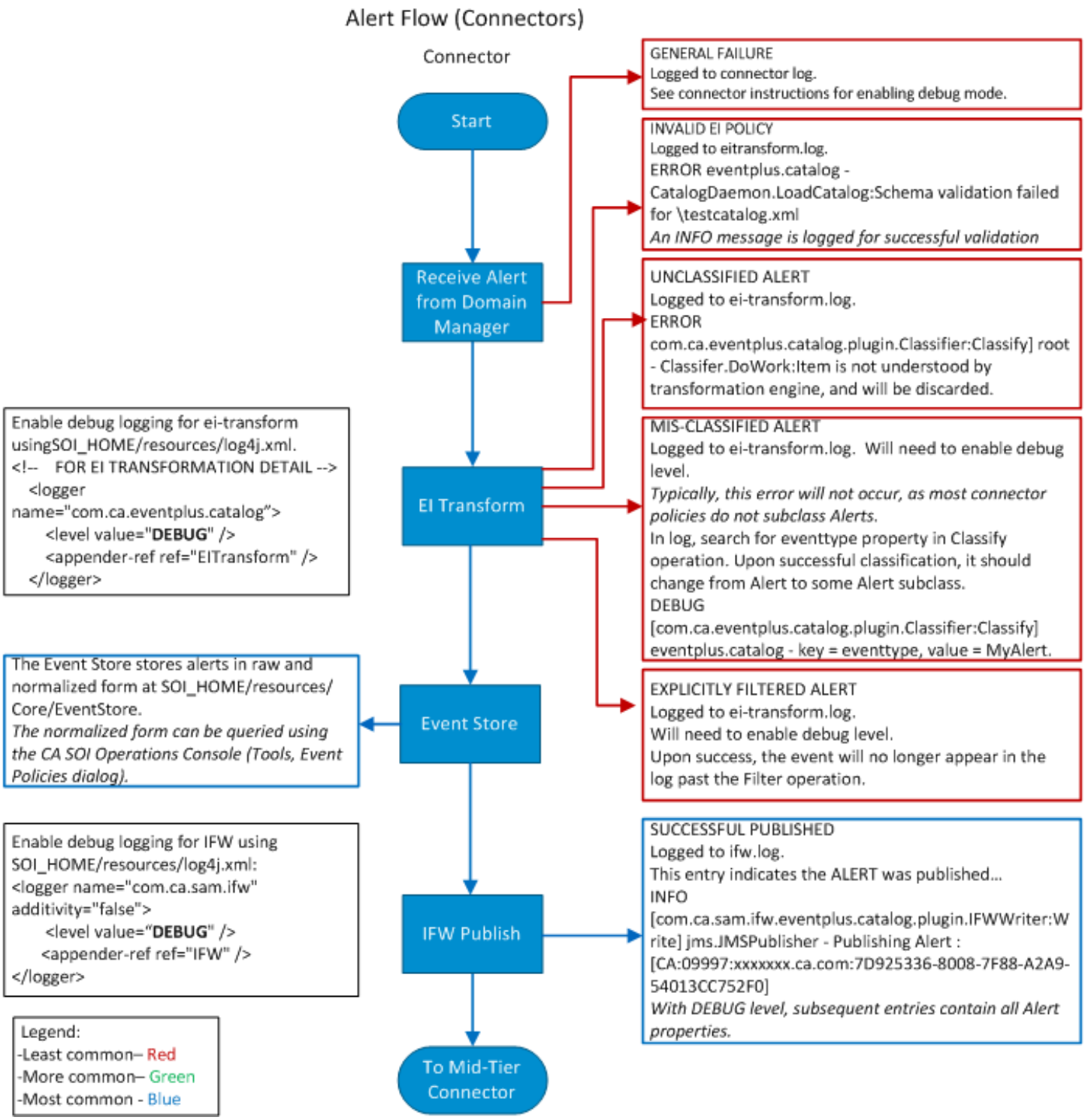
Alert Flow in CA SOI and Log File Outputs

Contents

As an administrator, view the following graphics to understand the flow of alerts through various components. The graphics also show the outputs that are generated to the log files during the alert flow so that you can trace alerts through the system when necessary.

Alert Flow in CA SOI Connectors

The following graphic shows the alert flow and related log files in connectors:



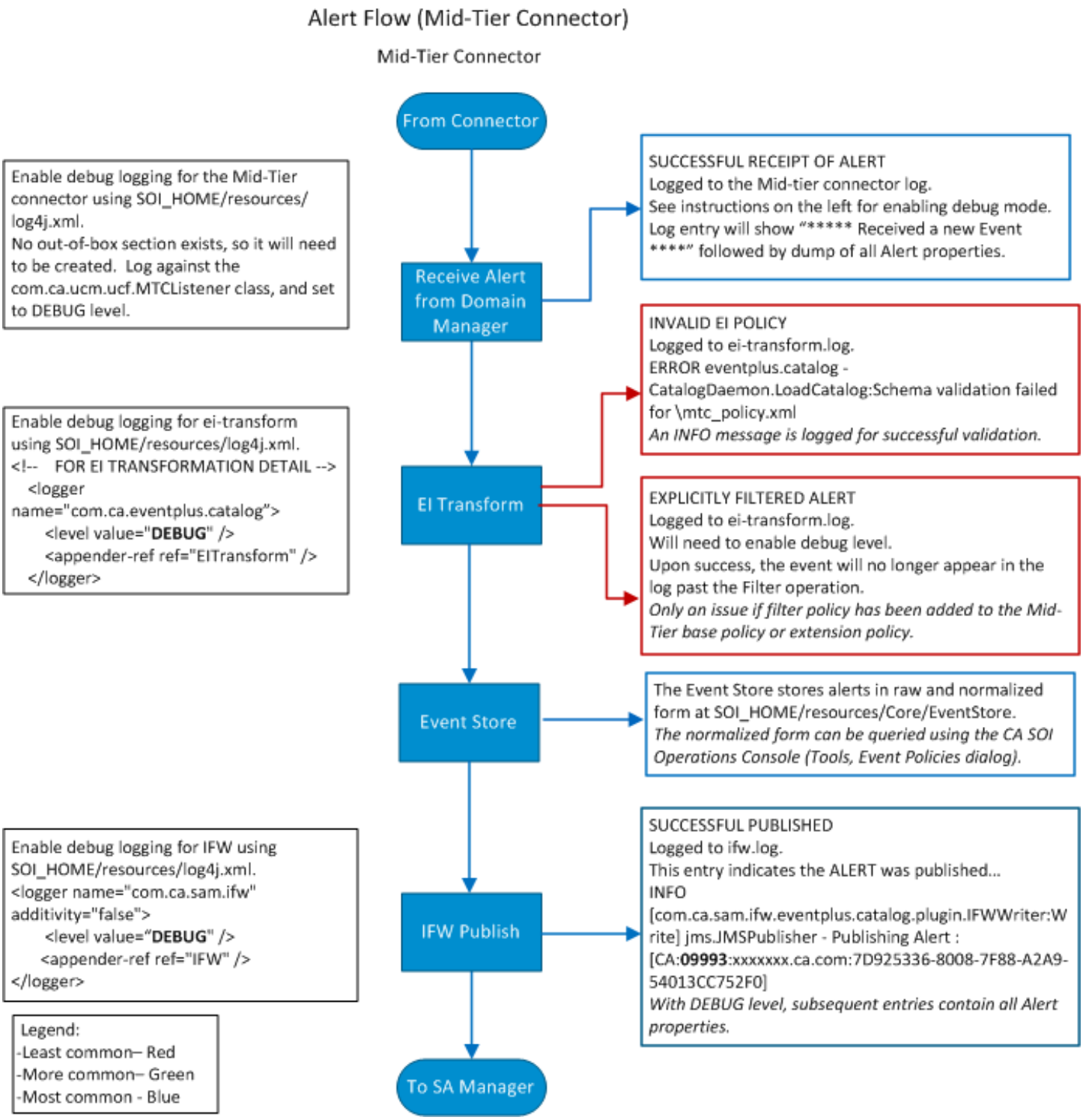
The following points explain the information that is covered in the graphic:

- Connector receives alerts from the domain manager.
- Alerts are transformed to the USM format.
- The Event Store stores all alerts.
- Alerts are passed to the IFW for publishing.
- The IFW publishes transformed alerts to the Mid-Tier connector (if available).

NOTE
If the Mid-Tier connector is not present, alerts directly move to the SA Manager.

Alert Flow in CA SOI Mid-Tier Connector

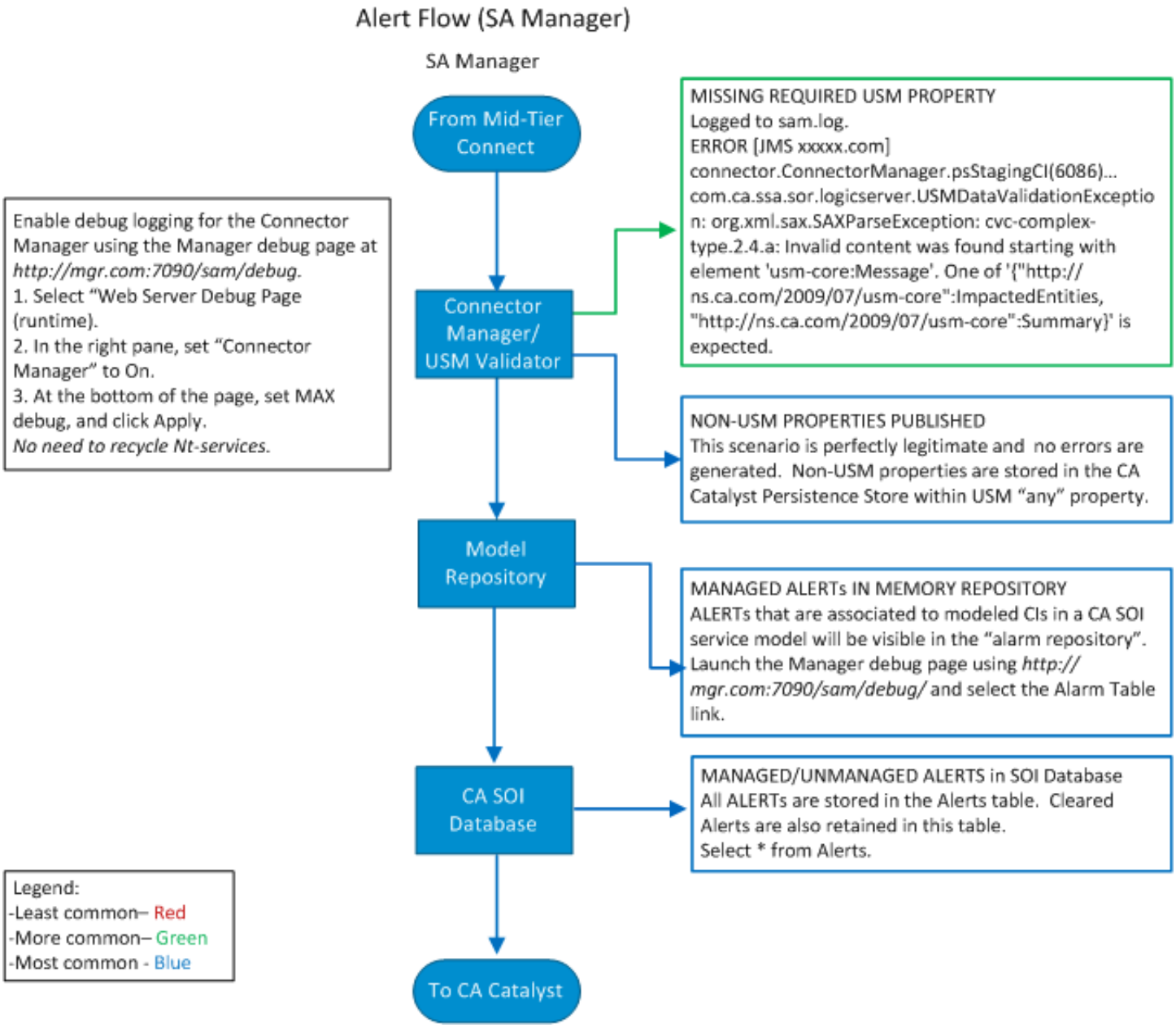
The following graphic shows the alert flow and related log files in the Mid-Tier connector:



The same process is followed as explained in the Alert Flow in CA SOI: Connectors section. However, in this case, the IFW publishes alerts to the SA Manager.

Alert Flow in CA SOI SA Manager

The following graphic shows the alert flow and related log files in the SA Manager:

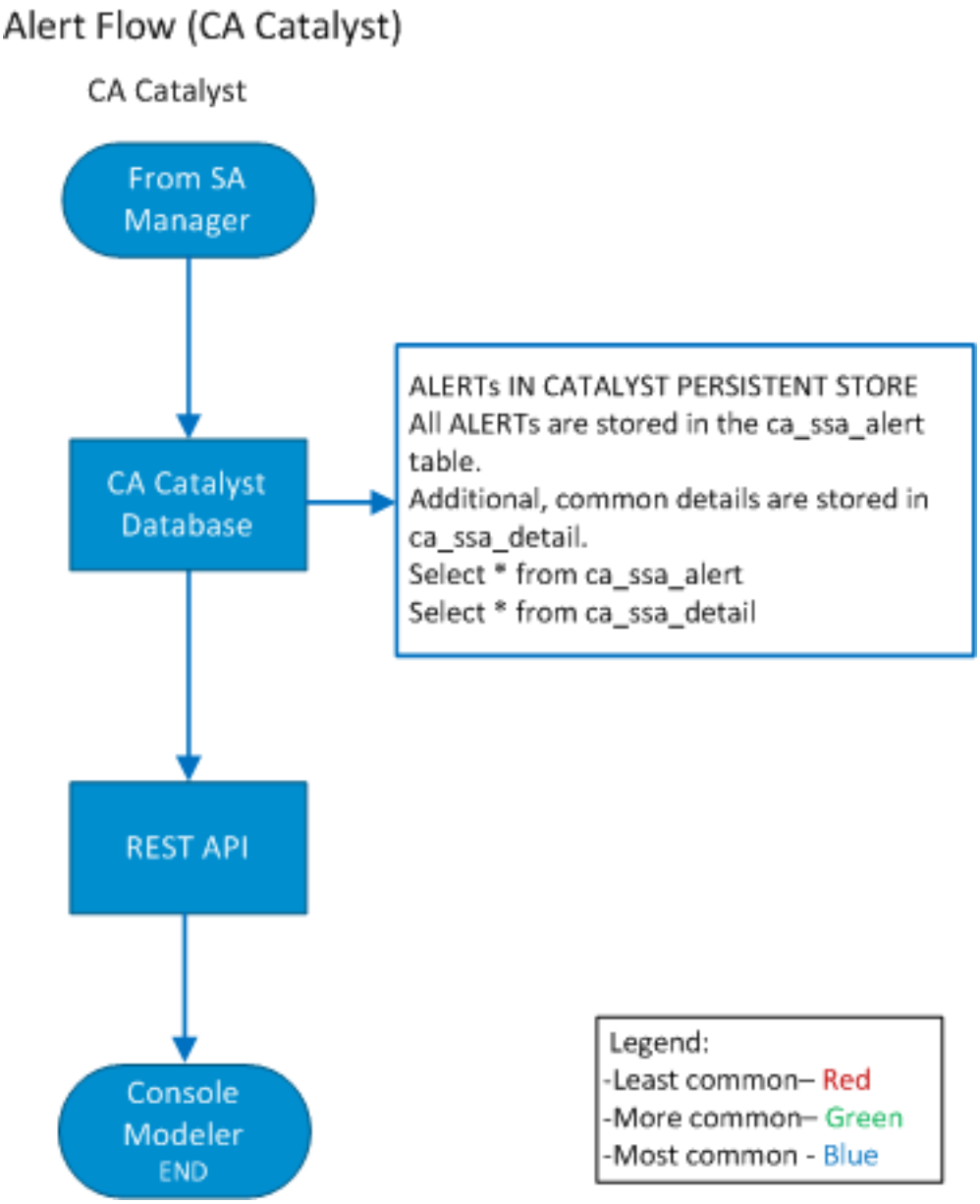


The following points explain the information that is covered in the graphic:

- Connector Manager (or USM Validator) validates the alerts (for example, validates for any missing USM properties) coming from the connector (or Mid-Tier connector).
- Model Repository maintains and caches the model information. All managed alerts become available in this repository.
- All alerts (managed and unmanaged) are stored in the CA SOI database (SA Store) and become available to CA Catalyst.

Alert Flow in CA SOI CA Catalyst

The following graphic shows the alert flow and related log files in CA Catalyst:



The following points explain the information that is covered in the graphic:

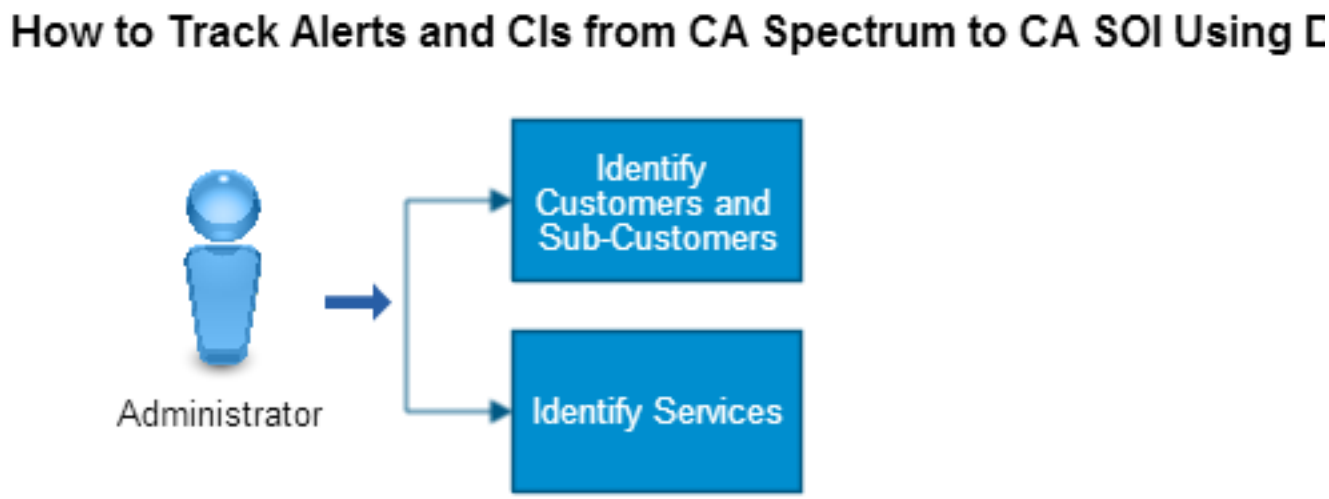
- All alerts coming from the SA Manager are stored in the CA Catalyst database.
- All stored alerts information becomes available to USM Web View.
- All alerts become available to the Console and Modeler.

How to Track Alerts and CIs from CA Spectrum to CA SOI Using Debug Logs

As an administrator, you can track the data flow from CA Spectrum to CA SOI to investigate and resolve related issues. This scenario describes how you can enable debug on the CA Spectrum connector. You can then investigate the debug logs to track alerts and CIs as they flow from CA Spectrum to CA SOI.

Use this scenario to guide you through the process:

Figure 55: how to track alerts from Spectrum



- How to Track Alerts from CA Spectrum to CA SOI Using Debug Logs.
- How to Track CIs from CA Spectrum to CA SOI Using Debug Logs.

For this example, only one particular device in CA Spectrum has been used against which alerts are created. The alerts are then tracked in the debug files to verify that the alerts are passed from CA Spectrum to the CA Spectrum connector and then to CA Catalyst.

> **NOTE**
> For more information about the command-line interface and Event Configuration interface in CA Spectrum, see the CA Spectrum documentation. For more information about the CA Spectrum connector, see the *CA Spectrum Connector Guide*.

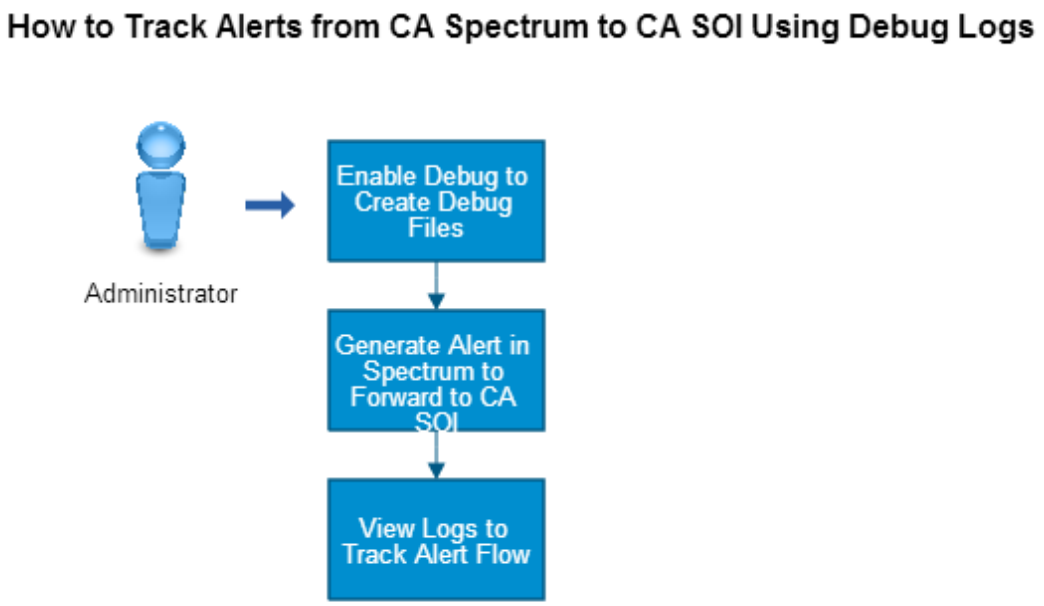
How to Track Alerts from CA Spectrum to CA SOI Using Debug Logs

Contents

As a CA SOI administration, you can enable debug on the CA Spectrum connector and can investigate the debug logs to track alerts as they flow from CA Spectrum to CA SOI.

Use this scenario to guide you through the process:

Figure 56: how to track alerts from Spectrum using debug logs



1. Enable Debug to Create Debug Files.
2. Generate an Alert in CA Spectrum to Forward to CA SOI.
3. View Logs to Track the Flow of Alerts.
   a. View Alert_RAW File.
   b. View Alert_PUB File.

Enable Debug to Create Debug Files

This procedure provides information about how to enable debug to create the debug files that you can use to track alerts from CA Spectrum to CA SOI.

Follow these steps:

1. Navigate to the SOI_HOME\resources folder on the computer where the CA Spectrum connector is installed.
2. Locate and open the log4j.xml file.
   The file opens in a text editor.
3. Scroll to the bottom of the file.
4. Change the IFW value from INFO to DEBUG as follows:

```
<!--
*****************************************************************************
* Root logger definition
*****************************************************************************
-->
    <!-- ROOT IFW LOGGER -->
    <logger name="com.ca.ssa.servicediscovery" additivity="false">
        <level value="INFO" />
                <appender-ref ref="SD" />
    </logger>
 <logger name="com.ca.ehealth" additivity="false">
```

```
        <level value="INFO" />
        <appender-ref ref="EH" />
    </logger>
     <logger name="com.ca.sam.ifw" additivity="false">
        <level value="DEBUG" />
        <appender-ref ref="IFW" />
    </logger>
    <logger name="com.ca.ucf" additivity="false">
        <level value="INFO" />
        <appender-ref ref="UCF" />
    </logger>
   <root>
        <level value="ERROR" />
        <appender-ref ref="ROOT" />
    </root>
```

5. Save the file.
   The changes are saved and the log level is set to DEBUG.

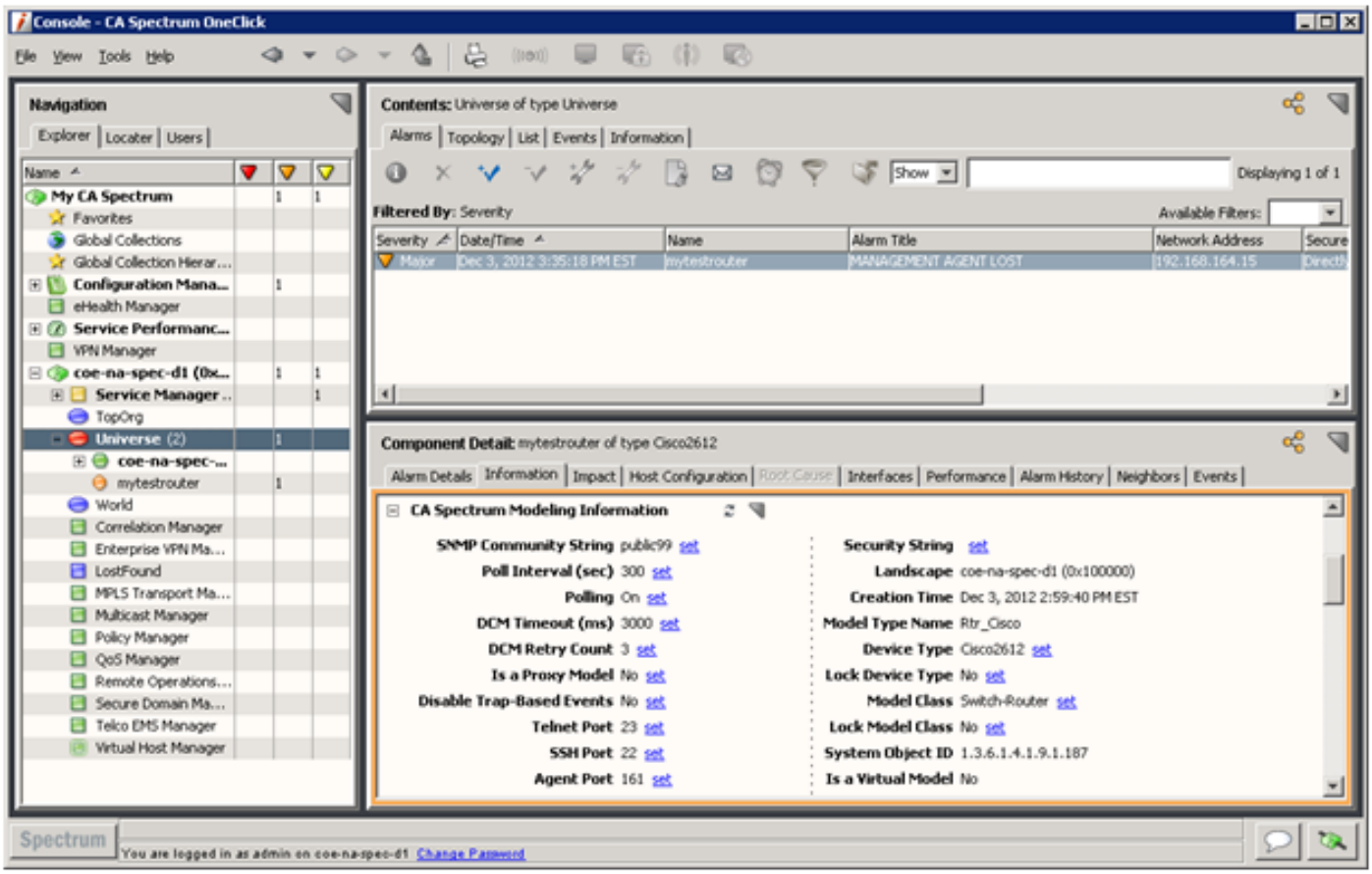
   NOTE
   You do not need to restart anything for the changes to take effect or to create the debug files.

Generate an Alert in CA Spectrum to Forward to CA SOI

After you enable the debug, generate an alert in CA Spectrum. For this scenario, you create an alert in CA Spectrum by changing the community string on a device to generate a MANAGEMENT AGENT LOST alert.

Follow these steps:

1. Launch the CA Spectrum OneClick console.
2. Select the appropriate device and click the Information tab in the Component Detail panel.
3. Navigate to the CA Spectrum Modeling Information section.
4. Click the *set* link next to SNMP Community String.
5. Add 99 to the end of the community string and press enter.
6. Right-click the device and select Poll to speed up the time it takes for the alert to appear in CA Spectrum OneClick.
7. Verify that your screen looks as follows (after the alert is created):



8. Verify that you can see the alert in CA Spectrum OneClick and CA SOI.

View Logs to Track the Flow of Alerts

To track the flow of the alert from CA Spectrum to CA SOI, view the debug RAW and PUB files:

1. View Alert_RAW File.
2. View Alert_PUB File.

View Alert_RAW File

The Alert_RAW file contains *raw* alerts coming from CA Spectrum. The information in this file also signifies that the alert that was generated in CA Spectrum has been sent successfully to the CA Spectrum connector.

Therefore, if you see your particular alert in this file, it implies that the alert has been sent from CA Spectrum to the connector.

Follow these steps:

1. Navigate to the SOI_HOME\log\debugData folder on the server where the CA Spectrum connector is installed.
2. Open the CA_00005_Spectrum-<*SShostname*>_Alert_RAW.txt file.

   NOTE
   SShostname specifies the host name of the CA Spectrum server.
3. Verify that the new MANAGEMENT AGENT LOST alert is present in the CA_00005_Spectrum-<*SShostname*>_Alert_RAW.txt file as shown in the following screenshot:

CA_00005_Spectrum-spec-d1_ALERT_RAW.txt - Notepad

File Edit Format View Help

```
cause=0x10701
connectorID=21aa40aa-b6bc-4fa1-a8b2-0bab75582bfc
causeText=MANAGEMENT AGENT LOST
User_Clearable=false
VirtualEntityType=0
type=Risk-Fault
alarmID=50bd0d06-0190-1004-0126-005056a906b1
siloName=CA:00005_Spectrum-spec-d1@soi-d1
publisher=JMS
Model_Handle=0x1002e1
Model_Class=4
publishAction=Add
Trouble_Ticket_ID=
siloHost=Spectrum-spec-d1
entitytype=Alert
Modeltype_Handle=0x21000c
LastModActivity=Update
Creation_Date=1354566918
severity=Major
domainName=spec-d1
ConnectorConfigMdrProduct=CA:00005
Model_Name=mytestrouter
Alarm_Status=
siloID=969aba27-4a40-4585-8a81-76158347393f
TroubleShooter=
connectorName=soi-d1
eventDetail=Device mytestrouter of type Rtr_Cisco is no longer responding to primary management requests (e.g. SNMP),
but appears to be responsive to other communication protocol (e.g. ICMP).  This condition has persisted for an extended
amount of time.  An alarm will be generated.
ConnectorConfigMdrProdInstance=Spectrum-spec-d1
eventtype=Alert
SymptomList=0.0.0.0
Acknowledged=false
```

View Alert_PUB File

The information in the Alert_PUB file signifies that the alert that was in the Alert_RAW file has now been transformed using the USM properties. The details also imply that the alert has been sent to CA Catalyst and should show in the CA SOI Operations Console.

Follow these steps:

1. Navigate to the SOI_HOME\log\debugData folder on the server where the CA Spectrum connector is installed.
2. Open the CA_00005_Spectrum-<*SShostname*>_Alert_PUB.txt file.

   NOTE
   SShostname specifies the host name of the CA Spectrum server.
3. Verify that the new MANAGEMENT AGENT LOST alert is present in the CA_00005_Spectrum-<*SShostname*>_Alert_PUB.txt file as shown in the following screenshot:

CA_00005_Spectrum-spec-d1_ALERT_PUB.txt - Notepad

File Edit Format View Help

```
AlertedMdrProduct=CA:00005
AlertedMdrProdInstance=Spectrum-spec-d1
MdrProduct=CA:00005
IsClearable=false
Message=Device mytestrouter of type Rtr_Cisco is no longer responding to primary management requests (e.g. SNMP), but
appears to be responsive to other communication protocol (e.g. ICMP).  This condition has persisted for an extended
amount of time.  An alarm will be generated.
connectorID=21aa40aa-b6bc-4fa1-a8b2-0bab75582bfc
MdrProdInstance=Spectrum-spec-d1
AlertedMdrElementID=0x1002e1
SAMID=
siloName=CA:00005_Spectrum-spec-d1@soi-d1
publisher=JMS
Summary=MANAGEMENT AGENT LOST
publishAction=Add
IsAcknowledged=false
Severity=Major
AlertType=Risk-Fault
OccurrenceTimestamp=2012-12-03T15:35:18-05:00
siloID=969aba27-4a40-4585-8a81-76158347393f
MdrElementID=50bd0d06-0190-1004-0126-005056a906b1
IsAcknowledgeable=true
ReportTimestamp=2012-12-03T15:35:18-05:00
connectorName=soi-d1
ClassName=Alert
entitytype=Alert

AlertedMdrProduct=CA:00005
AlertedMdrProdInstance=Spectrum-spec-d1
IsClearable=false
connectorID=21aa40aa-b6bc-4fa1-a8b2-0bab75582bfc
AlertedMdrElementID=0x1002e1
siloName=CA:00005_Spectrum-spec-d1@soi-d1
SAMID=
AlertedSiloName=CA:00005_Spectrum-spec-d1@soi-d1
publisher=JMS
publishAction=Add
Severity=Major
AlertType=Risk-Fault
OccurrenceTimestamp=2012-12-03T15:35:18-05:00
IsAcknowledgeable=true
MdrElementID=50bd0d06-0190-1004-0126-005056a906b1
ReportTimestamp=2012-12-03T15:35:18-05:00
ClassName=Alert
entitytype=Alert
siloHost=soi-d1
dns_resolution=0
MdrProduct=CA:00005
Message=Device mytestrouter of type Rtr_Cisco is no longer responding to primary management requests (e.g. SNMP), but
appears to be responsive to other communication protocol (e.g. ICMP).  This condition has persisted for an extended
amount of time.  An alarm will be generated.
AlertedConnectorName=soi-d1
MdrProdInstance=Spectrum-spec-d1
Summary=MANAGEMENT AGENT LOST
IsAcknowledged=false
siloID=beec7939-e4b3-4813-9f24-fb696f3bc204
connectorName=soi-d1
eventtype=Alert
```

Therefore, an alert that was originally generated in CA Spectrum shows in the Alert_RAW file and the Alert_PUB file. This flow indicates that this alert was forwarded successfully from CA Spectrum to CA SOI and is now displayed in the CA SOI Operations Console.

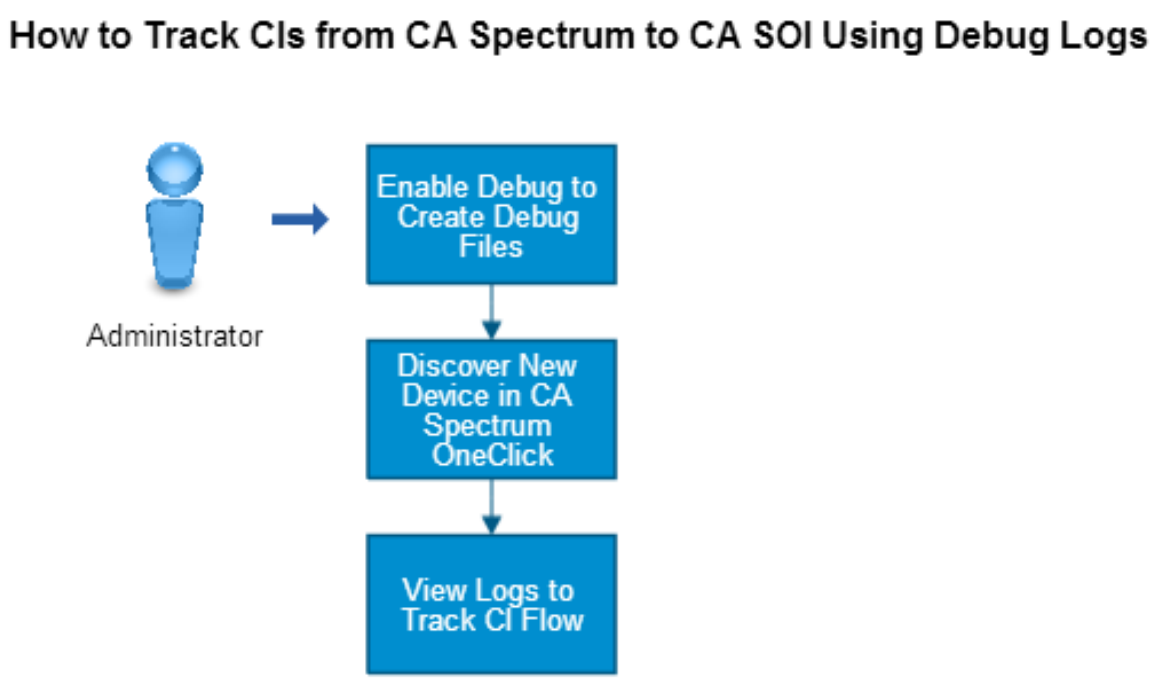
How to Track CIs from CA Spectrum to CA SOI Using Debug Logs

Contents

As a CA SOI administrator, you can enable debug on the CA Spectrum connector and can investigate the debug logs to track CIs as they flow from CA Spectrum to CA SOI.

Use this scenario to guide you through the process:

Figure 57: how to track cis from Spectrum



1. Enable Debug to Create Debug Files.

   NOTE
   If the debug level is already set to DEBUG, do not perform this step. The steps are similar to the information already mentioned in the alert procedure.
2. Discover a New Device in CA Spectrum OneClick.
3. View Logs to Track the Flow of CIs.
   a. View ITEM_RAW File.
   b. View CI_PUB File.

Discover a New Device in CA Spectrum OneClick

Launch the CA Spectrum Oneclick console and discover a new device.

NOTE
For more information about working with the CA Spectrum Oneclick console, see the CA Spectrum documentation.

View Logs to Track the Flow of CIs

To track the flow of new CIs that are added to CA Spectrum into CA SOI, view the debug RAW and PUB files:

1. View ITEM_RAW File.
2. View CI_PUB File.

View ITEM_RAW File

The ITEM_RAW file contains *raw* details of the newly discovered device in CA Spectrum. The information in this file signifies that the device in CA Spectrum was sent successfully to the CA Spectrum connector.

Therefore, if you see your particular device in this file, it implies that the device has been sent from CA Spectrum to the connector.

Follow these steps:

1. Navigate to the <SOI_HOME>\log\debugData folder on the server where the CA Spectrum connector is installed.
2. Open the CA_00005_Spectrum-<*SShostname*>_ITEM_RAW.txt file.

   NOTE
   SShostname specifies the host name of the CA Spectrum server.
3. Verify that you see the details of the new device and its child CIs (such as interfaces and ports) in the ITEM_RAW file as shown in the following screenshot:

CA_00005_Spectrum-spec-d1_ITEM_RAW.txt - Notepad

File Edit Format View Help

```
connectorID=
USMRedundancyTypes=Unknown
VirtualEntityType=0
DeviceType=ProLiant DL380
sysName=DEVFAX01
siloName=CA:00005_Spectrum-spec-d1@con-d1
IsVirtual=0
publisher=JMS
isManaged=1
createtime=1360940024
Model_Handle=0x100386
publishAction=Add
Model_Class=9
Modeltype_Name=Host_Compaq
USMProtocoltypes=Unknown
Dev_Contact_Status=1
siloHost=Spectrum-spec-d1
entitytype=Item
USMPrimaryIPV6Address=
Modeltype_Handle=0x1160069
MAC_Address=
Description=
USMOtherIPAddresses=
USER_AssetID=
domainName=spec-d1
Serial_Number=D044FK41K793
Vendor_Name=Compaq Computer Corporation
Significant_Model_ID=0x100386
Network_Address=
ConnectorConfigMdrProduct=CA:00005
Model_Name=DEVFAX01
ModelClassName=Workstation-Server
Criticality=1
siloID=3a8f0260-8eae-46d5-8f32-eb6f6ddf4a1f
connectorName=con-d1
NRM_RunningFirmware=5.0
ConnectorConfigMdrProdInstance=Spectrum-spec-d1
eventtype=Item
USMOtherMACAddresses=

connectorID=
USMDeviceAssetNumber=
DeviceType=
Component_OID=1
sysName=DEVFAX01
siloName=CA:00005_Spectrum-spec-d1@con-d1
publisher=JMS
isManaged=1
```

View CI_PUB File

The CI_PUB file contains details of the transformed CI and its child CIs. The information in the file signifies that the CI that was in the Item_RAW file has now been transformed using the USM properties. The details also imply that the CI has been sent to CA Catalyst and should show in the CA SOI Operations Console.

Follow these steps:

1. Navigate to the <SOI_HOME>\log\debugData folder on the server where the CA Spectrum connector is installed.
2. Open the CA_00005_Spectrum-<*SShostname*>_CI_PUB.txt file.

   NOTE
   SShostname specifies the host name of the CA Spectrum server.
3. Confirm the presence of the new CI in the CI_PUB file as shown in the following screenshot:

CA_00005_Spectrum-spec-d1_CI_PUB.txt - Notepad

File Edit Format View Help

```
Model=ProLiant DL380
CreationTimestamp=2013-02-15T09:53:44-05:00
connectorID=
PrimaryIPV4Address=
siloName=CA:00005_Spectrum-spec-d1@con-d1
publisher=JMS
publishAction=Add
MdrElementID=0x100386
OtherMacAddresses=
ClassName=ComputerSystem
entitytype=CI
AdministrativeStatus=Active-Available
PrimaryMacAddress=
MdrProduct=CA:00005
LastModTimestamp=2013-02-15T09:53:44-05:00
Description=
LastModActivity=Create
PhysSerialNumber=
MdrProdInstance=Spectrum-spec-d1
ComputerName=
Tags=System-Layer-Hardware
OtherIPAddresses=
IsInMaintenance=false
Vendor=Compaq Computer Corporation
siloID=3a8f0260-8eae-46d5-8f32-eb6f6ddf4a1f
Label=DEVFAX01
connectorName=con-d1
SysName=DEVFAX01

CreationTimestamp=2013-02-15T09:53:44-05:00
connectorID=
DeviceMacAddress=
DevicePhysSerialNumber=
PrimaryIPV4Address=
siloName=CA:00005_Spectrum-spec-d1@con-d1
DeviceDnsName=null
publisher=JMS
publishAction=Add
DeviceSysName=DEVFAX01
MdrElementID=0x100390
IsPhysical=true
ClassName=Port
entitytype=CI
InstanceName=Port:softwareLoopback:1:Module:Unknown:1
AdministrativeStatus=Active-Available
MdrProduct=CA:00005
LastModTimestamp=2013-02-15T09:53:44-05:00
IfType=softwareLoopback
```

4. Verify that the new CI and its child CIs now show up in the CA SOI Service Modeler.

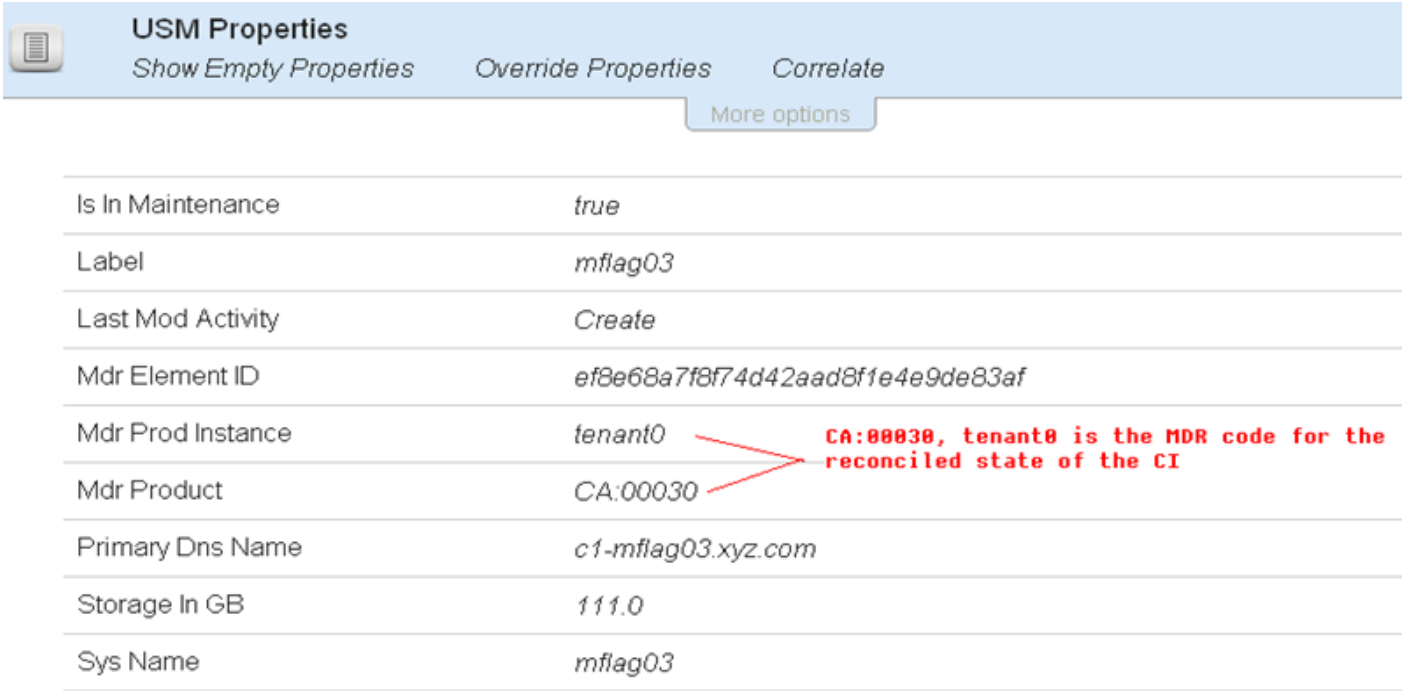
Therefore, a CI that was discovered in CA Spectrum shows in the Item_RAW file and the CI_PUB file. This flow indicates that this CI was forwarded successfully from CA Spectrum to CA SOI and is now displayed in the CA SOI Operations Console.

Trace a CI Using USM Web View for Diagnosing Synchronization Errors

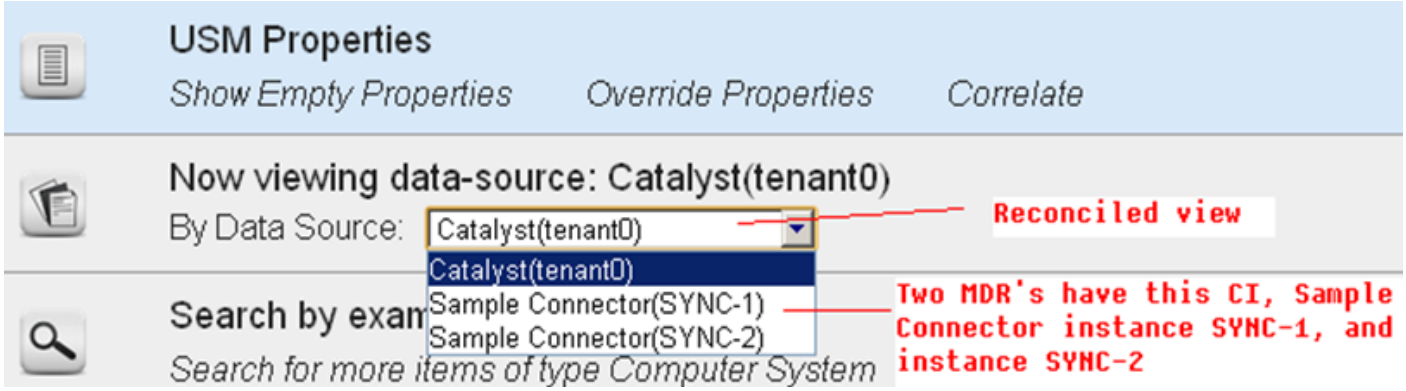
As an administrator, you can USM Web View to view the current state of a CI as CA Catalyst views it. You can use the USM Web View interface to get the CA Catalyst notebook ID and use it for tracing purposes. For example, to troubleshoot synchronization issues, you can use the notebook ID in the CA Catalyst Trace UI and can track the flow of tracing for the specific notebook. You can use that information to determine which component (Reconciler, Sync Planner, and Sync Executor) is creating the error and then review the error details.

Follow these steps:

1. Open the CA SOI Dashboard and click the USM Web View link.
   The USM Web View interface opens.
2. Search for a known CI by entering a valid keyword (for example, label and sysname) value in the Search field.
3. View the USM Properties section that displays a reconciled view of the CI (for example, CA:00030, tenant0) by default. CA:00030, tenant0 is the MDR code for the reconciled state of the CI.



4. Click the More options link to display additional options.
5. Use the By Data Source drop-down list to get different projections of the CI.
6. Expand the drop-down list to see which domain managers are associated with the CI.



7. Review the URL to get the internal CA Catalyst notebook ID for tracing purposes. The notebook ID is embedded in the URL. In this example, the notebook ID is 01A356C6861948B08E0FC6E0E1EBCA31. You can use this notebook ID in the CA Catalyst Trace UI for further analysis.

Use the CA Catalyst Trace UI for Diagnosing Synchronization Errors

Contents

As an administrator, use the CA Catalyst Trace UI to view the processing flow through the CA Catalyst components. The UI displays a list of actions that are performed by internal CA Catalyst components based on the notebook ID and transaction ID. A notebook ID represents the internal ID that CA Catalyst uses to reference a CI. A transaction ID represents a unique ID that is used to link together a sequence of operations that are performed on a CI. The entire processing flow shares the same transaction ID.

> **NOTE**
> For more information about the CA Catalyst Trace UI processing, see the CA Catalyst Trace UI Processing Flow section.

You can use the CA Catalyst Trace UI to determine what synchronizer is doing, whether it is succeeding or failing, and any reason for the failure. You filter the notebook ID in the UI and trace the processing flow to troubleshoot the error.

Follow these steps:

1. Enter the following URL in the web browser:
   `http://host:7090/sor.application/trace.jsp`
   – *host*
   Specifies the host name of the SA Manager server. For example, http://soi-server:7090/solr.application/trace.jsp.
   The CA Catalyst Trace UI page displays.
2. Click the Reload Trace Log link.
   The trace log is reloaded.
3. Specify the notebook ID in the Filter field and click Submit Query.
   The UI shows only traces for the specific notebook.

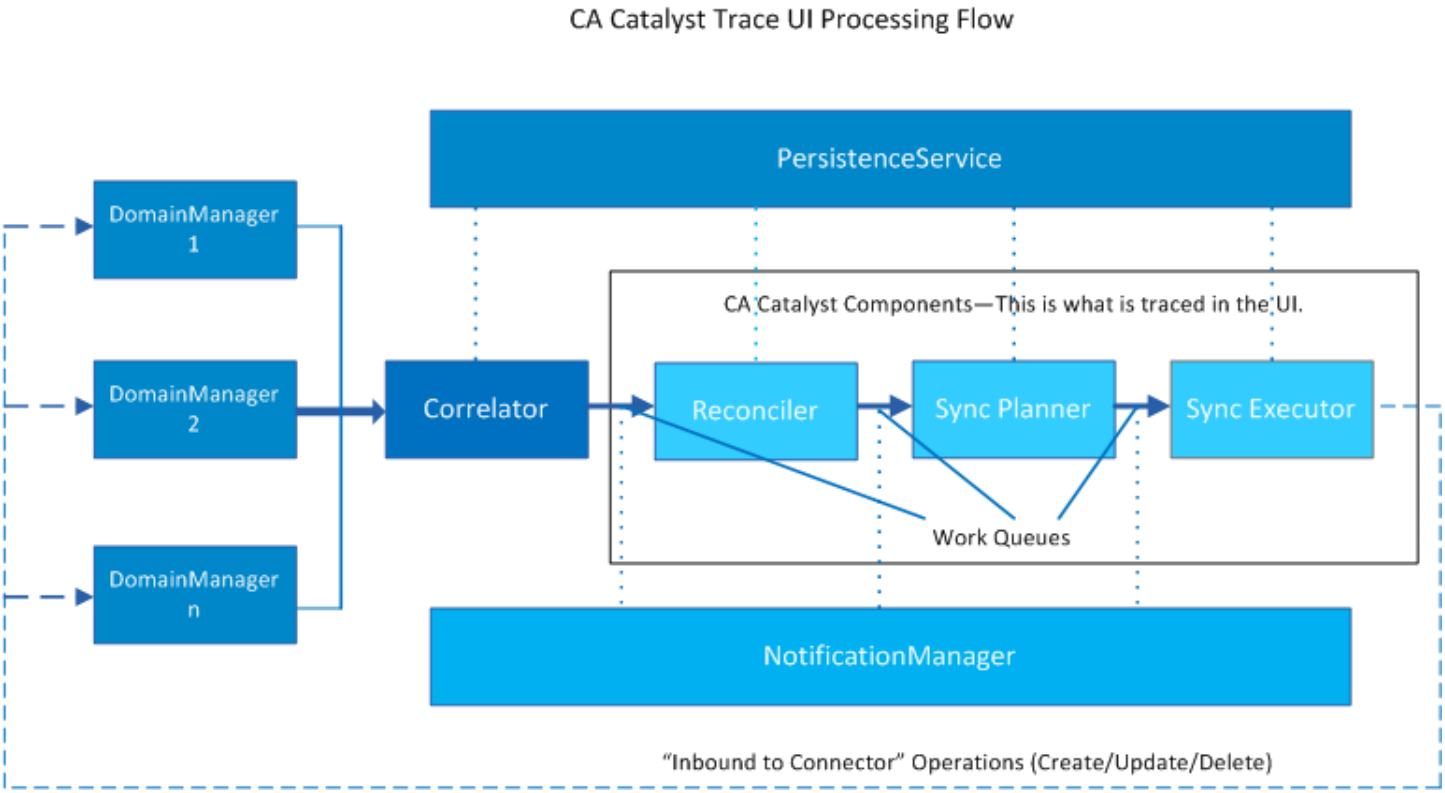
   > **NOTE**
   > For more information about how to find the notebook ID using USM Web View, see Trace a CI Using USM Web View for Diagnosing Synchronization Errors.
4. Click the link in the Timestamp column to view the trace in more detail. For example, if the Action column shows the value as Error, you can click the corresponding timestamp value to view the exception that was encountered.

The Trace details page opens and displays the detailed information.

5. Review the information.

CA Catalyst Trace UI Processing Flow

The following diagram shows the processing flow:



The components in the diagram are explained as follows:

- Reconciler: Creates or updates the reconciled view of a CI.
- Sync Planner: Creates a synchronization plan based on the changes to the reconciled view.
- Sync Executor: Executes the synchronization plan.

The actions that are involved in the process are as follows:

- Dequeued: Specifies that the component has taken a notebook off the work queue for processing.
- Processed: Specifies that the component has completed working on the notebook.
- Error: Specifies that the component has encountered an error while trying to complete work on the notebook.
- Retry: Specifies that the component has decided to retry work on the notebook at a later time due to a recoverable error.
- Email: Specifies that the component has stopped trying to complete the work, either due to an unrecoverable error or because it has retried a maximum amount of time. By default, a component stops after retrying for 25 minutes, with a retry interval of 1 minute.

Product Troubleshooting

This section describes how to troubleshoot problems with specific CA SOI components.

Important! Before Troubleshooting a Problem

Before you troubleshoot any problem in this section, run the Triage Tests on the SA Manager Server and the UI Server. The Triage Tests help you diagnose major problems in the system. Many minor problems that you experience can be the result of a much greater problem in the system.

If any Triage Test fails, resolve the Triage Test failure problem before attempting to resolve any other issue.

Once all Triage Test problems are resolved, then continue to the other troubleshooting topics in this section.

CA Catalyst Troubleshooting

Synchronization - Priming Utility Runs for Hours

Symptom:

I ran the priming utility to perform a full synchronization to the realtime environment. The utility ran for hours without finishing.

Solution:

As a best practice, synchronize the CI types first then relationships. For synchronization procedures, see Synchronization.

Depending your system and the number of CIs, the priming utility can run for hours or days before completing synchronization.

Also, while the priming utility runs, CA SOI typically appends many ObjectNotFound exceptions in the SA Manager log. The exceptions occur because CA SOI does not enforce sequencing of items and relationships. Therefore, CA SOI can process the relationships before the reconciled sheets are created for the parent, child, or relationship scope.

Installation or Initialization Errors

Symptom:

You experience installation errors or have problems initializing CA Catalyst components. The specific problems can include the following:

- Components do not install correctly.
- The Registry is inaccessible.
- SA Manager or IFW does not start.
- UCF failed.

Solution:

Try the following solutions to resolve common installation or initialization problems:

- Check the installation logs at <SOI_HOME>\log. The following log files are available for debugging CA Catalyst components at <SOI_HOME>\log:
  - CA-SSA-LogicInstallDebug.log
  - CA-SSA-RegistryInstallDebug.log
  - ucf.log
- Check the following services and ensure that they are all started:

  - CA SAM Application Server
  - CA SAM Integration Services
  - CA SAM Store Indexer
  - CA SAM User Interface Server
  - CA UCF Broker

    NOTE
    For more information about using services for diagnosis, see Using Services for Diagnosis.
- Check the services logs at <SOI_HOME>\jsw\logs if a service does not start.
- Check the CA SOI error logs if problems persist. CA Catalyst components run under the SA Manager, so problems with this and other CA SOI components can affect CA Catalyst components.
- For IFW problems not caused by a service failure, check the <SOI_HOME>\log\ifw.log file for errors. Verify the appropriate IFW configuration in the <SOI_HOME>\resources\Configurations\SSA_IFW_*servername*.xml file.

  NOTE
  For more information about using log files for diagnosis, see Using Log Files for Diagnosis.

ActiveMQ Server Errors

Symptom:

The ActiveMQ server must be operating properly for information to make it to the Logic Server. Problems with all CA Catalyst components can originate in the ActiveMQ server. The following problems can occur:

- ActiveMQ not started
- ActiveMQ connectivity problems

Solution:

Perform the following actions to resolve the errors:

- Check the <SOI_HOME>\tomcat\logs\soimgr-error.log file for errors that are related to ActiveMQ startup.
- Check the ActiveMQ log file for errors.
- Verify that activemq-web.war is deployed.
- Verify the ActiveMQ configuration in the <SOI_HOME>\tomcat\webapps\activemq-web\WEB-INF\caifwmq.xml file.
- Verify that the ActiveMQ connection information is correct in all appropriate configuration files.
- Verify the ActiveMQ topic and queue permissions.

Registry Errors

Symptom:

You experience errors with Registry functionality, such as the following issues:

- Problems connecting to the SA Store database
- Cannot access Registry
- Registry content missing
- The sor.application program within the Registry is not starting

Solution:

Try the following solutions to fix Registry errors:

- Verify that the SA Store database credentials in the <SOI_HOME>\ws02registry\repository\conf\registry.xml file are correct.
- Access problems can be caused by invalid credentials or an unencrypted password that is stored in the Registry. If you cannot access the Registry, check the credentials in the ssaserver.xml and sorapp.xml files at <SOI_HOME>\tomcat\registry\topology\physical\node0\sor.
- If Registry content is missing, check the database credentials in the sorapp.xml and ssaserver.xml file.
- Check the <SOI_HOME>\tomcat\registry\topology\logical\tenant0\usmschema folder and verify that the following files exist:
  - addressing.xsd
  - sml.xsd
  - usm-core-200907.xsd
  - usm-query.xml
  - usm-elemconstraints.xml
  - usm-extensions-201001.xsd
  - usm-infradefaults.xml
  - usm-metadata-200907.xsd
  - usm-metrics-200907.xsd
  - usm-openenums.xml

  If these files do not exist or are corrupted, the USM schema did not install correctly.
- If sor.application does not start, check the topology\physical\node0\sorappbootstrap.xml file and verify that it correctly points to the sorapp.xml file. Also, verify that sor.application.war is deployed.

Correlation and Persistence Errors

Symptom:

You experience errors with the correlation and the information added to the database by the Persistence Service, which can include the following issues:

- No notebooks created
- No projection sheets created
- Correlation did not occur
- Missing correlated events

Solution:

Try the following solutions:

- Verify that the SA Manager is initialized.
- Verify that the Persistence Service is initialized.
- Verify that the Notification Manager is initialized with the correct configuration.

Notification Manager Errors

Symptom:

You experience problems with correlation or other functions, and log file investigation pinpoints the Notification Manager as the source of the problem. The Notification Manager must be initialized and configured correctly.

Solution:

Try the following solutions:

- Verify that the MQ server is initialized.
- Verify that the following topics are defined in the sorapp.xml and ssaserver.xml files:
  - PlansToBeApproved
  - SynchronizationPlans
  - ReconciledSheets
  - InstructionQueue
  - CorrelatedNotebooks

Reconciliation Errors

Symptom:

You experience errors that are related to the reconciliation, which include the following issues:

- Outbound from connector operations failed
- No reconciled sheet is created
- Reconciler does not initialize

Solution:

Try the following solutions:

- Verify that all components installed and initialized properly.
- Verify the correct operation of the correlation and Persistence Service components.
- Verify that reconciliation policy is configured correctly. Verify the existence of the defaultsheet.xml file in the Registry and the Reconciler contents in the sorapp.xml and ssaserver.xml files.

Synchronization Errors

Symptom:

You experience problems with the synchronization functionality. Synchronization errors usually appear as failed inbound to connector operations or mismatches between projections and reconciled sheets. Synchronization errors may be caused by the following issues:

- Sync Planner failed
- Sync Executor failed
- Synchronization not enabled

Solution:

Try the following to fix synchronization errors:

- Review Synchronization. The section provides detailed information about making sure that everything is configured correctly from the connector configuration to UCF Broker to CA Catalyst registry.
- Verify that the Synchronizer is enabled. Enabling the maintenance mode synchronization automatically enables the Synchronizer.
- Verify that the connector to which you are trying to synchronize is configured properly. Open the CA SOI Administration UI, select the connector, and review the following information:
  - In the Connector Controls section, verify that the isRemotable connector control is enabled.
  - In the InboundToConnectorTypes section, verify that a list of all USM types that are supported for synchronization exists. To synchronize on a particular CI, its type must be included in the list.
- Verify the correct location of the UCF Broker.

NOTE
For more information about how to configure the UCF Broker location, see How to Enable Alert Synchronization.

- Verify that all applicable connectors support inbound to connector operations.
- Verify that all connectors have inbound to connector operations enabled.
- Verify that sor.application started correctly.
- Verify that the CI on which synchronization is required has been reconciled with CA SOI. Check for mismatches between projections and reconciled sheets using the USM Web View interface:
  - Find the reconciled view of the CI in USM Web View (for example, CA:00030, tenent0).
  - Get different projections for the CI using the *By Data Source* drop-down list.

NOTE
For more information about these points, see Trace a CI Using USM Web View for Diagnosing Synchronization Errors.

- Verify that the Synchronizer is executing the synchronization transactions.
  - Get the internal CA Catalyst notebook ID for the CI using the USM Web View interface.
  - Use the CA Catalyst Trace UI (http://<host>:7090/sor.application/trace.jsp) to track events on the notebook using the notebook ID.
- Review the related log file. CA Catalyst uses log4j for logging. By default, only Errors and Warnings are logged to the <SOI_HOME>\tomcat\logs\soimgr.log file. Enable the debug logging in CA Catalyst by editing the <SOI_HOME>\tomcat\lib\log4j.xml file. Update the existing logger entry in the file by changing the debug level from INFO to DEBUG and restart the CA SAM Application Server service:

```
<logger name="com.ca.ssa.sor" additivity="false">
    <level value="DEBUG" />
    <appender-ref ref="CAT" />
</logger>
```

- CA Catalyst debug log entries start appearing in the soimgr.log file.
  Failures can also occur in connectors when fulfilling the update request. Review the individual connector logs to view the request and any errors that occur.

XML of CI Attributes Before and After EI transformation not Available

Symptom: I cannot view the XML of all CI attributes before and after EI transformation.

Solution: Follow these steps:

1. Debug the CI mapping and the UIM connector.
2. Enable DEBUG level in the following lines:
   - log4j.logger.com.ca.catalyst.intercepter.ei=DEBUG, eiinterceptor
   - log4j.logger.com.ca.connector.impl.util.ei= DEBUG, eiinterceptor
   - log4j.logger.com.ca.eventplus.catalog= DEBUG, Transformer
   - log4j.logger.com.ca.eventplus.catalog.test= DEBUG, Transformer
3. Restart the catalyst container service
4. Enable DEBUG in Container\etc\log4j-ei.properties.
   Two files are created in the CA\Catalyst\CatalystConnector\container\data\log\EIDebugData folder. One file contains the Alert and the other contains the CI mapping in the UIM connector. Both the files contain what is being intercepted and transformed.

Exception Occurs when the Container Service is Stopped

Symptom:

An interrupted exception occurs when the Catalyst Container service is stopped and the following log appears.

```
INFO | jvm 1 | 2016/11/24 06:26:37 | Please wait while connector cleans up its
 resources....
INFO | jvm 1 | 2016/11/24 06:26:37 | java.lang.InterruptedException: sleep interrupted
INFO | jvm 1 | 2016/11/24 06:26:37 | at java.lang.Thread.sleep(Native Method)
INFO | jvm 1 | 2016/11/24 06:26:37 | at com.ca.soi.catalyst.IFWProxy
$2.run(IFWProxy.java:405)
INFO | jvm 1 | 2016/11/24 06:26:37 | 2016/11/24 06:26:37 : US-AZR-
PEEM02_CatalystConnector ConnectorService : CLOSING : Broadcasting HeartBeat Message
INFO | jvm 1 | 2016/11/24 06:26:37 | 2016/11/24 06:26:37 : US-AZR-
PEEM02_CatalystConnector : CA:00050_US-AZR-PPHUB01@US-AZR-PEEM02_CatalystConnector :
 ONLINE - Connector was stopped due to CA Catalyst Container Service stop : Broadcasting
 HeartBeat Message
INFO | jvm 1 | 2016/11/24 06:26:37 | Stopping CA:00050_US-AZR-PPHUB01@US-AZR-
PEEM02_CatalystConnector. Connector was stopped due to CA Catalyst Container Service
 stop
```

Solution:

This log is not an exception but a normal process log. This log appears when the container service is stopped.

Connectors Troubleshooting

> **TIP**
>
> To be proactively notified of connector failures, consider enabling CA SOI self monitoring.

For information related to specific connectors, see the respective *Connector Guide* provided with the connector.

Alert Synchronization Not Working for a Connector

Symptom:

I am trying to implement alert synchronization (*inbound to connector* operation) for my connector, but it is not working. How do I troubleshoot this issue?

Solution:

For *inbound to connector* synchronization, the update() method of connectors is called with the alert details. If the update() method is not getting called, this problem usually means an issue in either the connector configuration or CA Catalyst synchronizer policy. To troubleshoot, perform the following steps:

- Verify that the connector is enabled for the inbound to connector operations. To do so, verify that the value of the *isRemotable* connector control is set to 1 in the connector configuration file.
- Verify that the connector is exposing *Alerts* as a supported USM type for synchronization (*inbound to connector*). You can do so by verifying that the connector configuration file has *Alert* listed under *InboundToConnectorTypes*.
- Enable CA Catalyst synchronizer from the CA SOI Administration UI.
- The default synchronization policy that is shipped with the product enables synchronizations (inbound to connector) for a specific connector or (domain manager). You may need to add your connector to this list as follows:
  a. Open the <SOI_HOME>\tomcat\registry\topology\physical\node0\sor\syncfilters\alert_filter.xml file in a text editor.
  b. Add your domain manager product identifier (for example, CA:01234) to the list of domain managers in the file. An example is provided as follows:

```
<action defaultBehavior="drop">
  <send>
    <mdr productName="CA:00003" /> <!-- NSM-DSM -->
     <mdr productName="CA:00005" /> <!-- SPIM -->
     <mdr productName="CA:00031" /> <!-- SCOM -->
     <mdr productName="CA:01234" /> <!? Your connector -->
  </send>
 </action>
```

  c. Reload the registry and restart the CA SAM Application Server service for the change to take effect.

Anti-virus Programs Affecting the Connector Performance

Symptom:

My anti-virus programs are impacting the connector performance. How do I improve the performance?

Solution:

Anti-virus programs that perform real-time scans on open or updated files can affect the connector performance. For example, real-time scans could be occurring every time a connector log file is updated.

To eliminate this problem, you can exclude the following files in the CA SOI installation directory from the real-time virus scans:

- *.log
- *.xml
- *.xsd
- *.txt
- *.out
- *.conf
- *.tmp

CA Spectrum Connector Binding in a Dual NIC Environment

Symptom:

CA Spectrum alerts (through the CA Spectrum connector) are not showing up in CA SOI in an environment that uses dual NIC servers.

Solution:

On dual NIC servers, binding order on both CA Spectrum connector and CA Spectrum servers needs to be correct, and must bind to the same NIC.

CA Spectrum must not bind to the secondary, non-routable IP address on the connector server. Setting the proxy host can help in this dual NIC case when the CA Spectrum server can only connect to one of the two IPs/FQDNs for the connector server.

As a workaround, hard code the primary IP address of the correct NIC in the <SOI_HOME>\jsw\conf\SAM-IntegrationServices.conf file on the connector server. For example:

wrapper.java.additional.*number*=-Dvbroker.se.iiop_tp.host=

wrapper.java.additional.*number*=-Dvbroker.se.iiop_tp.proxyHost=*proxyhostname*

proxyhostname in this case is the local IP or hostname that the remote server can use to connect back.

CA Spectrum Connector Firewall Limitations

Symptom:

CA Spectrum alerts are not showing up in CA SOI (after connector initialization) in an environment that uses a firewall between the CA Spectrum server and CA Spectrum connector.

Solution:

If the firewall exists between the CA Spectrum server and CA Spectrum connector, the CA Spectrum connector uses a random listener port (instead of the bidirectional static port) to communicate with the CA Spectrum server to receive callbacks. Because of this reason, no new alerts are passed on to CA SOI after the connector initialization phase.

As a workaround, hard code the listener port on the connector server to a static port (for example, 14001) in the <SOI_HOME>\jsw\conf\SAM-IntegrationServices.conf file, and open the port bidirectionally on the firewall. For example, wrapper.java.additional.*number*=-Dvbroker.se.iiop_tp.scm.iiop_tp.listener.port=14001. This port has to be open from the CA Spectrum server to the connector server.

CA Spectrum Connector Keeps Reinitializing

Symptom:

After I have installed CA Spectrum in a Microsoft Cluster Server environment, the CA Spectrum connector fails to start (it keeps reinitializing).

Solution:

Each node in the cluster must have at least two network adapters. One adapter is used for the client public network and the second one is used for internal cluster communication.

If the order of adapters and bindings is wrong, the CA Spectrum connector fails to start. The adapter for public network must be listed first in the Adapters and Bindings list in Windows Network Connections Advanced Settings.

Change Connector Credentials

Symptom:

How do I change the username and password for a connector?

Solution:

You can change the username and password in the respective connector configuration file.

Follow these steps:

1. For each connector, the configuration file is located in the following folder:
   <SOI_HOME>\resources\Configurations\<connectorname_hostname>.xml

2. Open the configuration file for editing.
3. Locate the username and password fields.
   Because the password is encrypted, you next run the encryption utility.
4. Open a command line and locate the encryption utility in the following folder:
   <SOI_HOME>\Tools\EncryptSAMCreds.bat
5. Run the following command:
   ```
   EncryptSAMCreds.bat password_to_encrypt
   ```
6. Copy and paste the encrypted password in the properties file password field.
7. Save the properties file.
8. Restart the CA SAM Integration Services service to implement the changes.

   NOTE
   For more information about managing passwords in CA SOI, see Password Maintenance.

Connector Data Not Imported

Symptom:

CA SOI is not importing connector data.

Solution:

There are two possible reasons:

1. The connection to the SA Store database may have failed. Configure the database connection failure email notifications. If there is a failure, the email provides the resolution.
2. A connector connection has failed. Configure the connector failure email notifications. If there is a failure, the email provides the resolution.

Connector Not Online

Symptom:

I started my connector, but the connector status is not Online.

Solution:

Review the following points to determine why the connector is not Online after it is started:

- The connector could have trouble connecting to its domain manager, or it may not have published the data to the SA Manager yet.
- Use the status description of the connector, and, if necessary, the appropriate log files to determine whether there is an issue.
- Check the Status Description field for the connector in the CA SOI Administration UI. You can also enable DEBUG to see details of the connector status in its <ConnectorName>_HEARTBEAT_PUB.txt file.
  For example, if Waiting for Connector Specific Initialization is the last status in the <ConnectorName>_HEARTBEAT_PUB.txt file, there can be a problem with connecting to the domain manager. Make sure that the domain manager is accessible, and check the connection details that are provided for the domain manager in the connector configuration.
- Check the SAM-IntegrationServices_wrapper.log file for specific messages regarding the connector. This information can help you identify whether the connector is having issues connecting to the domain manager.
- Connector names have been added to messages in the ifw.log file to make it easier to trace individual connector processing. You can review these messages to get more specific information.

NOTE
For more information about troubleshooting connector shutdown behavior, see How to Troubleshoot Connector Shutdown Behavior.

Connector Unable to Connect to the SSL-Enabled Domain Manager

Symptom:

A connector is unable to connect to the SSL-enabled domain manager.

Solution:

If SSL is enabled on the integrated domain manager, configure the connector to connect to the domain manager using SSL. You also must ensure that the CA SOI Dashboard can launch an SSL-enabled domain manager UI.

Follow these steps:

1. Open <SOI_HOME>\resources\Configurations\<connector configuration file> on the connector system.
2. Locate the LICURLS section, provide the appropriate HTTPS protocol and port values, and save the file.
   The file contains the correct information for connecting to the domain manager configured for SSL.
3. Verify that you can launch the domain manager UI from the connector system using the specified protocol and port information.
4. Import the security certificate for your integrated domain manager as follows:
   a. Obtain the installcert java file from a reliable location and download it to <SOI_HOME>\jre\bin on the connector system.
   b. Use the JDK installed with CA SOI on the connector system to compile the program from the command line as follows:
      ```
      <SOI_HOME>\jre\bin\javac InstallCert.java
      ```
   c. Run the compiled program using the following command:
      ```
      java InstallCert <DomainServer>:<SSLport>
      ```
      - **DomainServer**
        Specifies the domain manager host name. This name must match the URL host entry that is specified in the connector configuration file. In many cases, the name is the short form of the host name and not the FQDN. If the short name is used, then the URL host entry must also be the short form.
      - **SSLport**
        Specifies the SSL port.

      The program runs, and a prompt appears to save the certificate to a trusted keystore.
   d. Enter 1 to save the certificate to a key store file named jssecacerts, which gets created in the local directory.
   e. Copy the jssecacerts key file to <SOI_HOME>\jre\lib\security.
5. Restart the CA SAM Integration Services service.

   NOTE
   Verify that the host name in the LICURLS section of the connector configuration file matches the host name that is used. Change the name if necessary, and save and close the file.

The connector is configured to connect to the SSL-enabled domain manager.

How Do I Troubleshoot the Connector Policy?

Symptom:

I am facing some issues with my connector, and I want to troubleshoot the connector policy.

Solution:

The eitransform.log file contains information about all policy operations (such as parse, normalize, and format) included in your connector policy. This log file helps you find out whether the information is getting processed correctly, in case you encounter any issues. Enable this log file before you can start using it for troubleshooting problems with your connector policy. You can find the eitransform.log file at <SOI_HOME>\log after it is enabled.

> **NOTE**
> The size of the eitransform.log file grows quickly, so use caution when enabling it.

Follow these steps:

1. Locate and open the log4j.xml file available at <SOI_HOME>\resources.
2. Uncomment the following entry, and save and close the file:

```
<!-- FOR EI TRANSFORMATION DETAIL -->
<logger name="com.ca.eventplus.catalog" additivity="false">
<level value="DEBUG" />
    <appender-ref ref="EITransform" />
</logger>
```

   The eitransform.log file is now enabled.
3. Restart the CA SAM Integration Services service.

How Do I Troubleshoot Connector Connection Problems?

Symptom:

I am having some connection problems with my connector (that the IFW manages). How can I troubleshoot this issue?

Solution:

To verify whether connectors (running under the IFW) are online and running in CA SOI, use the Connection Status button in the Operations Console. If the connector does not appear or shows as offline, do the following tasks to troubleshoot the problem:

- Enable self monitoring to be proactively notified of connector problems.
- Check the install log file specific to your connector in the <SOI_HOME>\log directory for connection errors.
- Check the connector configuration file under <SOI_HOME>\resources\configurations and verify that it has the correct connection information. You can also access the content from the Administration tab.
- Ping the domain manager server from the computer where the connector is installed and verify that you can connect.
- Review the following:
  - SAM-IntegrationServices_wrapper.log file
  - Heartbeat history for the connector
  - Connector message in the UI
  - Email notification for the connector

How to Troubleshoot Connector Shutdown Behavior

Contents

As a CA SOI administrator, you need immediate notification whenever any connector shuts down in your IT infrastructure so that you can take quick actions and can resolve the issue. CA SOI provides notifications and detailed logging whenever a connector shuts down. This improved and faster notification mechanism contains detailed information about the connector shutdown behavior, including an appropriate reason for the shutdown. The availability of the relevant information in the log messages helps you troubleshoot connectors, keep track of the connector status in your infrastructure, and take any prompt actions, if necessary. You can review the related log files to find more information

about any anomaly that you encounter in the connector behavior. The easy identification of the problem that is associated with the connector also lets you manage your domain managers more efficiently.

The Integration Framework (IFW) as part of its heartbeat mechanism sends detailed description of the connector status to the SA Manager. The IFW sets and maintains a connector property *statusDesc* for this purpose. The statusDesc property includes information about the connector status and any error conditions that occur. The SA Manager reads and stores the property value to display it to the user in the product UI. The connector status information is also logged in the SAM-IntegrationServices_wrapper.log file and <ConnectorName>_HEARTBEAT_PUB.txt file.

The enhanced connector notification and logging mechanism, therefore, helps you as follows:

- Logs the appropriate reason (for example, Administration UI stop, connector failure, IFW shutdown) for the connector shutdown in the log file, SAM-IntegrationServices_wrapper.log. Review the log file to view the message; an example message that is included in this log file is as follows:

  ```
  2012/11/26 01:33:57 : abc-vm02.xy.com : CA:00056_service-discovery@abc-vm02.xy.com : OFFLINE - Connector
   was stopped from Administration UI. : Broadcasting HeartBeat Message
  ```

- Includes the shutdown message in the connector status notification as part of the *heartbeat* message. The heartbeat message is logged in the <ConnectorName>_HEARTBEAT_PUB.txt file. Review this file if you want to see the message included in the statusDesc property. An example message that is included in this file is as follows:

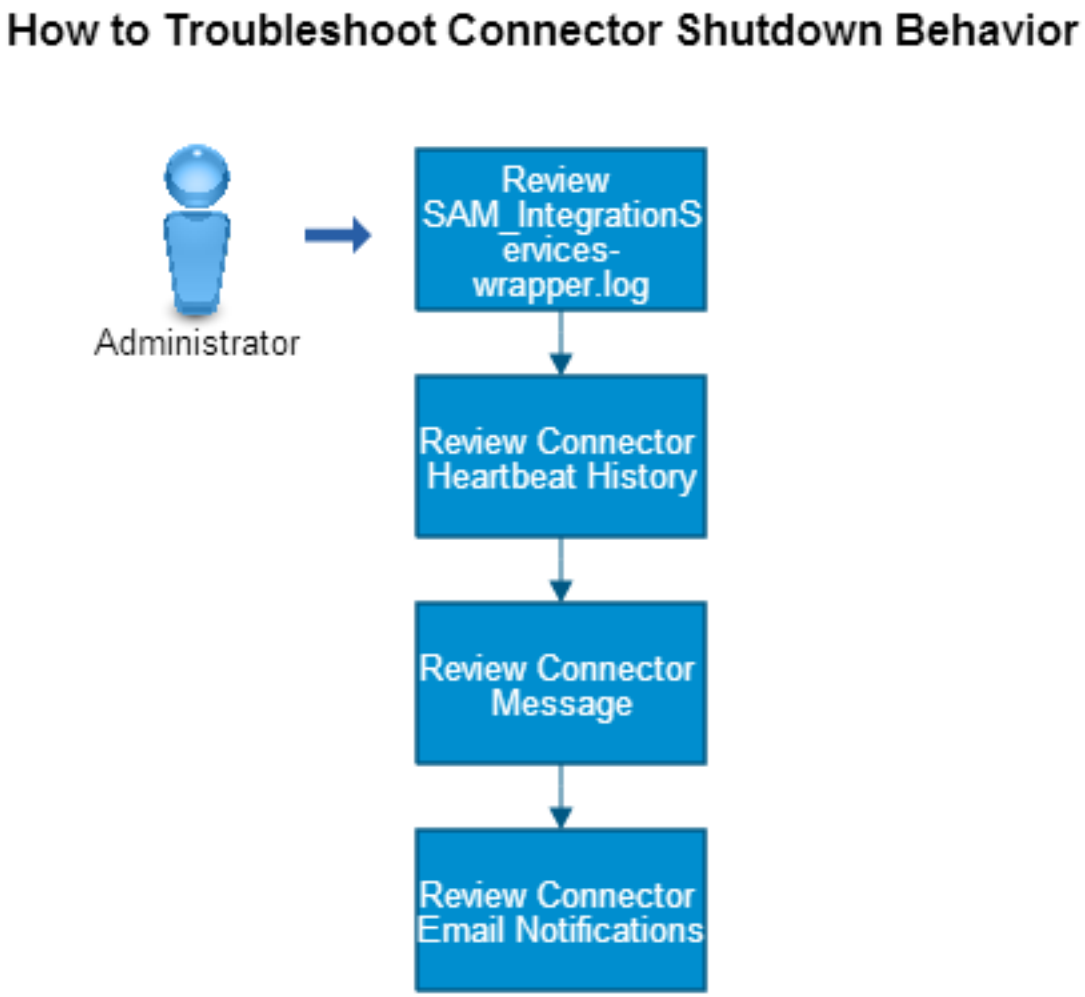
  ```
  statusDesc=Connector was stopped from Administration UI.
  ```

- Displays the reason for the connector shutdown in the Administration UI and Console.
- Sends an email notification to the CA SOI administrator about the connector shutdown.

For example, consider a scenario where a connector is stopped from the Administration UI. In this case, an appropriate message about the connector status is displayed at various locations. You can see the message in the product UI, SAM-IntegrationServices_wrapper.log file, <ConnectorName>_HEARTBEAT_PUB.txt file (if enabled), and in the email that is sent to the administrator (if enabled). By reading the message, you can quickly identify the reason for the connector shutdown and take appropriate measures.

Use this scenario to guide you through the process:

Figure 58: how to troubleshoot connector shutdown



1. Review the SAM-IntegrationServices_wrapper.log File.
2. Review the Heartbeat History for the Connector.
3. Review the Connector Message in the UI.
4. Review the Email Notification for the Connector.

Review the SAM-IntegrationServices_wrapper.log File

When a connector shuts down, CA SOI immediately logs the appropriate reason for the connector shutdown in the SAM-IntegrationServices_wrapper.log file. You can review and analyze the reason and perform the required steps to fix the issue.

Follow these steps:

1. Navigate to the <SOI_HOME>\jsw\logs folder.

   NOTE
   <SOI_HOME> represents the location where CA SOI is installed; for example, C:\Program Files\CA\SOI.

2. Locate and open the SAM-IntegrationServices_wrapper.log file.
   The log file opens in a text editor.
3. Search for the connector status message.
   The following example shows a connector status message that is logged to this file:
   ```
   2012/11/26 01:33:57 : abc-vm02.xy.com : CA:00056_service-discovery@abc-vm02.xy.com : OFFLINE - Connector
    was stopped from Administration UI. : Broadcasting HeartBeat Message
   ```
4. Review the reason for the shutdown.

Review the Heartbeat History for the Connector

CA SOI includes the shutdown message in the connector status notification as part of the heartbeat message that is periodically sent to the SA Manager. By reviewing the heartbeat history for the connector, you can determine the trend in the connector behavior.

The statusDesc property value in the connector debugData logs describes the status of the connector for which the heartbeat is sent. Enable the debugData logging for the connector (enabled in the log4j.xml file). After you enable the debug logging, a file <ConnectorName>_HEARTBEAT_PUB.txt is created and displays the heartbeat history for the connector. The connector writes any published heartbeat messages into this text file.

Follow these steps:

1. Navigate to the <SOI_HOME>\resources folder.
2. Locate and open the log4j.xml file in a text editor.
   The file opens in a text editor.
3. Modify the *com.ca.sam.ifw* level to DEBUG.
   The debugData logging is enabled for the connector.
4. Verify that the <ConnectorName>_HEARTBEAT_PUB.txt file is created under the <SOI_HOME>\log\debugData folder when the connector writes any published heartbeat message.
5. Open the file in a text editor and start reviewing the heartbeat messages in the file.
   An example of how the statusDesc property is displayed in the file is as follows:
   ```
   statusDesc=Connector was stopped from Administration UI.
   ```

Review the Connector Message in the UI

The SA Manager parses and stores the status description property value to display it to the user in the Administration UI and Console. The SA Manager receives this value as part of the heartbeat message that the IFW periodically sends to the SA Manager. When the connector goes offline for any reason, a relevant message is displayed in the Administration UI and Console as appropriate. This message includes detailed connector status and reasons for the shutdown (if any).

The Dashboard Administration tab provides a field named Status Description that provides detailed connector status information. This information is identical to the most recent entry in the Operation Console Connection Status dialog Message field.

To fix the issue, review and analyze the information that is provided in the message and perform relevant tasks.

Review the Email Notification for the Connector

When the connector goes offline, CA SOI notifies the specified administrator by sending an email. The email notification includes information about the failure.

To fix the issue, review and analyze the information that is provided in the email and perform relevant tasks.

An example of an email notification that is sent to the administrator when you stop any connector from the Administration UI is as follows:

```
SOI Manager - Connector 'build.abc.com:CA:09998_build.abc.com@build.abc.com' on SOI Manager build.abc.com is
 off-line and has not been restarted. The reported reason is 'Connector was stopped from Administration UI'
```

NOTE
If the administrator email is not configured in the Administration UI, specify the appropriate email ID to receive failure notifications by email. For more information, see Configure the CA SOI Administrator Email for Error Notification.

Configure the CA SOI Administrator Email for Error Notification

NOTE
You can configure the CA SOI administrator email in the Administration UI so that CA SOI can notify the administrator by email about failures.

Follow these steps:

1. Click the Administration tab.
2. Click the plus sign (+) next to CA SOI Insight Manager Configuration.
3. Click the plus sign (+) next to the server you want to configure.
4. Click Error Notification Configuration.
5. Enter the full email address for the administrator in the Administrator Email section.
6. For the Connector Failure Notification Settings section, perform the following actions:
   a. Select Yes from the drop-down to turn on email notifications.
   b. Enter the number of minutes until CA SOI sends the email notification.
7. Click Save.

Improved Connector Lifecycle Management

CA SOI provides an improved connector lifecycle management. The improved process enables you to reduce the time that you spend on diagnosing any connector-related connection issues. For example, when the domain manager of a connector goes offline, the IFW no longer stops trying to start the connector even after the completion of the configured retry count. The connector goes into an extended retry mode, if enabled. And, whenever the domain manager comes online, the connector establishes the connection and starts the processing, thereby reducing the downtime duration. Additionally, an email notification about the connection status is also sent to the administrator so that the administrator is aware of the issue and can take appropriate steps. This mechanism helps you proactively manage unexpected connection conditions and take correct and quick decisions.

NOTE
This feature is only available for connectors that are installed and run in CA SOI Integration Service. The lifecycle management for CA Catalyst r3.x connectors that run in CA Catalyst is outside the control of CA SOI, and is handled by CA Catalyst through connector configuration.

CA SOI has improved the connector reconnection process as follows:

- A new connector control *retryUntilMdrAvailable* is now available. This connector control specifies whether a connector should retry indefinitely to connect to its domain manager after the completion of the already configured connection retry count. You can configure this connector control to enable or disable the extended retries. To add this configuration, in the connector configuration file (<SOI_HOME>\resources\Configuration\<ConnectorName>.xml), add the following property to the <ConnectorControls> element:

  ```
  retryUntilMdrAvailable="1"
  ```

  NOTE
  For more information about how to configure a connector control, see Connector Configuration Tasks.

When the configured retries to connect to the domain manager are exhausted, the retryUntilMdrAvailable connector control determines whether the connector should continue its retries. If the value of the control is set to 1 (enabled), the connector goes into an extended retry mode. In this mode, the connector tries to connect to its domain manager periodically until that connection is successful, or the user manually shuts down the connector.

> **NOTE**
> The retry period for this mode is specified in the <SOI_HOME>\resources\Configurations \SSA_IFW_*HostName*.xml configuration file using the following property in the <ConnectorConfig> element:

```
extendedWaitBeforeRetry="<seconds>"  - the default is 600 seconds
```

If the value of retryUntilMdrAvailable is set to 0 (disabled), the connector does not go into the extended retry mode. The connector does not try infinitely to connect to the domain manager after the configured retry count is exhausted. If retryUntilMdrAvailable is not defined within the connector configuration, it will default to 1 (enabled). It is only necessary to add the control parameter to disable this feature.

- If the extended retry mode is enabled, the IFW sets the connector status as OFFLINE (RETRYING) after the IFW is unable to start the connector within the configured retry count. The IFW also sends a handshake message to the SA Manager. The SA Manager reads and stores the new status. The message includes detailed information that the configured retry count is exhausted, the IFW is unable to start the connector within the specified retry count, and it is marking it as OFFLINE (RETRYING). The SA Manager also sends an email notification that includes the handshake message information (received from the IFW) to the administrator.

  > **NOTE**
  > For more information about how to configure email notifications, see Configure Email and Failure Notifications.

- When the connector is in the extended retry mode, the status of the connector is displayed as OFFLINE (RETRYING) in the Administration tab and Connection dialog. In this state, the Stop button is available from the Administration tab. Users can use this button to stop the extended retries.
- Each retry logs a message in the Info dialog on the Console, indicating that the IFW is attempting to restart the connector. The message is also displayed in the Administration tab.

No Connector Data in the Operations Console

Symptom:

My connector is ONLINE, but I do not see any corresponding data in the Operations Console.

Solution:

Because data is retrieved and sent through multiple checkpoints, it is necessary to trace it from the domain manager to connector to CA SOI. However, there are a few simple diagnostics that you can try before getting into the details:

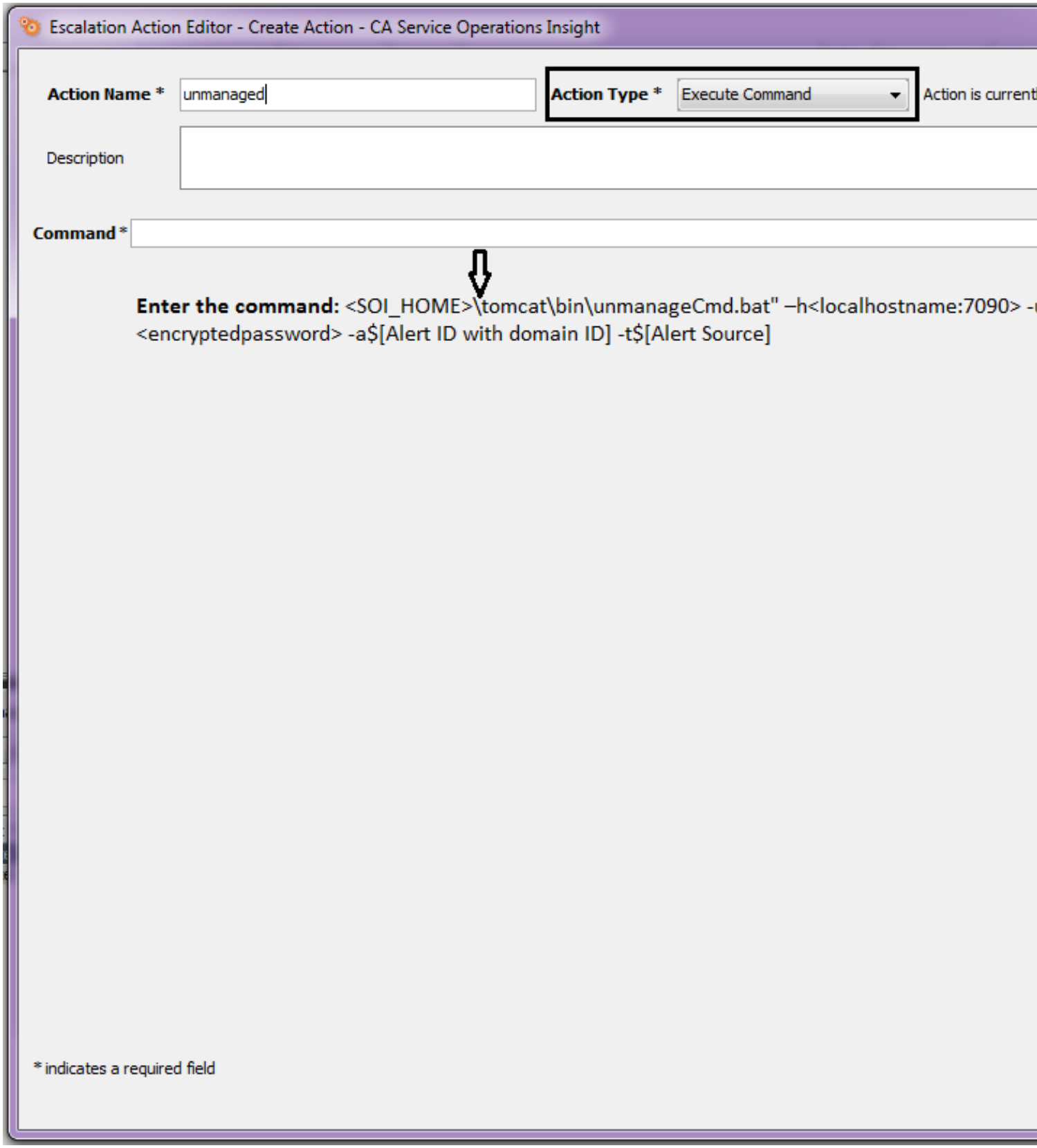
- If you believe the connector published alerts, check the Alert Queues tab in the Operations Console. CIs do not have to be modeled into a service before their corresponding alerts populate the default alert queue.
- CIs that have not been modeled into a service, or imported as part of a service, do not appear in the Operations Console. From the Modeler, use the Browse By, Data Source option to see if there are any CIs that are imported from your connector.
- To determine if any data is associated with your connector in the database, use the CA SOI toolbox.
- If no data is associated with your connector in the database, it is necessary to trace the data.

How to Convert Unmanaged CIs to Managed CIs

To publish the CIs (managed and unmanaged) that are associated with CIs in Enterprise SA Manager, follow these steps:

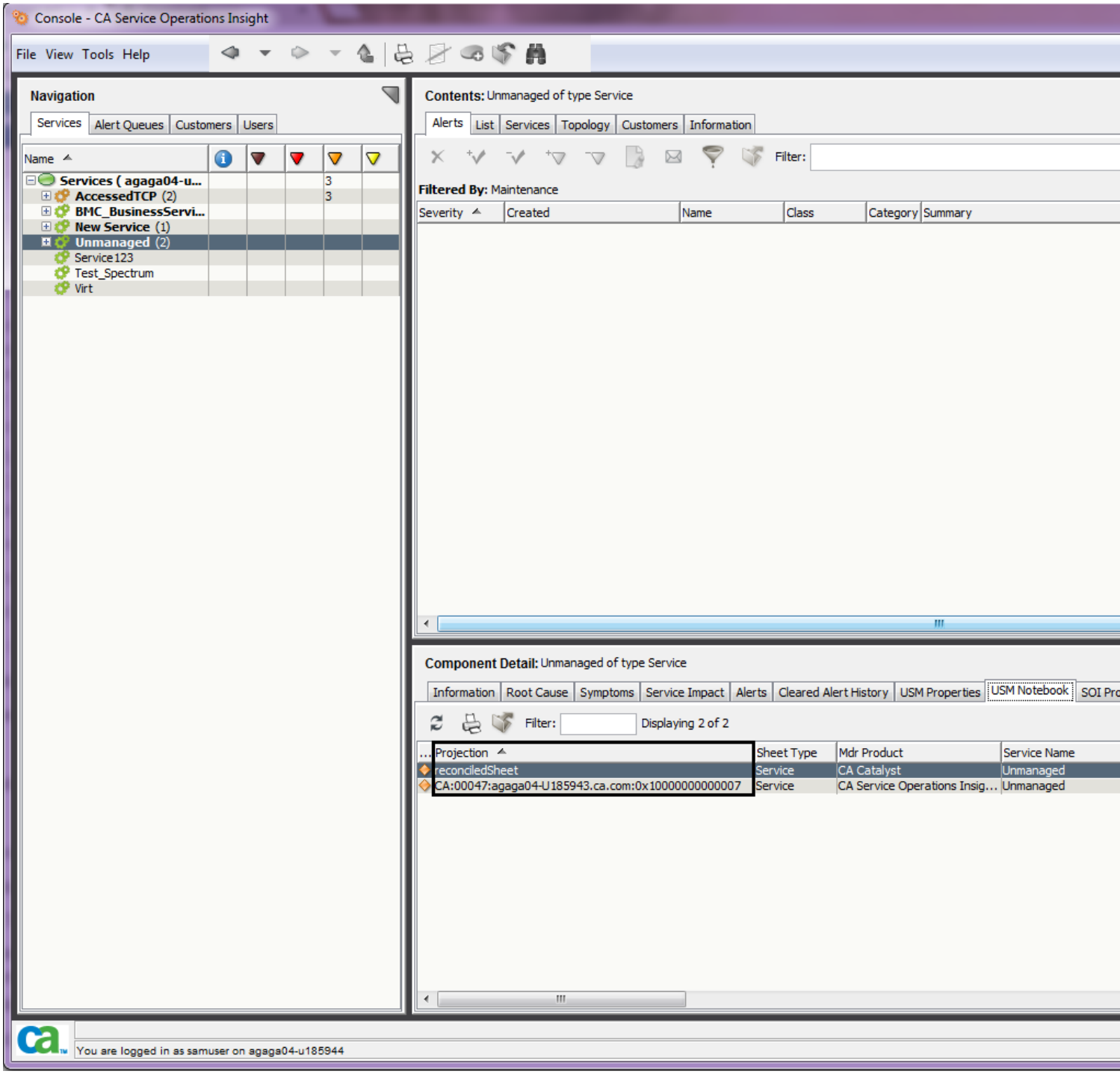
1. Unzip **T6CP075_4.0.2.132_12_21_2016.zip** folder on the Domain SA Manager

2. The zip folder contains the following files:
   a. **Utility** folder - contains ServiceResource.properties, unmanageCmd.bat, unmanage.jar files.
   b. **T6CP075.exe**
3. Run **T6CP075.exe file** and install the patch.
4. Copy the **ServiceResource.properties**, **unmanageCmd.bat**, **unmanage.jar** files from the Utility folder to the **<SOI_Home>\tomcat\bin** folder.
5. Start the Operation Console, and create an **Escalation Policy and Action**.
   For more information about creating escalation policy and action, see Create Escalation Policy and Action.
6. In **Actions** tab, select the action type as **Execute Command**.
7. Enter the following command in the **Command*** field.
   <**SOI_HOME**>\tomcat\bin\unmanageCmd.bat" –h<localhostname:7090> -u<username> -p<encryptedpassword> -a $[Alert ID with domain ID] -t$[Alert Source]
   Where,
   – **-h** defines the hostname where SA Manager is installed
   – **-u** defines the username
   – **-p** defines the password in encrypted format
   – **-a** defines the alert id with the domain id. Right click in the **Command*** field and select $[Alert ID with domain ID].
   – **-t** defines the alert source. Right click in the **Command*** field and select $[Alert Source].
   For example, "c:\Program Files (x86)\CA\SOI\tomcat\bin\unmanageCmd.bat" -habcd02-U178375:7090 -usamuser -pEJC6HRY79+PZirsnrwXJgpG6W0N3emgFQMVCfhvIbuy8 -a$[Alert ID with domain ID] -t$[Alert Source].



8. Click **OK** to save the changes.

9. Click **Services** tab, and select a service.
10. In **Component Details** pane, click **USM Notebook** tab and note the **Mdr Element ID** for reconciledSheet and CA:00047 projection.



11. Open **ServiceResource.properties** file, and enter the **Mdr Element ID** values that you have noted in step 8.

NOTE

- For projection CA:00047, update the same Mdr Element ID (noted in step 7) in service.soi.project.mdr.element.id and service.soi.source.mdr.element.id.
- For projection reconciledSheet, update the same Mdr Element ID (noted in step 7) in service.scope.mdr.element.id.

For example,

– #MDR element ID of Service in which CI needs to Added(USM Notebook) in Projection CA:00047
service.soi.project.mdr.element.id=0x1000000000000f
– #MDR element ID of Service in which CI needs to Added(USM Notebook) in Projection CA:00047
service.soi.source.mdr.element.id=0x1000000000000f
– #MDR element ID of Service in which CI needs to Added(USM Property) in short reconciled sheet ID
service.source.mdr.element.id=57d07a55be394314a13e60bfcdaa7735
– #MDR element ID of Service in which CI needs to Added(USM Property) in short reconciled sheet ID
service.scope.mdr.element.id=57d07a55be394314a13e60bfcdaa7735

12. Save the file.
Based on the alerts that meet the Escalation Policy criteria, CIs are added to the service as mentioned in step 11. Hence, the Unmanaged CIs are converted to Managed CIs.

Sample Connector Changes Are Not Reflected in CA SOI

Symptom:

I made changes to the Sample connector sample data files; however, I do not see those changes in CA SOI.

Solution:

If you make changes to the Sample connector sample data files or connector policy and these changes are not reflected in the Sample connector data that CA SOI displays, check the location of the files you manipulated.

Sample data and policy files for the Sample connector exist in the following places:

- **<SOI_HOME>\resources**
Contains data and policy files that the Sample connector uses during runtime.
  – **<SOI_HOME>\resources\SampleConnector\data**
  Contains sample data files.
  – **<SOI_HOME>\resources\Core\Catalogpolicy**
  Contains connector policy files.
- **<SOI_HOME>\SampleConnector**
Contains the code that implements the Sample connector and the framework for building a custom connector.

If you want to make changes to the sample data or policy files and see the effect on a running Sample connector, make the changes in the <SOI_HOME>\resources\SampleConnector and <SOI_HOME>\resources\Core\Catalogpolicy directories and restart the CA SAM Integration Services service. Changing the files in the <SOI_HOME>\SampleConnector directory has no effect on a running Sample connector.

Unable to View CA Catalyst r3.2 Connectors in CA SOI r3.x

Symptom:

I cannot view my CA Catalyst r3.2 connectors in CA SOI r3.x.

Solution:

The IFW Proxy provides access to data from CA Catalyst r3.2 connectors to CA SOI. The IFW Proxy is embedded in the CA Catalyst r3.4.1 Container. For more information about working with the r3.4.1 Containers and integrating its installed connectors with CA SOI, see Catalyst Installation Guide Title Page.

Dashboard Troubleshooting

Administration Tab or Dashboard Links Not Working with Firefox

Symptom:

I am experiencing either or both of the following errors when accessing the Dashboard with Mozilla Firefox:

- I click the Administration tab, but the interface does not open.
- I click any of the Dashboard links (Google Earth, Console, and so on), but nothing happens.

Solution:

This problem can result from a conflict between some versions of Firefox (before v17.x and after v18.x) and the Java Deployment Toolkit plugin.

Follow these steps:

1. If you are running Firefox, verify that you are using a version later than 12 and upgrade to a later version if required.
2. If you are unable to upgrade to a later version of Firefox, disable the Java Deployment Toolkit plugin in Firefox. This plugin checks for Java updates.

Administration Tab Values Not Saving

Symptom:

When I update values on the Dashboard Administration tab, CA SOI does not save the values.

Solution:

This issue is related to the browser security. Add the UI server to your web browser trusted sites and allow cookies from the domain.

Follow these steps:

> **NOTE**
> The following example uses Internet Explorer. For a list of supporter browsers, see Web Browser Support.

1. Add the UI server to your Internet Explorer web browser Trusted Sites:
   a. Select Tools, Internet Options.
   b. Click the Security tab then click Sites.
   c. Click Advanced.
   d. Enter the fully qualified domain name of the UI server and click Add.
2. Allow cookies from the UI server:
   a. Select Tools, Internet Options.
   b. Click the Security tab then click Sites.
   c. Enter the domain name (for example, mydomain.com) of the UI server and click Allow.

Alerts Not Created

Symptom:

CA SOI is not creating alerts.

Solution:

1. The connection to the SA Store database may have failed. Configure the database connection failure email notifications. If there is a failure, the email provides the resolution.
2. See Alert Flow in CA SOI and Log File Outputs to trace alert problems.

CIs Not Created

Symptom:

CA SOI is not creating CIs.

Solution:

1. The connection to the SA Store database may have failed. Configure the database connection failure email notifications. If there is a failure, the email provides the failure details and a link to the resolution.
2. See CI flow in CA SOI and Log File Outputs to trace CI problems.

Failure Notification Emails Not Sending

Symptom:

I have activated the failure notifications, but the emails do not send.

Solution:

1. If you are sending the email to multiple addresses, verify that you are using commas (,) to separate the emails; semicolons (;) do not work.
2. Test the email server by manually sending an alert email:
   a. Log in to the Operations Console.
   b. Select an alert and click the envelope icon.
   c. Complete the email dialog and send the email.
3. Verify the correct mailhost is defined in the following location: \Windows\System32\drivers\etc\hosts file
4. Restart the CA SAM Application Server service.

Google Earth Does Not Display New or Updated Locations on the Map

Symptom:

I have set or updated a service location on the Operations Console, but Google Earth does not update the location on the map.

Solution:

There is a known issue with Google Earth 7.0. If you edit or modify the service location while Google Earth is open, the location shows immediately on the Google Earth Places list, but can take 20 to 30 minutes to update on the map. As a workaround, restart Google Earth and the updated locations appear immediately on the map.

Administration Tab Display Issues

Problem

When you view any page on the Administration tab, the page contents take up the entire screen, obscuring the tree view on the left.

Solution

This is a known problem when using Internet Explorer 11. Either disable IE enhanced security, or add the domain of your CA SOI Dashboard to the browser's Compatibility View settings.

Event Management Troubleshooting

Event Management Connection Problems

Symptom:

The Event Management components are failing or unable to integrate with certain data sources.

Solution:

Perform the following actions:

- Verify that the CA SAM Event Management Service is running on the SA Manager and connector systems.
- Verify that each connector appears as Online in the Operations Console and the Administration UI.
- To see an updated list of available connectors, refresh the Events tab.

Event Search Returns no Results or Unexpected Results

Symptom:

An event search returns no results or unexpected results.

Solution:

Verify the following items:

- Verify that you are using the correct properties and values that are listed in Event Properties and Event Information.
- If you are using optional properties in search patterns, verify that the appropriate data sources produce that property.
- Verify that the pattern syntax adheres to the conventions in Event Search Syntax Guidelines and Best Practices.
- Try using the 'ANY event occurs' criterion first, because it does not consider time intervals. Once you have established that events exist that match the patterns, try a time-based pattern detection.
- If you receive an error message before the results display that you need help interpreting, see the section on error messages.

Expected Alerts Not Appearing on Operations Console After Processing

Symptom:

Alerts are not appearing on the Operations Console after Event Management processing. For example, events do not appear as alerts.

Solution:

Perform the following actions:

- Verify your alert filter settings at the IFW level. The IFW or specific connectors may be preventing alerts that do not affect services from appearing.
- To verify that you are not filtering out alerts that you want to see, view the applied alert filters at the Operations Console level.

NOTE
For more information about alert filters, see View Alerts, Alert Details, and Extended Information.

- To verify that you are not prevented from seeing alerts that are appearing in specific queues, view the defined alert queues and their access privileges.
- All alerts must be associated with a valid CI.

Event Policies Not Producing Expected Actions

Symptom:

Deployed event policies are not producing the expected actions. For example, enrichments are not occurring.

Solution:

Perform the following actions:

- See the section about error messages for help interpreting any errors that appear while creating policy.
- Verify that a file for the policy exists in the <SOI_HOME>\resources\EventManagement\Policies directory. Print this file for a record of the raw policy syntax.
- Enable detailed transformation policy logging for a more granular view of connector and event policy operations. The <SOI_HOME>\log\eitransform.log file includes operations that even policies add. This log file helps you find out whether the information is processed correctly. Enable detailed transformation logging in the eitransform.log file before you can start using it for troubleshooting your event policy. You can track individual operations that occur based on your event policies. For example, you can see whether trap properties are being normalized properly.
- If you changed the enrichment connection information (for example, edited script parameters), restart the CA SAM Integration Services service.
- Policies that you create through the user interface do not support aspects of certain actions, such as combining include and exclude filters. If the user interface does not provide the functionality that you require, consider refining your policy manually.

How Do I Control the Event Management Data Flow to the Operations Console?

Symptom:

I want to control the Event Management data flow to the Operations Console so that I can manage my events more efficiently. How can I do that?

Solution:

To decide how you want to control the Event Management data flow to the Operations Console, configure the eventManagerClientConfig.xml configuration file.

Follow these steps:

1. Open the <SOI_HOME>\tomcat\lib folder.
2. Locate and right-click the eventManagerClientConfig.xml file, and select Open With, Notepad from the context menu.
3. Configure the following parameters, and save and close the file:
   - **timeoutValues**
     Specifies the amount of time in seconds that the event service waits for a response to a query request. Each type of request can have its own value. You can set the timeout values for the following actions:

  - DeployPolicy
  - DeployScript
  - GetConnectorInfo
  - GetDeployedPolicy
  - GetDeployedScript
  - GetEvents
- **synchInterval**
  Specifies the polling interval in seconds for determining the available event services. The list of data sources available in the Event Policy dialog (Tools, Event Policies) in the Operations Console reflects the current state of this polled information. Any changes to the status of data sources can take up to the interval time (specified for this parameter) to update in the Operations Console.
  Default: 45

4. Restart the CA SAM Application Server service.

Searches Taking Too Long

Symptom:

The event searches are taking too long, or the data source information is incorrect.

Solution:

Edit the timeout values in the eventManagerClientConfig.xml file.

Error Messages on an Event Result Error Dialog

Symptom:

I received an error message on an Event Result Error dialog when I ran an event search. How do I interpret the error message?

Solution:

The following messages can appear on an Event Result Error dialog when you run an event search and click a result button that has turned yellow or red:

- ***Connector name*: No events matched**
  Indicates that the search returned no matches for a connector that you included in the scope. After you close the dialog, events may appear for other connectors included in the scope. This error does not indicate a problem with the search itself or the returned results. The error is only a notification that at least one scoped source returned no results. In response to this message, you can either refine your search if you want to return events from that source, or do nothing and simply work with the returned results if the search is accurately scoped and defined.
- ***Connector name*: Warning:droolsconvert_failed**
  Indicates that some portion of the search syntax cannot be converted to the Drools language. The actual Drools conversion takes place when you save or deploy an event policy. Therefore, this warning does not affect the accuracy of a simple event search, but creating an event policy based on the search will fail. Common reasons for Drools conversion failure include the use of unsupported functions or inappropriate use of operators (for example, a greater than or less than operator with non-numeric values). For specific information about the error, see the EventMgmt.log file at <SOI_HOME>\log.
  For more information about constructing valid searches, see Event Search Syntax Guidelines and Best Practices.
- ***Connector name:* Connector is not available for request**
  Indicates that Event Management could not access the connector to query its events. Check the status of each connector if this error occurs.
- ***Connector name:* Request timed out. Event Service did not respond within 30 seconds.**

Indicates that the search did not complete due to an unresponsive Event Service. Check the status of the CA SAM Event Management service if this error occurs.

- **Large event set matched. Reduce the scope.**
  Indicates that the search returned more than 25,000 events for a single data source, which is the upper limit for search results per connector. You must reduce the scope, either through time range or data sources, to return an acceptable number of events.
- **An Internal error occurred. Please check server logs**
  Indicates that an unspecified error occurred that prevented the search from completing. To investigate the source of the error, see the EventMgmt.log file at <SOI_HOME>\log.
- **Policy file not found**
  Indicates that a saved or deployed policy that you selected on the Events tab is not available in its expected location, and its pattern does not display in the Event Search tab. Verify that the policy file exists on the SA Manager at <SOI_HOME>\resources\EventManagement\Policies.

The following messages may appear when you click Create Policy or Map Events on the Event Search tab:

> **NOTE**
> Some of these error messages also appear when you click the result button.

- **Map Events needs search results**
  Indicates that no current search results exist for the entered raw event search pattern. Completing a raw event search is required before creating a normalization action based on that search, so that you can use the results to access the raw event properties for mapping. The message gives you the option to continue, but no raw event properties are available to map on the Normalize Event page.
- **Error: Unable to Resolve: property='value**
  Indicates that the event search is invalid due to a missing quotation mark on either side of the property value. Add the missing quotation mark and rerun the search.
- **Error: OR operands NOT supported in policy deployment for Raw Events**
  Indicates that the raw event search uses an OR operand. The search returns valid results, but event policies that are based on the raw event searches do not support the use of this operand.
- **Error: Operator: NOT supported in policy deployments for Raw Events**
  Indicates that the raw event search uses an operator that is not supported in event policies that are based on the raw event searches. Only the '=' operator is supported in this situation.
- **Error: Operator: NOT supported in policy deployments for Normal Events**
  Indicates that the normalized event search uses an operator that is not supported in event policies that are based on the raw event searches. Only the '=' and != operators are supported in this situation.
- **Error: not syntax incorrect. Not supported in policy deployments pattern**
  Indicates that the event search uses unsupported syntax. For details, see Event Search Syntax Guidelines and Best Practices.
- **Error: contains / starts-with / ends-with is NOT supported in policy deployments pattern**
  Indicates that the event search uses a function that is not supported in event policies.

The following messages can appear on the Create Event Policy dialog when you try to deploy or save an event policy:

- **Search errors**
  Indicates that the search errors previously listed can appear in the Event Log table that displays the current search results for use in previewing how a create event or enrich event action affects an event.
- **UNKNOWN_ERROR - [QueryParms.ConvertToElPolicy: Drools conversion failed, see log files]**
  Indicates that some portion of the search syntax cannot be converted to the Drools language. This message appears after you click Finish on the Select Data Sources page. The Drools conversion takes place when you save or deploy an event policy, and the operation is prevented if Drools conversion fails. Common reasons for Drools conversion failure include the use of unsupported functions or inappropriate use of operators (for example, a greater than or less than operator with non-numeric values). For specific information about the error, see the <SOI_HOME>\tomcat\logs \soimgr.log file.

For more information about constructing valid searches, see Event Search Syntax Guidelines and Best Practices

- **One or more Data Sources are currently disabled!**
  Indicates that one or more connectors are currently unreachable. This message appears on the Select Data Sources page. You can complete the policy creation if all of the data sources that you need are available.

Various other warnings appear at the bottom of the Create Event Policy dialog when input is required before you can progress to the next page.

How to Search Archived Event Store Files?

Symptom:

I want to search archived the Event Store files for event searches that are not time-scoped.

Solution:

To search archived the Event Store files, configure the EventManager-wrapper.conf file.

Follow these steps:

1. Open the <SOI_HOME>\jsw\conf\EventManager-wrapper.conf file.
2. Add the following Java define as necessary or change its default value in the Java Additional Parameters section, and save and close the file:
   - **EQUERY_UNZIP_ARCHIVE=**
     Determines whether to unzip and search the archived Event Store files for event searches that are not time-scoped. Enter 1 to search archived files. The value 0 implies that you do not want to search archived files.
     Default: 0

   Preface the parameter with the correct additional Java parameter syntax and the appropriate sequential number, as shown in the following example:
   ```
   wrapper.java.additional.2=-DEQUERY_UNZIP_ARCHIVE=1
   ```
3. Restart the CA SAM Event Management service.
   Subsequent event searches use the configured settings.

Not Enough Event Groups in Search Results

Symptom:

I want to increase the number of event groups in search results so that more groups are returned in results.

Solution:

To create the maximum number of event groups in search results, configure the EventManager-wrapper.conf file.

Follow these steps:

1. Open the <SOI_HOME>\jsw\conf\EventManager-wrapper.conf file.
2. Add the following Java define as necessary or change its default value in the Java Additional Parameters section, and save and close the file:
   - **ESTORE_MAX_QGROUP=**
     Determines the maximum number of groups to return in search results.
     Default: 10

   Preface the parameter with the correct additional Java parameter syntax and the appropriate sequential number, as shown in the following example:
   ```
   wrapper.java.additional.3=-DESTORE_MAX_QGROUP=10
   ```
3. Restart the CA SAM Event Management service.
   Subsequent event searches use the configured settings.

Event Processing Performance Due to Mid-Tier Connector

Symptom:

I am experiencing some event processing performance issues due to the Mid-Tier connector. The Mid-Tier connector is not required in my infrastructure for event processing. How can I disable it?

Solution:

You can disable the Mid-Tier connector if the holistic action processing layer that it provides is not necessary for your Event Management implementation. Bypassing the Mid-Tier connector if you are not using it for event policies improves the event processing performance. The setMTCstate utility disables the connector and reroutes events so that they proceed directly to the SA Manager from connectors.

Run this utility on the SA Manager system. The utility assumes the default Mid-Tier connector configuration, with one Mid-Tier connector that is installed on the SA Manager.

Follow these steps:

1. Shut down the Mid-Tier connector using the CA SOI Administration tab.
2. Navigate to <SOI_HOME>\Tools on the SA Manager and run the following command:
   ```
   setMTCstate Disabled
   ```
   The Mid-Tier connector is disabled.

   NOTE
   You can also enable the Mid-Tier connector if you have previously disabled it by substituting the term Enabled for Disabled.
3. Restart the CA SAM Application Server service.
   The change is applied and the connector is disabled.

Help Desk Integrations Troubleshooting

CA Service Desk Integration Troubleshooting

If you cannot connect to CA Service Desk or create CA Service Desk tickets in CA SOI, do the following to troubleshoot CA Service Desk connection problems:

- Verify that the connection settings are correct on the Help Desk Configuration page of the Administration UI. Click Test to test the connection.
- If CA Service Desk is configured for SSL, verify that you selected the SSL check box. The integration does not work if this check box is not synchronized with the CA Service Desk SSL configuration.
  For more information, see Export CA Service Desk SSL Certificate.
- Try to access the following CA Service Desk URL independent of CA SOI:
  ```
  http://<ServiceDeskServer>:<ServiceDeskPort>/axis/services/
  USD_R11_WebService?wsdl
  ```

  NOTE
  The default CA Service Desk port is 8080.

  If you cannot reach this URL, do the following:
  - Verify that the Service Desk Web Services component is installed on the CA Service Desk system. This component must be installed for the integration to work.
  - See the CA Service Desk documentation.
- If you are using CA Service Desk r12.1 that has been upgraded from a previous release, clear the CA Service Desk browser cache.

CA Service Desk Ticket Not Created by Escalation Policy Action

Symptom:

I have configured an escalation policy action to create a CA Service Desk help desk ticket, but CA SOI does not create the ticket.

Solution:

Follow these steps:

1. Try to create the ticket manually. If the ticket fails, note the error message that is returned. For more information about creating a ticket manually for CA Service Desk, see How to Work with Configured Help Desk Integrations.
2. Review the CA SOI escalation policy action to verify that any CA Service Desk specific values such as the template, requested area, and assignee are valid values in CA Service Desk.
3. Verify that there is a Description property with a value set. Certain versions of CA Service Desk does not work without a Description property.

Cannot Create a BMC Remedy or HP Service Manager Ticket in CA SOI

Symptom:

I cannot create a BMC Remedy or HP Service Manager Ticket in CA SOI.

Solution:

Follow these steps:

1. Verify the connection to CA Process Automation in the CA SOI Administration tab Help Desk Configuration screen. For more information about configuration, see CA Process Automation Integration.
2. Verify the connection to BMC Remedy or HP Service Manager using the CA Process Automation TestRemedyServerConnection (BMC Remedy) Form or the TestHPSMServerConnection (HP Server Manager) Form.
3. Create a test ticket in BMC Remedy or HP Service Manager using the CreateRemedyTicket (BMC Remedy) Form or the CreateTestTicket (HP Service Manager) Form. For more information about creating a test ticket, see How to Work with Configured Help Desk Integrations.

HP Service Manager Integration Troubleshooting

Use the following information to troubleshoot issues with the HP Service Manager integration:

Triggers are not initiated when a user in HP Service Manager edits an incident that the integration created.

- Verify that the us.launch.external.NEW application is installed on the HP Service Manager server. If the application is not installed, download it from HP Support.
- The launch in context URL that opens when you click a Ticket ID in CA SOI does not show the incident details in HP Service Manager.
- Verify that the following parameters in the web.xml file of the HPSM Web Client have the following values:

Property Name	Value
querySecurity	false
useservertabs	true
essuser	false

Ticket Creation, Closure, or Update is Failing

Symptom:

Any or all of the following help desk ticket problems occur:

- I created a ticket in CA SOI, but the ticket does not appear in the help desk product.
- I cleared an alert in CA SOI, but the ticket does not close in the help desk product.
- I updated the alert in CA SOI, but the ticket does not update in the help desk product.

Solution:

Use the following process to resolve the problem:

1. Review the escalation policies and actions. Verify that the policies are valid.
2. There could be a connection problem to the CA Process Automation server or the help desk server. The failure results in escalation actions for help desks failing or the synchronization of certain CA SOI properties with the help desk ticket failing. Configure email notifications for third-party server failures to determine if this is the cause. The email provides failure information.
3. In CA Service Desk, look in the Action History table for an error message that is related to the ticket.
4. In CA Process Automation, look for workflow error messages for BMC Remedy or HP Service Manager tickets.

Ticket Status Changed when Connector Shut Down or Removed

Symptom:

When I shut down or remove a connector, CA SOI changes the status of tickets to Closed or another status.

Solution:

In the Help Desk Configuration dialog (Operations Console menu: Tools, Help Desk Configuration), you set the option "Auto change trouble ticket status when alert is cleared." When you shut down or remove the connector, CA SOI changes to the status you specified for all cleared alerts or Closed, by default. You cannot undo this operation.

Integration Framework Troubleshooting

Java Heap Space Out Of Memory Error

Symptom:

When I try to import a large number of CIs from a domain manager, I receive the Java heap space OutOfMemory error.

Solution:

If there are a large number of CIs, relationships, and services to be imported from a domain manager, the IFW may run out of allocated JVM heap space and stop responding. This error appears as a "java.lang.OutOfMemoryError:Java heap space" exception in the <SOI_HOME>\jsw\logs\SAM-IntegrationServices_wrapper.log file. You may need to increase the IFW JVM heap space if this situation occurs frequently to process all imported CIs and relationships.

Do any of the following tasks to manage the IFW JVM heap space:

- Increase the JVM heap size in the IFW (default is 1 GB) as follows:

  - Stop the CA SAM Integration Services service.
  - Increase the wrapper.java.maxmemory value (in MB) in the <SOI_HOME>\jsw\conf\SAM-IntegrationServices.conf file.
  - Start the CA SAM Integration Services service.
- Turn off the getRelationshipsAtStartup control for connectors with a large number of CIs and relationships. This operation prevents rediscovery of all relationships every time the connector starts. Importing services still imports all relationships with this control turned off.

Unable to Start the IFW Services

Symptom:

I am unable to start the IFW services, due to which the connectors are not able to send the data to CA SOI.

Solution:

Start the CA SOI Services in the following order:

1. CA SAM Application Server
   You must wait until the CA SAM Application Server is started. To verify the service status, navigate to the CA SOI User Server Debug Console, and click **Triage Tests**.
2. CA UCF Broker
3. CA SAM Event Management
4. CA SAM Integration Services
5. CA SAM Store Indexer
6. CA SAM User Interface Server

Mobile Dashboard Troubleshooting

Service Names do not Appear on the Mobile Dashboard

Symptom:

When I access the Mobile Dashboard, the service names are empty.

Solution:

Service names on the Mobile Dashboard are obtained from the USM Properties that are named Service Name and Label.

It is possible that the USM properties were not created properly.

Follow these steps:

1. Stop the CA SAM Application Server service.
2. Navigate to the following folder:
   <SOI_HOME>\tomcat\registry\topology\physical\node0\sor
3. View the following files and verify it matches the server name with the output of %COMPUTERNAME% case sensitive.
   - restserver.xml
   - sorapp.xml
   - ssaserver.xml

   Example:
   ```
   <tns:brokerURL>tcp://<Computername>:61616</tns:brokerURL>
   <Computername> has to be the exact content of %COMPUTERNAME%
   ```
4. Open a cmd window, navigate to the SOI_HOME\tomcat\registry folder and run the command "registryloader".

NOTE
Ignore any warning messages about log4j settings.

5. Perform one of the following actions:
   - For new created Services verify that the USM Properties "Service Name" and "Label" are correctly populated.
   - For existing Services run the Primerutility script in <SOI_HOME>\Tools\Priming Utility\ which creates the missing reconciled sheets.

Operations Console Troubleshooting

Access - Proxy Server Prompt Opens When Accessing the Operations Console

Symptom:

Every time I open the Operations Console, I must enter credentials in a proxy server prompt.

Solution:

You must configure your Java settings to use your browser settings or establish a direct connection when starting Java applications.

Follow these steps:

1. Launch the Java Control Panel.
2. Click Network Settings.
3. Select 'Use browser settings' or 'Direct connection', and click OK.
4. Click OK on the Java Control Panel.

Access - Operations Console Link Disabled

Symptom:

The Operations Console is inaccessible, because the Console link on the Dashboard is disabled.

Solution:

Ensure that your browser and Java version are compatible. For example, if your browser is 32-bit and your Java installation is 64-bit, the Console link may not be available. Install the correct version of Java to match your browser to fix the issue.

Access - Unable to Start the Operations Console

Symptom:

When I try to open the Operations Console, the message "Unable to Launch Application" appears.

Solution:

Verify that you are using a Java version of 1.7.0_55 or above; if not, upgrade to a supported version. If the automatic installation does not work, manually download and install the latest JRE from the Java website and try to launch the Operations Console again.

If you are using a supported Java version, try clearing the JNLP cache.

Follow these steps:

1. Launch the Java Control Panel.

2. Click the View button in the Temporary Internet Files section.
3. Right-click the extra CA SOI applications in the list, and select Delete.

Alerts - How Do I Find an Alert?

Symptom:

I want to track the flow of alerts in CA SOI to pinpoint the error. How can I trace an alert in CA SOI?

Solution:

Review the Alert Flow in CA SOI and Log File Outputs section in the guide. This section provides the complete flow of an alert. You can review the stages involved in the flow and the corresponding log file outputs to identify, analyze, and fix the error.

CIs - Domain Manager Administrative CI States not Reflected in CA SOI

Symptom:

I do not see the administrative CI state for my domain manager in CA SOI.

Solution:

Some domain managers are capable of setting administrative CI states. The states that are set in domain managers are not reflected in CA SOI. The corresponding CI displays a Normal status in CA SOI.

CIs - Duplicates Appear on the Operations Console

Symptom:

I see that duplicate CIs are appearing on the Operations Console.

Solution:

If duplicate CIs appear on the Operations Console that should have been correlated as one CI, check the DNS resolution settings in the source domain managers.

CIs - How Do I Find a CI?

Symptom:

I want to track the flow of CIs in CA SOI to locate the error. How can I trace a CI in CA SOI?

Solution:

Review the CI Flow in CA SOI and Log File Outputs section in the guide. This section provides the complete flow of a CI. You can review the stages involved in the flow and the corresponding log file outputs to identify, analyze, and fix the error.

CIs - Missing from CA SOI

Symptom:

I do not see CIs in CA SOI.

Solution:

If you do not see expected CIs in the CA SOI Operations Console or the staging area of the Service Modeler, they may have been rejected by a USM validation failure. The SA Manager validates CIs against the USM schema, and any CI that does not pass this validation is rejected and filtered out. A CI may fail USM validation for any of the following reasons:

- An error in connector policy that improperly converts the domain manager information to USM
- No data in a field that is required by USM
- The domain manager presents data in a way that cannot be processed by its connector policy

Check the <SOI_HOME>\log\ssa.log file to see if USM schema validation failures have occurred.

CIs - Property Values are Incorrect

Symptom:

I notice that some of the CI property values are incorrect. How do I correct them?

Solution:

If you notice that CI property values are incorrect (such as IP address, DNS name, and DeviceID), ensure that you adhere to the following best practices for DNS resolution and fixing incorrect values:

- If your environment uses DHCP, you must have DNS resolution enabled for all connectors.
- The IP address, SysName, and DNS name (when used) must be correct for managed objects in the domain manager. If necessary, rerun the discovery on the domain manager to correct these properties.
- If the IP address is incorrect in the domain manager, do not turn off DNS resolution on the connector. This approach makes the problem more severe.

Escalation Actions - CA Process Automation Forms Not Available

Symptom:

I am trying to create an escalation policy using the Execute Automated Process action type, but no CA Process Automation forms are available.

Solution:

CA SOI may have lost the connection to the CA Process Automation server.

Follow these steps:

1. Enable server connection error notifications on the Dashboard Administration tab.
2. Test the CA Process Automation server.
3. Restart the CA SAM Application Server service.

Escalation Actions - Email Actions Not Sending

Symptom:

I have created an escalation policy or manually used Take Action to send an email, but CA SOI does not send the email.

Solution:

You may have entered multiple email addresses incorrectly or CA SOI may have lost the connection to the email server.

Follow these steps:

1. If you are sending the email to multiple addresses, verify that you are using commas (,) to separate the emails; semicolons (;) do not work.

   Test the email server by manually sending an alert email:

   1. Log in to the Operations Console.
   2. Select an alert and click the envelope icon.

3. Complete the email dialog and send the email.
4. Verify the correct mailhost is defined in the following location:
   \Windows\System32\drivers\etc\hosts file
5. Restart the CA SAM Application Server service.

Escalation Actions - Tickets Not Created

Symptom:

I have created an escalation policy to create a help desk ticket, but CA SOI does not create the ticket.

Solution:

CA SOI may have lost the connection to the help desk server.

Follow these steps:

1. Enable server connection error notifications on the Dashboard Administration tab.
2. Test the help desk server.
3. Restart the CA SAM Application Server service.

Escalation Actions - Resolve Escalation Action Failures

Symptom:

I received an email notification that CA SOI has stopped an escalation policy action because the action continues to fail.

Solution:

CA SOI provides an error notification that automatically retries failed actions for a specified time period and then disables the action.

The following procedures describe how to resolve an action failure that is based on the action type.

Clear Alert

Follow these steps:

1. Try to clear the Alert manually. Right-click the alert and select Clear Alert.
2. Check the Clear Alerts flag on the Global Settings page, which is available on the Dashboard Administration tab.
3. Turn on debug tracing on the SA Manager and view the debug information:
   a. Access the SA Manager Debug page.
   b. Click Web Server Debug Page (Runtime).
   c. Locate the Alarm Data Model module and change the Desired State to ON and click Apply.
   d. Check for errors in the soimgr-debug.log file.
4. Enable the action once you resolve the error.

Create Announcement

Follow these steps:

1. Verify that all of the properties that are set for the Announcement are valid.
2. Verify that the help desk system is reachable and that the associated services are running.

Create Ticket

Follow these steps:

1. Verify that the Description ticket property is set in the action.

2. Verify that all properties set in the help desk system are valid.

Execute Automated Process in CA Process Automation

Follow these steps:

1. Review the failed process or form in the Default Process Watch in CA Process Automation.
2. Verify that the CA Process Automation server is reachable and that the CA Process Automation service is running.

Execute Command

Follow these steps:

1. Verify that the executable is in the system path.
2. Verify that the executable is functioning outside of CA SOI.

Send Email

Follow these steps:

1. Verify that the email server is configured. See email configuration documentation.
2. Ensure that a comma is used as a separator when using multiple email addresses.

Self Monitoring - Application CI Disappears

Problem: With Self Monitoring enabled, the Application CI in the CA Service Operations Insight service model disappears.

Resolution: Check to make sure that the Universal connector is installed and enabled. A running Universal Connector is required for self monitoring. If you removed the Universal connector from the Administration UI, it removes the Application CI in the CA Service Operations Insight service model. Re-install the Universal connector and restart the CA SAM Application Server service to re-create the CI.

Service Discovery - Default Significance does not Match CI Significance

Symptom:

I modified the default significance for a CI type (such as a Router). When I create dynamic relationships using Service Discovery, the significance is not reflected in the CIs.

Solution:

When you add CIs to a service in the Modeler, the default significance setting is evaluated dynamically on the UI Server. For the SA Manager to acquire these changes, restart the SA Manager service. The default significance is then assigned to new relationships that come in from Service Discovery or another connector.

Service Models - Resolve Looping Problems

Symptom:

CA SOI sent me an email notification that a looping problem can exist.

Solution:

You configured failure email notifications previously.

Perform the following actions to identify and resolve loops that CA SOI detects:

- A parent service is nested as a sub service of a child service.

  - Do not include the same service within its own service hierarchy.
  - To correct this type of loop, remove the child instance or parent service instance.
- Multiple linked bound relationships:
  - Do not attach bound relationships to other bound relationships.
  - To correct this type of loop, remove or replace one or more of the bound relationships with another relationship type.

SLA Recurrence Not Triggering

Symptom:

I have set an SLA to recur starting today, but it does not trigger.

Solution:

When you set the Recurrence date, it must be at least one day later than the Start Date. If the Start Date is Jan 1, 2013 and the Recurrence date is Jan 1, the first recurrence is Jan 1, 2014. However, if you set the Recurrence Date to Jan 2, the first recurrence happens on Jan 2, 2013.

Alerts Update are Delayed in the Alert Queues Tab

Symptom:

The alerts take more time to update in the **Alert Queues** tab.

Solution:

Follow these steps:

1. Verify the newly created policy.
2. Disable the created policy.

   NOTE

   check if the alerts take more time to update in the Alert Queues tab.
3. Identify the policy execution time.
4. Disable, delete, or change the policy based on the execution time.

SA Manager and UI Server Troubleshooting

Disk Full

Symptom:

The disk where I installed CA SOI is full.

Solution:

You can use a command to shrink the Microsoft SQL transaction log file to free some space. However, your database must be configured with the recovery model Simple. If the recovery model is Full, the shrink command does not work.

NOTE
For more information, see the MSDN document http://support.microsoft.com/kb/907511.

Follow these steps:

1. Perform a full database backup using the following command:

```
BACKUP DATABASE [SAMStore] TO DISK = N'C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\Backup\SAMStore.bak' WITH NOFORMAT, NOINIT,
NAME = N'SAMStore-Full Database Backup', SKIP, NOREWIND, NOUNLOAD,
STATS = 10
GO
```

2. Back up the transaction log:

```
BACKUP LOG [SAMStore] TO DISK = N'C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\Backup\SAMStore_log.bak' WITH NOFORMAT, NOINIT,
NAME = N'SAMStore-Transaction Log  Backup', SKIP, NOREWIND, NOUNLOAD,
STATS = 10
GO
```

3. Shrink the size of the log file:

```
DBCC SHRINKFILE (N'SAMStore_log', 100) WITH NO_INFOMSGS
```

   – *SAMStore_log*
     Specifies the logical transaction log file name.

   Sets the file size to 100MB. You can make it above or below this value depending on your disk space availability.

Error with Browser-Based UIs

Symptom:

Users are unable to access browser-based UIs such as the Dashboard, Mobile Dashboard, and USM Web View. The browser either indicates the page cannot be displayed, the certificate is invalid, or something similar.

Solution:

If you are using Windows XP/2003 clients through HTTPS, Microsoft provides a fix.

http://support.microsoft.com/default.aspx?scid=kb;EN-US;938397

All users must install this fix for their client.

Resolve an SA Store Database Connection Failure

Symptom:

I received an email notification that CA SOI has detected an SA Store Database connection failure.

Solution:

You set email notifications for database failures.

1. Verify that the server is configured correctly and functioning outside of CA SOI.
2. Verify that the SA Store connection is working correctly using the DB Connectivity debug page. There are debug pages for the SA Manager server and for the UI server:
   a. For each server, run the DB Connectivity report.
   b. View the report and review the following information:
      - The database connection, which determines if the server can connect to the SA Store Database at all.
      - The query speed test, which determines how fast the underlying database structure can respond to a simple query without accessing the database tables.
3. If the SA Manager server is offline and comes back online, CA SOI should detect the connection. If CA SOI is not connecting to the server, restart the SOI Application Manager service.

Resolve a Third-Party Server Connection Failure

Symptom:

I received an email notification that CA SOI has detected a third-party server connection failure.

Solution:

You previously set email notifications for third-party server failures.

Follow these steps:

1. Verify that the server is configured correctly and functioning outside of CA SOI.
2. If the server is offline and comes back online, CA SOI detects the connection. If CA SOI is not connecting to the server, restart the SA Application Manager service.

SA Manager Crashing on Startup or Dumping Memory and Performing Slowly

Symptom:

I notice that either the SA Manager crashes when I restart the service or the SA Manager is dumping memory and performing slowly.

Solution:

There is a known issue where the SA Manager is either crashing at startup or dumping memory and performing slowly.

The SA Manager is trying to read the following file into memory:

<SOI_HOME>\tomcat\webapps\sam\console\logs\client.log

The problem is that the file has grown very large.

Follow these steps:

1. Stop the CA SAM Application Server service.
2. Delete the client.log file.
3. Restart the CA SAM Application Server service.

WrapperStartStopAppMain Message

Symptom:

I have entries in the SA Manager log (soimgr.log) and the UI Server log (soiuis.log) with warnings similar to the following examples:

```
2013-10-15 11:23:32,547 WARN  [WrapperStartStopAppMain] config.ConfigurationFactory.parseConfiguration(133)
 - 2013/10/15 11:23:32:538 EDT [WARN] ConfigurationFactory - No configuration found. Configuring ehcache from
 ehcache-failsafe.xml  found in the classpath: jar:file:/E:/CA/SOI/wso2registry/repository/components/plugins/
ehcache-1.5.0.wso2v1.jar!/ehcache-failsafe.xml
2013-10-15 11:23:33,139 WARN  [WrapperStartStopAppMain] config.ConfigurationFactory.parseConfiguration(133)
 - 2013/10/15 11:23:33:138 EDT [WARN] ConfigurationFactory - No configuration found. Configuring ehcache from
 ehcache-failsafe.xml  found in the classpath: jar:file:/E:/CA/SOI/wso2registry/repository/components/plugins/
ehcache-1.5.0.wso2v1.jar!/ehcache-failsafe.xml
2013-10-15 11:23:33,198 WARN  [WrapperStartStopAppMain]
 ehcache.CacheManager.detectAndFixDiskStorePathConflict(322)  - 2013/10/15 11:23:33:197 EDT [WARN]
 CacheManager - Creating a new instance of CacheManager using the diskStorePath "../../samui/temp" which is
 already used by an existing CacheManager.
The source of the configuration was classpath.
```

```
The diskStore path for this CacheManager will be set to ../../samui/temp\ehcache_auto_created_1381850613197.
To avoid this warning consider using the CacheManager factory methods to create a singleton CacheManager or
 specifying a separate ehcache configuration (ehcache.xml) for each CacheManager instance.
```

Solution:

These warning messages generates when you restart the SA Manager and UI Server services. Ignore the warnings.

Customize the Default Session Timeout for a User in CA SOI

Symptom:

How do I change the default session timeout for a user in CA SOI?

Solution:

Set the timeout value in the iAuthrorty.conf file.

Follow these steps in the system where CA EEM Server is installed:

1. Stop the **CA iTechnology iGateway** service.
2. Perform *one* of the steps according to the CA EEM version.
   a. For CA EEM version 8.4, open **iAuthority.conf** in the **CA\SharedComponents\iTechnology** folder.
   b. For CA EEM 12.x, open **iAuthority.conf** in the **CA\SC\iTechnology** folder.
3. Add *<CredentialLifeTime>N</CredentialLifeTime>* tag before the *</iSponsor>* tag.

   NOTE

   N specifies the timeout value (hours). Based on your requirement, you can increase or decrease the timeout value.

   For example,

```
</TrustedRoot>

  <CredentialLifeTime>168</CredentialLifeTime>

</iSponsor>
```

4. Start the **CA iTechnology iGateway** service.

CA SOI Manager takes Long Time to Restart

Symptom:

CA SOI Manager takes longer time to restart than the expected time. The following error appears in the soimgr.log file.

*state.StateModeListener.*attributechanged*(170) -Error!!! loop detected*

Solution:

Follow these steps:

1. Stop all CA SOI services.
2. From the SQL Server system, access the SAM Store database.
3. Execute following commands:

```
select * from Alerts where Active=1 (save the result of this query)
```

```
update Alerts set Active=0 (this will clear all open alerts)

update Alerts set ClearedTime=getdate() where AlertID in (select AlertID from Alerts
 where Active=0 and ClearedTime is null)
```

4) Start the CA SOI Manager.

No Alerts in Operation Console

Symptom:

No alerts appear in Operation Console when connectors are up and running.

Solution:

Follow these steps:

1. Navigate to **CA SOI UI Server Debug Console**.
2. Click **Debug Pages**, **Queue Monitor.**

   NOTE

   Verify whether the status of jobs are pending in the Job Queues.
3. Wait for the job queues to be empty.
4. If the job queues are not empty, stop the Manager service, clean **tomcat\temp**, **activemq-data**, **logs** folder**,** and start the Manager service.
5. On the manager restart, all job queues are set to zero.
6. Wait for the manager service to be up and running.

SOI Toolbox Troubleshooting

Toolbox Fails to Run

Symptom:

Every time that I try to run the CA SOI Toolbox, I receive a message similar to *The system cannot execute the specified program* message.

Solution:

Check the Windows Event Log for an *Event Properties - Event 33, SideBySide* message. If you find this event in your log, download and install the Microsoft Visual C++ 2008 Redistributable Package (x86).

Toolkit Fails to Update Configuration Files

Symptom:

1. SOI HA toolkit picks an incorrect mapping share drive with the cluster disk name. The HA toolkit shows different cluster disk name in “MS Failover Cluster Manager”.
   The following logs are found in ssaha-debug.log:

   ```
   1/4/2017 4:48:43 AM: GetDiskResourceName: Comparing Drive 'F:' with Cluster Resource
    Drive 'Q:' Cluster Resource Name - 'Cluster Disk 1'
   ```

```
1/4/2017 4:48:43 AM: GetDiskResourceName: DiskDrive Name: \\.\PHYSICALDRIVE1
 DiskDrive Sig: BD223A37

1/4/2017 4:48:43 AM: GetDiskResourceName: DiskDrive and Cluster Resource signature
 matched: BD223A37

1/4/2017 4:48:43 AM: GetDiskResourceName: Win32_DiskDriveToDiskPartition Error 424

1/4/2017 4:48:43 AM: GetDiskResourceName: PartitionDeviceID: Disk #1, Partition #0
```

2. SOI HA toolkit does not work due to environment issues.

Solution:

As a workaround, update the following config files to replace SOI Physical Hostname with SOI Virtual Name:

File	Location
caifwmq.xml	SOI\tomcat\webapps\activemq-web\WEB-INF\
server-config.xml	\SOI\tomcat\webapps\sam\
• restserver.xml • sorapp.xml • ssaserver.xml • wsman.properties	\SOI\tomcat\registry\physical\node0\sor\
• jmsconnect.properties • eventManagerClientConfig.xml	\SOI\tomcat\lib\
• SSA_IFW* • mtc_*	\SOI\resources\Configurations\
eventManagerServerConfig.xml	\SOI\resources\

USM Web View Troubleshooting

USM Web View Does Not Display All CIs

Symptom:

When I perform a search in USM Web View, I do not see all CIs that the CA SOI Connectors import.

Solution:

CIs are not imported from CA SOI into the CA Catalyst Database until the CIs are modeled as part of a CA SOI service. Once a CI becomes part of a CA SOI service, a projection is created and published to the CA Catalyst Persistence Store through the CA SOI CA Catalyst Connector.

If the CI is part of a modeled service, view the SOIConnector.log to verify that a projection was received. Also, see the invalidCIs.log to determine if the CI was rejected for some reason.

USM Web View Search Returns Incorrect Results

Symptom:

When I perform a search in USM Web View, one or both of the following events occur:

- Web View interface full text search does not find a few CIs, but the CIs are in the database and can be accessed through Browse.
- Or, Web View interface full text search shows few CIs that have been already deleted. The CIs are not accessible through Browse.

However, after I perform a reindex, the CIs are correctly found using full text search.

Solution:

To avoid potential search problems in USM Web View, synchronize the time on all machines where CA SOI and CA Catalyst are installed. Either configure the machines in the same Windows Server domain (so the domain controller synchronizes the time) or synchronize over Network Time Protocol (NTP).

The items that are created, updated, and modified in the CA Catalyst database are continuously indexed. The items are accessible using the full text search. The process that indexes them recognizes these items by a timestamp. If the machines are not time-synchronized, it is possible that some older create/update/delete timestamp is inserted after a newer one. If the indexer process ran between these two inserts, the older item is ignored and therefore unavailable (or still available after deletion) in the full text search.

Cannot log in USM Web View after Password Change

Symptom:
I cannot log in to USM Web View after I change the EEM Admin user password.

Solution:

Follow these steps:

1. Back up the following files:
   - <SOI_HOME>\tomcat\webapps\sam\eem-config.xml
   - <SOI_HOME>\SamUI\webapps\sam\eem-config.xml
   - <SOI_HOME>\SamUI\custom\eem-config.xml
   - <SOI_HOME>\tomcat\custom\eem-config.xml
   - SamUI\conf\jaas.config file
   - <SOI_HOME>\tomcat\registry\topology\physical\node0\sor\eem.properties
2. Open command prompt, navigate to <SOI_HOME>\tools, and execute the following command: EncryptSAMCreds.bat <new password>
   An encrypted password appears.
3. Open the following files and replace the bold-faced text in the <password plain="false"><**![CDATA[ENi8GAx70iGluRfypSJ6SXEeXDJ8WviUNQj1YW/LtR3C]]** ></password> block with the encrypted password obtained in step 2:
   - <SOI_HOME>\tomcat\webapps\sam\eem-config.xml
   - <SOI_HOME>\SamUI\webapps\sam\eem-config.xml
   - <SOI_HOME>\SamUI\custom\eem-config.xml
   - <SOI_HOME>\tomcat\custom\eem-config.xml
4. Save the changes and close files.
5. Open command prompt, navigate to <SOI_HOME>\Tools\CatalystEncrypt, and execute the following command: encrypter <new password>
   An encrypted password appears.
6. Open the jaas.config and eem.properties files, paste the encrypted password obtained from step 5 as the value for the eiam.query.password property, and save the changes and close the file.
7. Run registryloader.bat located at <SOI_HOME>\tomcat\registry.

8. Ignore any "log4j:WARN" messages. The Registry loads an updated record of all files in which you changed CA EEM connection information.
9. Restarted all the services through SOIToolbox.

Service Discovery Troubleshooting

Service Discovery Connector Logging

For troubleshooting purposes, you can change the setting determining which events will be written to the service discovery log file.

The log file of the Service Discovery connector is located at <SOI_HOME>/log/service-discovery.log. The default logging severity is INFO and the severity can be changed in <SOI_HOME>/resources/log4j.xml of the logger com.ca..ssa.servicediscovery.

```
<logger name="com.ca.ssa.servicediscovery" additivity="false">
 <level value="INFO"/>
 <appender-ref ref="SD"/>
</logger>
```

The following table shows what information is logged with different logging severities:

Severity	Logged Events
ERROR	Network and database errors
INFO	Information about the connector start and about policies that have been changed
DEBUG	UCF calls: requests to create and delete CIs/relationships that are sent by the connector to the SOI Manager
TRACE	SQL statements executed by the service discovery connector

Reporting Troubleshooting

Contents

Multiple Instances of Reports Generating

Symptom:

BusinessObjects is generating multiple instances of reports.

Solution:

This is an issue with CA Business Intelligence r3.1 SP3 and below. Upgrade to CA Business Intelligence r3.3 SP1 or above. For more information, see Software Support.

Hibernate Errors Appear when Reports are run from CABI JasperReports Server 6.1

Symptom:

After successfully running the reports from CABI JasperReports Server 6.1, hibernate error messages, such as “Socket Write Error” and “Last Packet Not Finished” appear and we are unable to run reports.

Sample error messages:

org.springframework.transaction. CannotCreateTransactionException: Could not open Hibernate Session for transaction; nested exception is org.hibernate. TransactionException: JDBC being failed:

org.hibernate.TransactionException: JDBC begin failed:

org.springframework.tranaction. CannotCreateTransactionException: Could not open Hibernate Session for transaction; nested exception is java.lang.AssertionError: Last packet not finished

java.lang.AssertionError: Last packet not finished

Cause:

c3p0 hibernation setting is missing.

Solution:

1. Add the following property key values in hibernateProperties in <js-app>/jasperserver-pro/WEB-INF/applicationContext.xml file:

```
<property name="hibernateProperties">

      <props>

        <prop key="hibernate.dialect">${metadata.hibernate.dialect}</prop>

        <prop key="hibernate.show_sql">false</prop>

        <prop key="hibernate.generate_statistics">true</prop>

        <!--uncomment property below if a default schema should be specified such as
 for DB2-->

        <!--<prop key="hibernate.default_schema">
${metadata.hibernate.default_schema}</prop>-->

        <!--Cache Configurations-->

        <prop key="hibernate.cache.region.factory_class">
${hibernate.cache.region.factory_class}</prop>

        <prop key="net.sf.ehcache.configurationResourceName">/ehcache_hibernate.xml</
prop>

        <prop key="hibernate.cache.use_minimal_puts">false</prop>

        <prop key="hibernate.cache.use_query_cache">true</prop>

        <prop key="hibernate.jdbc.batch_size">20</prop>

        <prop key="hibernate.cache.use_second_level_cache">true</prop>

        <prop key="hibernate.cache.use_structured_entries">true</prop>
```

```
        <prop key="hibernate.c3p0.min_size">5</prop>

        <prop key="hibernate.c3p0.max_size">200</prop>

        <prop key="hibernate.c3p0.timeout">300</prop>

        <prop key="hibernate.c3p0.max_statements">500</prop>

        <prop key="hibernate.c3p0.idle_test_period">60</prop>

        <prop key="hibernate.c3p0.acquire_increment">2</prop>

        <prop key="hibernate.c3p0.testConnectionOnCheckin">true</prop>

    </props>

  </property>
```

2. Restart the Tomcat services.

Connectors

CA SOI uses connectors and the CA Catalyst infrastructure to connect to different domain managers. This section defines connectors and provides installation, configuration, and customization information for connectors.

Intended Audience

The following roles comprise the primary audience for this section:

- **CA SOI administrators**
  Install, configure, and maintain the CA SOI solution. These administrators work closely with the operations team and the service owners to define and build service models.
- **Domain manager administrators**
  Install, configure, and maintain domain-specific connectors and understand how connectors can assist them in development and integration.
- **Integration developers**
  Deliver new integrations with CA SOI.

Introduction to Connectors

This section introduces connector concepts. For information about installing connectors, see Connector Installation. For information about managing installed connectors, see Connector Administration.

Connectors Overview

Contents

A connector is software that provides the interface for data exchange between the CA Catalyst infrastructure and a domain manager. Connectors are the gateway through which data is retrieved from various domain managers for consolidated management. Each integrated product has its own connector that supports one or both of the following operation types:

- **Outbound from connector**
  Outbound from connector operations collect data (such as services, CIs, topology, alerts, and status) from the source domain manager. Data retrieved by connectors flows through the integration framework (IFW) and MQ server to the manager components. The manager components facilitate the reconciliation and display of the data on the product interfaces. All provided connectors support outbound operations.
- **Inbound to connector**
  Inbound to connector operations use records in the CA Catalyst Persistent Store to create, update, or delete items in the source domain manager. Inbound operations enable domain manager synchronization with changes spurred by CI reconciliation, CI creation, and CI updates in other domain managers. This synchronization helps reflect these changes in all domain managers. Many provided connectors support inbound operations.

You can configure, start, and stop connectors from the Administration UI.

> **NOTE**
> Each CA Catalyst Connector Guide contains information about connector installation, configuration, how the connector interprets data from its domain manager, and whether inbound operations are supported.

Bidirectional Connector

A *bidirectional connector* supports both inbound and outbound operations. Outbound-only connectors contain one connector policy file that transforms the gathered data to the standard USM format. Bidirectional connectors contain two connector policy files that transform outbound data to the USM format and transform inbound data to the source format of the domain manager.

Custom Integrations

CA SOI also provides the following tools for defining custom integrations:

- Universal connector
  Provides a web services interface that products can use to publish new services, CIs, and events, which are normalized to a common format and made available to the SA Manager. The Universal connector can retrieve services, CIs, and status events from various CA Technologies and third-party products.
- **Connector SDK**
  Provides the ability to develop custom connectors. The SDK includes a Sample connector, which provides the framework for writing a connector to integrate with important applications in your enterprise.

Connector Documentation

This section contains the following information:

- General information that applies to all connectors, such as architecture, basic configuration, and troubleshooting
- Detailed information about connectors provided on the CA SOI image, such as the Event and Sample connectors
- Advanced information about how to build custom connector integrations

All CA Catalyst connectors, which are provided separately from the CA SOI image as published solutions, contain a Connector Guide and Readme specific to that connector. This documentation contains the following information:

- Connector overview
- Prerequisites
- Detailed installation and configuration procedures
- Details about the data imported from the connector and other connector capabilities

Access connector-specific documentation from the Documentation directory of the connector package or from the <SOI_HOME>\Documentation directory after you install the connector.

Connector Infrastructure

Contents

This section explains the connector infrastructure, which includes the Integration Framework, MQ Server, UCF Broker, Event Management, and USM.

Integration Framework

The *integration framework (IFW)* is the mechanism that CA SOI uses to connect to domain managers and gather CI, service, topology, and state information.

The IFW exists on any system with a connector or the SA Manager, and it interfaces with the connector framework to prepare connector data for transmission to the manager components. The IFW contains a transformation engine that uses connector policy to transform connector data to the USM format. The IFW also includes the infrastructure of the Event Management component, which provides the mechanism for storing events from connectors for exposure to event policy and eventual display as alerts after event processing completes.

The IFW uses the Apache MQ message broker, which fully implements the Java Message Service (JMS) as its protocol.

Apache Active MQ JMS Server

Apache ActiveMQ is an open source message broker that implements the Java Message Service 1.1. The MQ Server controls all messaging and communication from external sources. The server also receives alerts and CI information from connectors and sends this information to various components for storage and analysis.

Event Management

Event Management provides a way to manage the event (USM type Alert) data from connectors more granularly and to control what makes into the Operations Console as alerts. It provides a processing layer between raw connector USM alert data and alert management. With Event Management, you can control the event stream so that a consolidated, high quality, and actionable set of alert conditions appear in the Operations Console as alerts.

Unified Service Model

The Unified Service Model (USM) is the schema that defines the internal format for all CA Catalyst data. Connectors transform all data collected from domain managers to the USM format before they send the data through CA Catalyst. The USM schema is stored in the CA Catalyst Registry.

UCF Broker

The UCF Broker is a communication layer that controls access to the enabled bidirectional connectors, which can invoke inbound to connector operations on source domain managers. The Logic Server communicates synchronization changes to bidirectional connectors through the UCF Broker.

How Connectors Fit into CA SOI and CA Catalyst

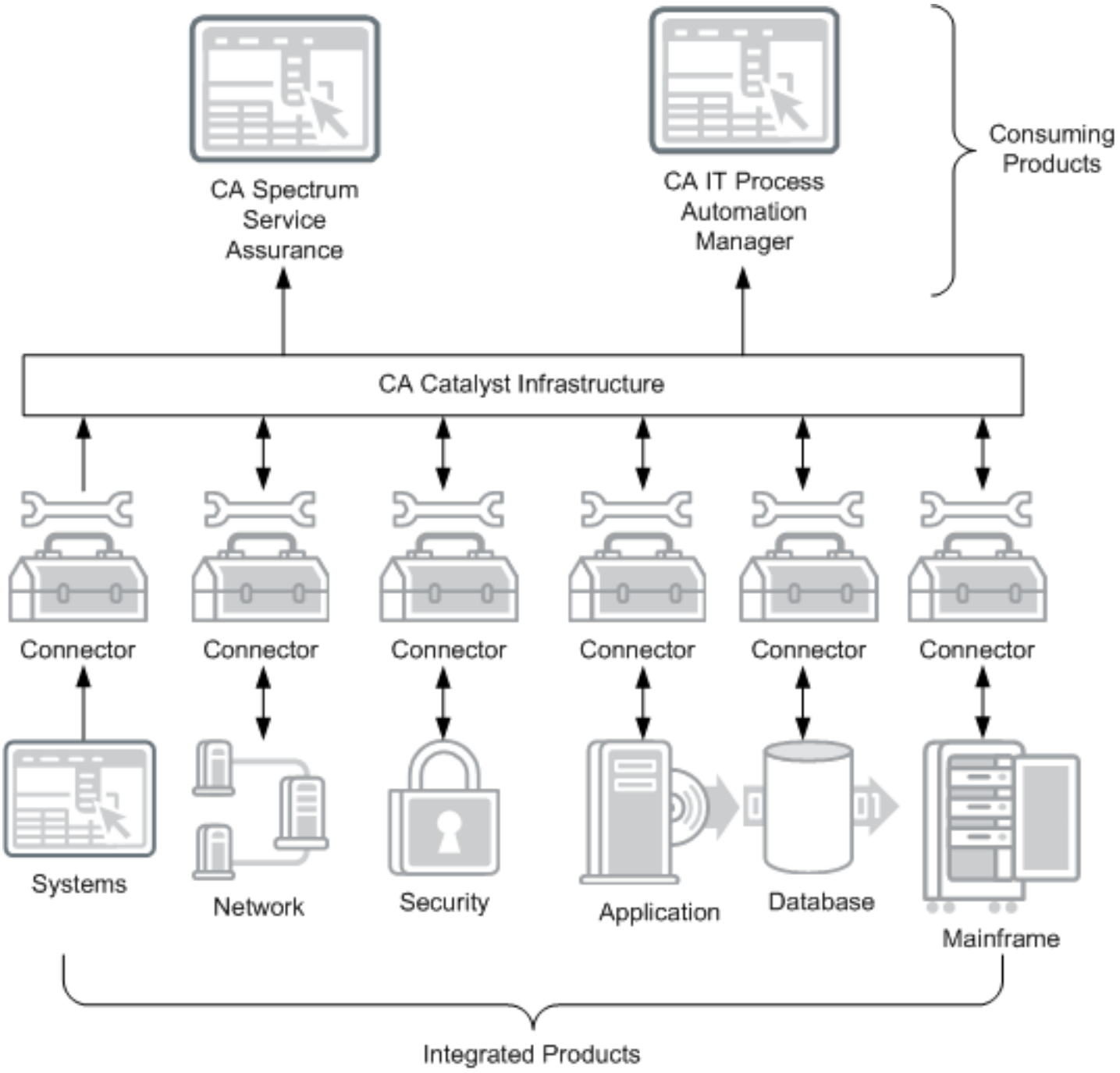
Connectors integrate through a common web services framework called the Unified Service Model (USM) to expose data from a wide range of CA Technologies and third-party products for management by a set of consuming products.

The *Unified Service Model (USM)* is the semantic schema that is used as the CA Catalyst and CA SOI infrastructure.

Connectors interact with consuming products and integrating products as follows:

- **Consuming Products**
  Consuming products leverage the infrastructure by consuming information that the connectors retrieve from integrated products. Consuming products provide a consolidated view of connector data for management in a unique, broader context. Consuming products and solutions include the following:
  - CA SOI
    CA SOI consolidates data retrieved by connectors and lets you model services to represent cross-product data based on logical business functions. CA SOI is a layer of management that brings together the information from domain management products.
  - CA Process Automation
    CA Process Automation lets you automate cross-product workflows by modeling processes that use functionality available from integrated products. CA Process Automation uses connectors to retrieve data from integrated products for use in process modeling.
- **Integrated Products**
  Integrated products are the products where connectors retrieve data. Each integrated product has a separate connector developed specifically to integrate with that product.

The following illustration shows how connectors retrieve data from integrated products, transform data into a common format using the infrastructure, and expose the data for consuming products to manage:



Connector Integration Types

Contents

The CA SOI infrastructure provides different types of connector integrations to accommodate different levels of detail. The following are the various levels and types of connector integrations that exist in provided connectors and that you can use to develop custom integrations:

- Level 1 connectors of type *CLI Based*
- Level 2 connectors of type *Generic Collection*
- Level 3 connectors of type *Process/Workflow*
- Level 4 connectors of type *Outbound without Service Import*
- Level 5 connectors of type *Outbound with Service Import*
- Level 6 connectors of type *Outbound and Inbound*

Level 1 Connectors

Level 1 connectors are *CLI-based* that can manually publish information to CA SOI. The provided Universal connector is a level 1 connector that lets you establish level 1 integrations. A Level 1 connector provides the following capabilities:

- Supports publishing of data to the consuming product (Publish)
- Supports importing of service representations (Service Import)
- Supports CIs from a domain manager
- Supports alerts from a domain manager

Level 2 Connectors

Level 2 connectors are of type *Generic Collection*. They can dynamically collect data from a generic source but cannot subscribe for changes to collected data. A Level 2 connector provides the following capabilities:

- Supports publishing of data to the consuming product (Publish)
- Supports CIs from a domain manager
- Supports alerts from a domain manager

Examples of Level 2 connectors include generic CA Catalyst connectors (such as SNMP) and the Event connector provided with CA SOI. You can use these connectors to establish custom Level 2 integrations based on generic data such as SNMP traps, text logs, and so on.

Level 3 Connectors

Level 3 connectors are of type *Process/Workflow*. They are responsible for complex process orchestration (for example, CA Process Automation workflow-based connectors). They include the ability to publish results for consumption by other management domains, subscribe to events from other connectors, and take actions based on a filtered set of events, CIs, or both to accomplish some form of business function (for example, provisioning a computer system or enabling the user account). A Level 3 connector provides the following capabilities:

- Supports publishing of data to the consuming product (Publish)
- Supports subscription to the domain manager changes (Subscribe)
- Supports inbound-to-connector operations: create, retrieve, update, and delete
- Supports advanced features such as metrics and custom operations.

> **NOTE**
> No Level 3 connectors are currently available that CA SOI can use.

Level 4 Connectors

Level 4 connectors are of type *Outbound without Service Import*. They can publish data and subscribe to domain manager changes but cannot import a service representation. A Level 4 connector provides the following capabilities:

- Supports publishing of data to the consuming product (Publish)
- Supports subscription to the domain manager changes (Subscribe)
- Supports advanced features such as metrics and custom operations, if applicable
- Supports CIs from a domain manager
- Supports alerts from a domain manager

Examples of Level 4 connectors include the NetAPP SANscreen connector and IBM Tivoli Monitoring connector.

You can build a custom Level 4 connector using the connector SDK provided within the Sample connector.

Level 5 Connectors

Level 5 connectors are of type *Outbound with Service Import*. A Level 5 connector provides the following capabilities:

- Supports publishing of data to the consuming product (Publish)
- Supports subscription to the domain manager changes (Subscribe)
- Supports advanced features such as metrics and custom operations, if applicable
- Supports CIs from a domain manager
- Supports alerts from a domain manager
- Supports importing of service representations

Level 6 Connectors

Level 6 connectors are of type *Outbound and Inbound*. These connectors can perform both inbound-to-connector and outbound-to-connector operations. A Level 6 connector provides the following capabilities:

- Supports publishing of data to the consuming product (Publish)
- Supports subscription to the domain manager changes (Subscribe)
- Supports advanced features such as metrics and custom operations, if applicable
- Supports CIs from a domain manager
- Supports alerts from a domain manager
- Supports importing of service representations (Service Import)
- Supports inbound-to-connector operations: create, retrieve, update, and delete (if applicable)

Examples of Level 6 connectors include the IBM Tivoli Service Request Manager connector and BMC Atrium/Remedy connector.

Summary of Connector Integration Levels and Types

The following table summarizes the connector levels, types of connectors, and supported functionality:

Level	Type	Publish	Subscribe to Domain Manager Changes	Service Import	CIs	Alerts	Advanced Features (Metrics, Custom Operations)	Inbound-to-connector Operations
1	CLI Based	Yes	No	Yes	Yes	Yes	No	No
2	Generic Collection	Yes	No	No	Yes	Yes	No	No
3	Process/Workflow	Yes	Yes	No	No	No	Yes	Yes

4	Outbound without Service Import	Yes	Yes	No	Yes	Yes	Yes (if applicable)	No
5	Outbound with Service Import	Yes	Yes	Yes	Yes	Yes	Yes (if applicable)	No
6	Outbound and Inbound	Yes	Yes	Yes (if applicable)	Yes	Yes	Yes (if applicable)	Yes

Connector Integration Example Scenarios

Contents

This section includes connector integration example scenarios for the Universal connector, Generic connector, Event connector, and CA SOI connector.

Universal Connector Integration

The Universal connector provides command line and web services programming interfaces that let the connector forward CIs, events, and services to CA SOI. The Universal connector is appropriate for Level 1 integrations with CA SOI provided any of the following conditions are met:

- The domain manager has a script interface that calls the Universal connector command line interface to create CIs and alerts in CA SOI.
- The domain manager has a programming interface that the Universal connector web services interface can use to create CIs and alerts in CA SOI.
- Custom events are forwarded to CA SOI.

The Universal connector only forwards events to the SA Manager, which indicates the following:

- The integrated application must call the Universal connector interface.
- It is not possible to get information from the integrated application. For example, you cannot import services from the Universal connector in the Operations Console.
- Because CA SOI cannot get the current status for a CI, the status for the Universal connector CIs might not be current after a CA SOIserver restart or in newly created service models.

Generic Connector Integration

Generic data collection connectors can dynamically collect data from various generic sources. They process and convert the data to the USM format using a connector policy and dispatch it to the consuming product. However, unlike product-specific connectors, generic connectors do not include any product-specific policy. One fixed default policy for the generic connector is not possible, because the type and format of the received information varies from one domain manager to another. Therefore, for Level 2 integrations using a generic connector, you must customize a template policy file so that the policy appropriately processes data from the data source and displays meaningful information in CA SOI.

The CA Catalyst connector for SNMP is a good example of Level 2 integration. This SNMP connector acts as an SNMP trap destination for any product that supports SNMP, which helps ensure that you get access to the SNMP trap data from a wide range of products. The ability to collect, transform, analyze, and manage SNMP trap data in CA SOI lets you determine trends and take appropriate measures for SNMP-capable devices in your enterprise. For example, you can analyze an SNMP trap in CA SOI to identify the root cause of the trap. You can determine whether the trap is because of an extraordinary event (indicating an issue or an error has occurred) or a confirmed event providing status information, such as a process ending normally or a printer coming online. Based on your analysis, you can take corrective actions to fine-tune or rectify the situation.

While developing a generic connector (Level 2) integration, consider the following:

- You can monitor additions and modifications to events (such as SNMP traps) coming from a domain manager by using a generic connector.
- For each new entity that generates an event, the first event is considered as a CI and an alert, and any subsequent event from the same source is considered only as an alert.
- You cannot subscribe for changes to collected domain manager data using a generic connector.
- You cannot support importing of service representations, advanced features (such as metrics and custom operations), and inbound-to-connector operations using a generic connector.
- You can map object types of the integrating domain manager as appropriate. For example, in case of the SNMP connector, any CA Workload Automation job scheduled on the specific agent host is a CI and is mapped to the USM type ITActivity.
- You must write a policy specific to the integrating domain manager.
- You must formulate, understand, and follow the event mapping guidelines for writing the policy. For example, in case of the SNMP connector, all system-level traps from an SNMP product are mapped to the ComputerSystem USM type; similarly, all CPU-related traps are mapped to the Processor USM type, and so on.

  NOTE
  For more information about how to download the connector and accompanying documentation, see the Download Connectors and Prepare for Installation section.

NOTE
In those scenarios where both the connectorsgeneric connector and the Event connectorare applicable, we recommend that you use the generic connector.

CA SOI Connector Integration

You can achieve the deepest and most robust level of integration when building a custom Level 4, 5, or 6 CA SOI connector. You develop a custom connector in Java using the connector SDK provided with the Sample connector. A custom connector has the following characteristics:

- Receives requests from the SA Manager
- Synchronizes with the SA Manager
  A CA SOI server restart or creation of a new service model causes a request to the connector for the current CI alerts and verifies that the CI status in CA SOI is up-to-date
- Queries the third-party manager native interface (API or files) for CIs and alerts
- Implements an event listener for forwarding to CA SOI
- Supports service import requests from the Operations Console (if applicable)
- Embedded in the Windows service CA SAM Integration Services

You can also build advanced features into custom connectors, such as the following:

- Metric collection
- Custom operations
- Inbound to connector operations

NOTE
For more information, see the "Building a Custom Connector Using the Sample Connector" section.

Basic Connector Information

For generic information about installing connectors and managing installed connectors, see Connector Installation and Connector Administration.

Unified Service Model

Contents

This section explains the basic concepts of the Unified Service Model (USM).

What is USM?

The *Unified Service Model (USM)* is the semantic schema that is used as the CA Catalyst and CA SOI infrastructure.

USM was developed to abstract and integrate information across many management products and domains and provide a single point for data federation, interoperability, and access to management data across an enterprise.

The USM schema defines the internal format for all CA Catalyst data, which is transmitted for display to CA SOI. Connectors transform all data collected from domain managers to the USM format before sending the data through CA Catalyst to CA SOI.

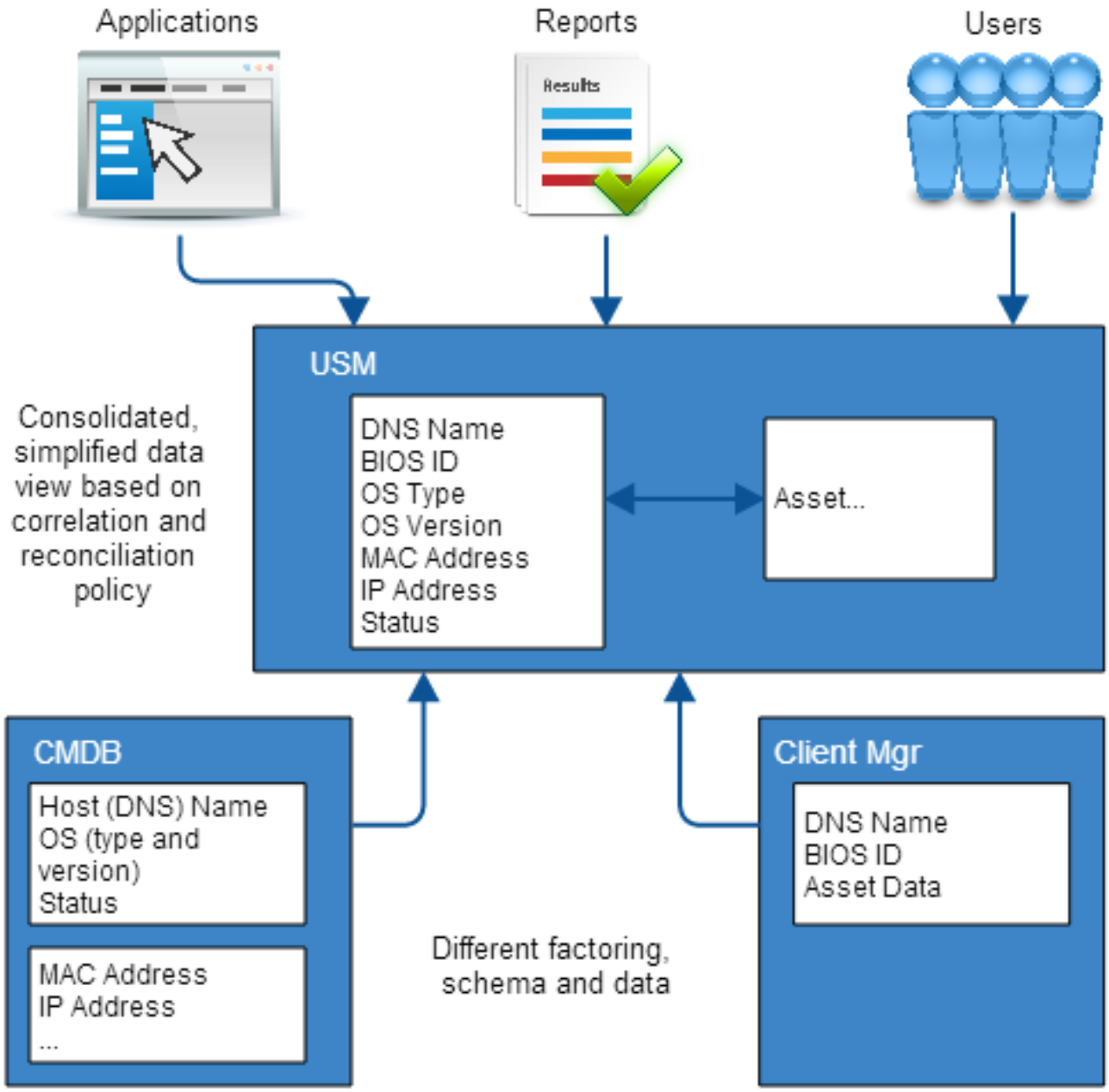
In IT management, specific domains exist such as networks, systems, fault, or security management systems and also specific standards exist for the domains. No mediating set of abstractions can discern the core concepts from the details and enable the merging of the specific domain semantics and data. USM addresses the mediation by defining logical, coherent, high-level, and common representations of general resource types, their data, and operations.

Specific data properties for USM types enable the following:

- High-level state management (available or unavailable)
- General alert management (informational to fatal alerts)
- CI identification and correlation
- Simple query (discover where to get more information)
- Facilitation of data manipulation, interoperability, and federation across a variety of IT management products

Data in different product-specific formats normalize to a single, consistent format that is correlated, reconciled, and accessed by users, applications, and reports.

Figure 59: USM_1



CA Catalyst is the integration and application platform that implements the USM schema, and CA SOI displays CA Catalyst data in USM format.

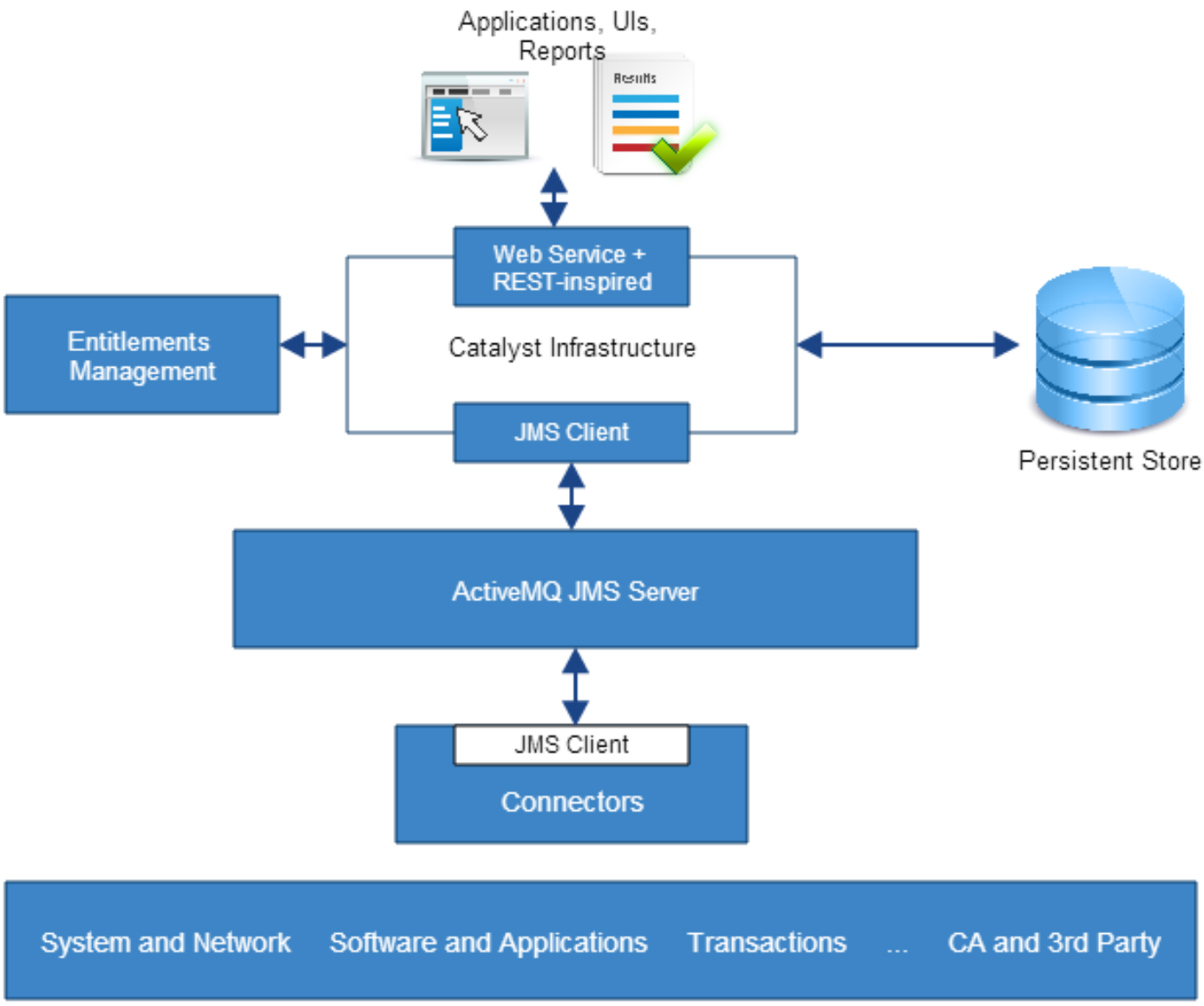
USM Schema

The USM schema provides facilities for CA Technologies and third-party management applications to do the following:

- Publish USM data and receive notifications of changes
- Persist and reconcile USM data
- Access data through web services and application interfaces

The following illustration shows the CA Catalyst infrastructure and how data from the infrastructure displays in CA SOI:

Figure 60: USM_2



This infrastructure is critical to fully using USM semantics. Because each management product supports a limited set of management domains, none can provide a complete view of the IT and business environment. CA Catalyst facilitates the transformation of product- and domain-specific data to the USM syntax and semantics and keeps products notified of changes, additional information, and alerts published by other applications. The complete data set in the CA Catalyst store is correlated and reconciled, which creates a more inclusive and accurate representation of the environment.

Correlation

Correlation compares the key property values of a CI from one domain manager to another domain manager. CA Catalyst performs correlation and CA SOI, as a CA Catalyst consumer, can receive the correlated data from CA Catalyst. In CA Catalyst, correlation is based on one or more algorithms to achieve this mapping. You can adjust certain correlation

criteria using additional semantics such as AND, OR, ANDIFEXIST, and NOT to modify the default rule. You can use the default rule, also named as PriorityListUSM, that better suits a solution.

Also, you can configure correlation to correlate across types that share a common base type. This results in *one* of the following:

- In a derived CI type correlating to a base type, which indicates both are identical
- In peers (children of the same base type) correlating, which is a multifunctional device (for example, a computer system acting as a router)

CA Catalyst supports relationship correlation that involves comparing source, target, scope, and semantics of a relationship with another CI from a different domain.

To correlate instances, the CA Catalyst infrastructure compares the following information from the USM schema:

- Type (root element) of the instance
- Correlation rule
- Multi CI correlation inclusion rule
- Equivalent element across types
- Identifying name for the instance (the element, InstanceName)

> **NOTE**
> For more information about correlation, see the CA Catalyst documentation.

Reconciliation

Reconciliation is the act of resolving differing property values across multiple CI projections into a single reconciled CI.

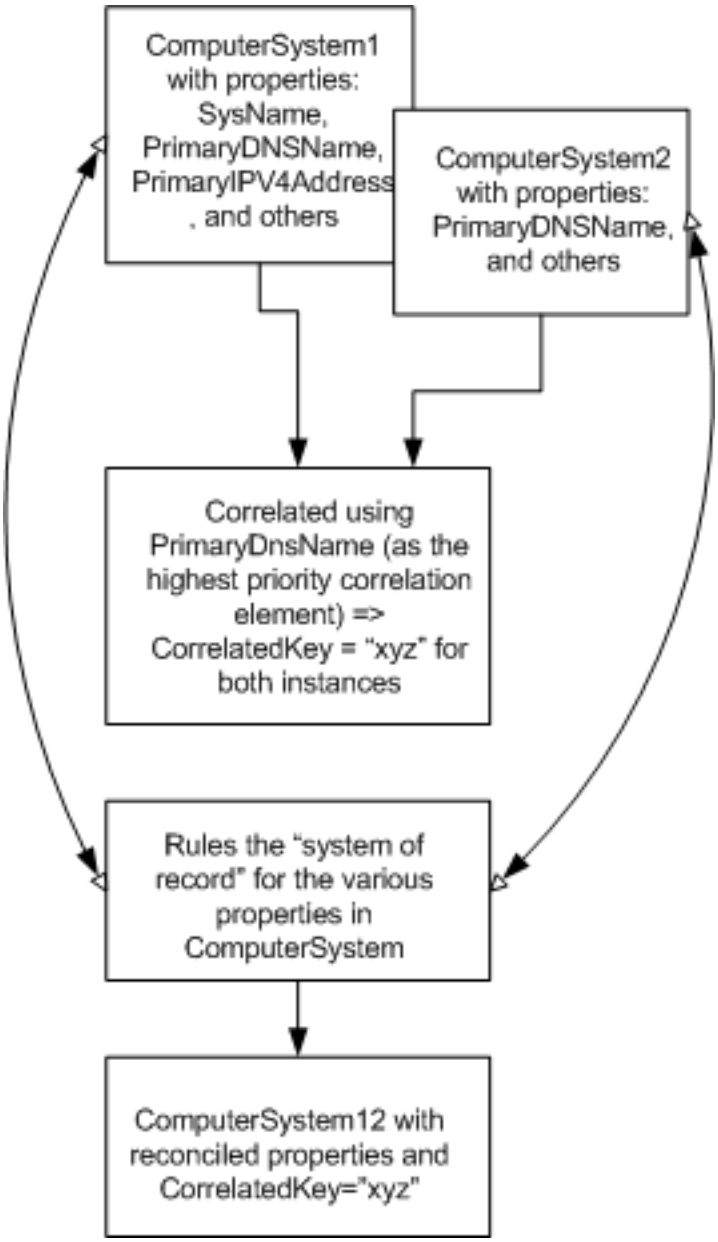
Reconciliation operates across correlated CIs by using reconciliation formulas to determine property values that take precedence when discrepancies occur.

You can set various reconciliation formulas, including the following:

- First non-null wins
- Majority wins
- Data from specific domain manager wins
- Last in wins (default formula)

> **NOTE**
> For more information about reconciliation, see How to Perform CA Catalyst Reconciliation.

The following illustration shows the correlation and reconciliation of two ComputerSystem CIs:



This illustration shows three instances of ComputerSystem1 from CA CMDB, one from CA ITCM, and a reconciled instance that CA Catalyst created. Some of the correlation elements overlap with each other and have the same CorrelatedKey value.

Advanced Features

The following are the advanced USM features:

- Custom Operations
- Metrics
- Profiles
- Filters

Custom Operations

Custom operations are operations that a connector can invoke to implement domain manager functionality that may be beneficial to a wider range of domain managers integrated through CA Catalyst. For example, a connector for an incident management product could implement a custom operation to create an incident for an alert on a CI managed by multiple domain managers.

> **WARNING**
> CA Catalyst consumers directly invoke custom operations. While developing custom connectors, you can build custom operations into those connectors, and consuming products that support the custom operations can then use them. However, CA SOI currently does not support invoking custom operations from its user interfaces.
>
> **NOTE**
> For more information about supported custom operations, see the "USM Specifics" appendix.

Metrics

Metric represents a measurement or reported value that characterizes the operation of a CI, such as a counter of incoming messages or a gauge reporting CPU utilization. The abilities to report the metrics available for an entity and to obtain the metric values are accomplished through a custom WSDL. A metric changes in response to external stimuli, as opposed to being configured.

Metrics data is gathered using the two USM custom operations, GetAvailableMetrics and GetMetricsValues. Connectors derive metrics from imported domain manager data and forward the same metrics information to the consuming product for viewing and analysis.

> **WARNING**
> CA Catalyst consumers directly support metrics. While developing custom connectors, you can build metrics information into those connectors, and consuming products that support metrics can then use them. However, CA SOI currently does not support viewing of collected metrics in its user interface.
>
> **NOTE**
> For more information about supported metrics, see the "USM Specifics" appendix.

Profiles

Profiles help you define a minimum standard set of capabilities and behavior across connectors. Usage of a profile facilitates interoperability and integrations. The Incident Profile is an example of the USM profile. Various CA Catalyst connectors such as BMC Atrium/Remedy and IBM Tivoli Service Request Manager use the Incident Profile.

> **NOTE**
> For more information about the Incident Profile, see the "USM Specifics" appendix.

Filters

Filters in the context of CA SOI are provisions that help connectors understand what type of data is being requested and retrieved from the domain manager. The get() connector method uses filters to retrieve information based on the predefined conditions.

SiloDataFilter is the primary filter that CA SOI uses when calling interfaces that the connector implements. SiloDataFilter supports various filter elements (such as entitytype, itemtype, recursive, and so on). Using combination of these filter elements, you can retrieve specific information from the domain manager.

For example, if the entitytype filter element is set to Item and itemtype to ComputerSystem, the connector would search for and display all entities of type ComputerSystem; it would not display any other entity. This way, you get to view and analyze only the required data in which you are interested.

How to Find USM Properties for a CI

Contents

As an integration developer, you provide USM properties information while using the Universal connector command line utility. You can provide complete and correct information if you know how to find various USM properties that are associated with a CI. This scenario helps you locate all USM properties for a CI.

Use the following methods to find the USM CI properties:

- Use the USM Web View Interface.
- Use the USM Schema Documentation.

Use the USM Web View Interface

You can use the USM Web View interface to browse and search for all USM data that is available in the CA Catalyst Persistence Store.

You can access USM Web View from the CA SOI Dashboard. To access the USM Web View interface from the CA SOI Dashboard, open the CA SOI Dashboard and click the USM Web View link available on the Dashboard. The USM Web View interface opens to the home page.

Use the USM Schema Documentation

The USM schema HTML documentation includes detailed information about all USM types, USM properties, schema summaries, and so on.

How to Access the USM Schema Documentation

To access the USM Schema documentation, navigate to Catalyst USM Schema on the SA Manager Debug Console.

USM Parts

Contents

The Unified Service Model (USM)Schema files and documentation link in the Document Library of the CA USM-Catalyst Global User Community page provides information about the USM parts.

XSD Files

The following are XSD files in the USM schema:

- usm-core-200907
- usm-extensions-201001
- usm-metadata-200907
- usm-metrics-200907

Schema contents are defined by the usm-core-200907.xsd file. This file contains the type declarations for all root elements in USM.

The usm-extensions-2001001.xsd file contains properties that you can add to specific USM types, but are not expected to be included in most instances of these types. For example, the USM BinaryRelationship type is instantiated to associate instances and the type's semantic property defines the specific meanings to ascribe to the association. In some cases, this type defines an ordering of the relationships and you use the usm-exts:Order property. This type is inserted into the BinaryRelationship instance at the location of the xs:any declaration.

Information about the new property and meta-data that describes the type extended are included in the usm-extensions XSD.

Based on this example, the following is the definition for the Order property:

```
<xs:element name="Order" type="usm-meta:NonNegativeInteger">
<xs:annotation>
  <xs:appinfo>
    <usm-meta:NewElementFor>usm-core:BinaryRelationship
       </usm-meta:NewElementFor>
    </xs:appinfo>
    <xs:documentation>An integer value indicating the ordering of members in a relationship. This element
 extends all semantics of BinaryRelationship.
    </xs:documentation>
  </xs:annotation>
</xs:element>
```

The usm-metadata-200907.xsd file specifies the semantics and formats of USM meta-information. You define meta-data using the following techniques:

- As a global attribute, such as isReserved that indicates an element is set by the infrastructure
- As type data, such as MultiValuedString that indicates a string property holds a list of data values; the meta-data defines the ordering of the values in the list and their delimiters
- As root elements for separate XML instance files such as the usm-correl-query.xml file

The xs:appinfo element of the type definition holds meta-data that is defined as type data. An example is the usm-exts:Order element as indicated in the example.

The usm-metrics-200907.xsd file specifies the format of metrics that you can collect from domain managers.

XML Instance Files

The following XML instance files are in the USM schema:

- usm-query
- usm-openenums
- usm-infradefaults

The usm-query file defines meta-data specifying type elements that you can use in a query. For performance queries, the CA Catalyst infrastructure assumes you can query some elements, but not all (some properties cannot be queried to select particular instances returned). Elements you can query are specified as QueriableElements in the usm-correl-query file.

The usm-openenums.xml file defines the values for enumerations that are expected to evolve over time and are hierarchical. These types of enumerations are difficult to define as opposed to static enumerations, which are not expected to change. Static enumerations are defined within an XSD, using the xs:restriction of a string and the xs:enumeration facet. However, taking this approach means that any change to the enumeration is a revision of the schema. Schema revisions that update one enumeration value are not desirable.

Using xs:enumeration works well when defining closed sets of values such as the days of the week. It does not work well when defining value lists for country, vendor, or operating system names, which change over time, sometimes daily.

Static (closed) enumerations are an exception. Most enumerations are open (evolving). Enumerations can constantly grow, be subject to interpretation or perspective, or can be hierarchical. Another example of an open enumeration is a US region value list that specifies a country by quadrants, versus a different customers perspective that defines smaller groupings of selected states. USM can define enumerations in an XML instance file where a propertys value can be checked against the instances in the usm-openenums file. An error or warning (depending on the needs of the application)

can report when a new value is encountered and alert needs to spell-check, correct the value, or add to the enumeration meta-data.

The following illustration shows an example of an open enumeration from the usm-openenums.xml file for the AlertTypeEnum:

```
<usm-meta:EnumDetails enumName="AlertTypeEnum">
  <usm-meta:Documentation>A categorization of Alerts along the lines of risk, quality, compliance
  and cost.</usm-meta:Documentation>
  <usm-meta:HierarchicalValue value="Risk">
    <usm-meta:Description>Indicates that the Alert deals with errors, exceptions, changes in state, etc.
    such as out of memory conditions or application exceptions.</usm-meta:Description>
    <usm-meta:HierarchicalValue value="Risk-Capacity"/>
    <usm-meta:HierarchicalValue value="Risk-Change"/>
    <usm-meta:HierarchicalValue value="Risk-Fault"/>
    <usm-meta:HierarchicalValue value="Risk-Security"/>
    <usm-meta:HierarchicalValue value="Risk-Performance"/>
    <usm-meta:HierarchicalValue value="Risk-RootCause"/>
  </usm-meta:HierarchicalValue>
  <usm-meta:HierarchicalValue value="Quality">
    <usm-meta:Description>Indicates that the Alert deals with perceived quality by the user/consumer.
    An example of this type of Alert is an SLA violation based on time to respond to a query.
    </usm-meta:Description>
  </usm-meta:HierarchicalValue>
  <usm-meta:HierarchicalValue value="Compliance"/>
  <usm-meta:HierarchicalValue value="Cost">
    <usm-meta:Description>Indicates that the Alert deals with cost - either increased or decreased.
    An example of this type of Alert is a necessary switch to a different cloud provider that results in
    a change in cost.</usm-meta:Description>
  </usm-meta:HierarchicalValue>
</usm-meta:EnumDetails>
```

The CA Catalyst infrastructure and CA Technologies products require specific data. The USM schema is not directly tagged with this data as it is stored in a separate XML instance file called usm-infraadefaults.xml (infrastructure defaults).

WSDL Files

The usm-metrics-200907.wsdl file specifies the operations that can be invoked against USM instances.

The following are valid operations:

- **GetAvailableMetrics**
  Returns a list of metrics for a specified instance. The list is an aggregation of the metric data from all products that have published an instance that was correlated.
- **GetMetricsValues**
  Returns the values of one or more requested metrics for a specified instance. The data returned can be live data or historical (time-stamped).

As metrics are complex, USM includes basic metric operations.

Other USM Files

The following information is available from the CA Technologies USM-Catalyst Global User Community page:

- **USM Metrics Dictionary**
  A list of the key metrics and performance indicators available from the CA Technologies products for many USM types. This dictionary is an Excel spreadsheet.
- **USM Schema Documentation**
  An HTML document defining the USM XML Schema constructs.
- **USM Schema Overview**
  A Word document providing an overview of the USM schema. The document also includes information about the structure, goals, and components of the schema and how it was designed.
- **USM Areas of Work**
  A presentation on the current areas of work for future USM extensions.
- **USM Schema Files**
  A compressed file including the schema XSD, XML, and WSDL files.

Advanced USM Features (Additional Information)

Contents

Examples of Supported Custom Operations

This section includes two examples that list the supported custom operations:

- Example 1 lists the custom operations that the BMC Atrium/Remedy connector supports
- Example 2 lists the custom operations that the IBM Tivoli Service Request Manager connector supports

Example 1 Custom Operations Supported by the BMC Atrium Remedy Connector

The BMC Atrium/Remedy connector supports the following custom operations as part of the Incident Profile, which contains a common definition for incident management for use across incident management products:

- CreateIncident
- UpdateIncident
- GetComments
- GetAttachments
- GetIncidents

> **NOTE**
> For more information about the BMC Atrium/Remedy connector and how these custom operations work, see the *BMC Atrium/Remedy Connector Guide*.

Example 2 Custom Operations Supported by the IBM TSRM Connector

The IBM Tivoli Service Request Manager connector supports the following custom operations as part of the Incident Profile, which contains a common definition for incident management for use across incident management products:

- CreateIncident
- UpdateIncident
- GetComments
- GetIncidents

Note: For more information about the IBM Tivoli Service Request Manager connector and how these custom operations work, see the *IBM Tivoli Service Request Manager Connector Guide*.

Examples of Supported Metrics

This section includes two examples that list the supported metrics:

- Example 1 lists the metrics that the BMC Atrium/Remedy connector supports
- Example 2 lists the metrics that the IBM Tivoli Service Request Manager connector supports

Example 1 Metrics Derived by the BMC Atrium Remedy Connector

The BMC Atrium/Remedy connector derives the following metrics from imported data:

- AvgNumberOfIncidentsPerUser
- AvgResolutionTimeForPriorityXxx IncidentsInMinutes
- AvgResponseTimeOfPriorityXxx IncidentsInMinutes
- PercentMajorIncidents
- NumberOfIncidentsCreated
- UnassignedIncidents

> **NOTE**
> For more information about the BMC Atrium/Remedy connector, see the *BMC Atrium/Remedy Connector Guide*.

Example 2 Metrics Derived by the IBM TSRM Connector

The IBM Tivoli Service Request Manager connector derives the following metrics from imported data:

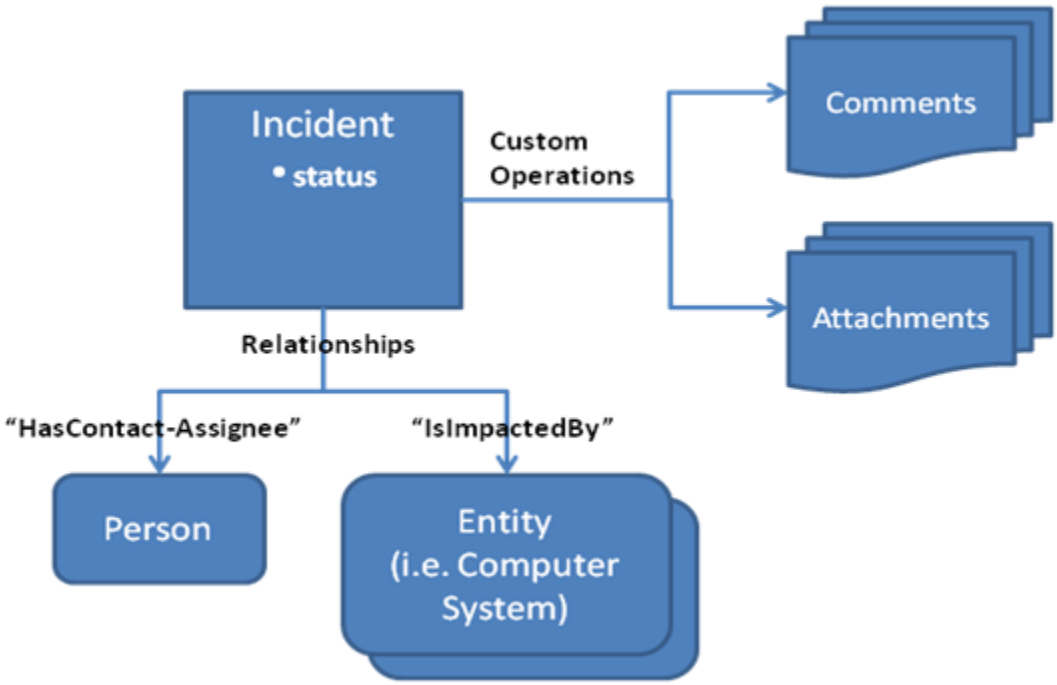
- AvgNumberOfIncidentsPerUser
- PercentMajorIncidents
- NumberOfIncidentsCreated
- UnassignedIncidents
- AvgDailyIncidentsPerAgent

> **NOTE**
> For more information about the IBM Tivoli Service Request Manager connector, see the *IBM Tivoli Service Request Manager Connector Guide*.

Incident Profile

The Incident Profile contains a common definition for incident management that you can apply to connectors, CA Technologies and third-party incident management products. A CA Catalyst connector that complies with the Incident Profile supports the CI types, custom operations, query filters, and metrics defined in the profile.

The following illustration shows the Object Model for incidents:



Incident objects are accessible using the query custom operation, with optional filters. Incident comments and attachments are accessible through custom operations; the related persons and impacted entities are exposed with relationships. Metrics about incidents are accessible through the USM metrics operations.

A CA Catalyst connector indicates support for the Incident Profile by returning the Incident Profile namespace value using the ConnectorDescriptor.getManagementCats() interface.

USM Types

A connector that is compliant with the Incident Profile supports the USM Incident, Person, and any impacted entity types (for example, ComputerSystem, Router, and so on).

The connector enumerates its supported CI types using the ConnectorDescriptor.getManagedEntityTypes() interface.

Supported Custom Operations

The Incident Profile supports the following custom operations:

- CreateIncident
- UpdateIncident
- GetComments
- GetAttachments
- GetIncidents

Supported Metrics

The Incident Profile supports the following metrics:

> **NOTE**
> For more information about the USM metrics dictionary, see the USM documentation.

- AvgResolutionTimeForPriorityXxx IncidentsInMinutes
- PercentPriXxxIncidentsResolved OnTime
- AvgResponseTimeOfPriorityXxx IncidentsInMinutes
- PercentPriorityXxxIncidents RespondedOnTime
- AvgNumberOfIncidentsPerUser
- PercentMajorIncidents
- NumberOfInstallIncidents
- NumberOfLearnabilityIncidents
- NumberOfOperationIncidents
- NumberOfUnderstandability Incidents
- PercentIncidentsResolvedInTarget
- PercentXxxIncidentsResolvedInTarget
- AvgIncidentCreateToResolveInMinutes
- AvgXxxIncidentCreateToResolveInMinutes
- PercentIncidentsAutoGenerated
- PercentIncidentsDueToChanges
- PercentIncidentsDueToDataIntegrity
- PercentIncidentsDueToVirus
- PercentIncidentsDispatched
- PercentIncidentsDueToLackOfTraining
- AvgDailyIncidentsPerAgent
- NumberOfIncidentsCreated
- UnassignedIncidents
- IncidentSLACount

Connector Identification Numbers

The USM schema assigns an ID value to each connector defined by enumerations of the MdrProduct property. These five-digit values appear as a part of the connector (or CA SOI plugin) name in the CA SOI interfaces. The following table shows the MdrProduct values for all available connectors and plugins:

NOTE
To find values for connectors not listed on this table, see the USM schema documentation.

For more information about plugins, see CA SOI Plugins.

MdrProduct	Connector/CA SOI Plugin
CA:00001	CA Application Performance Management
CA:00002	CA eHealth Performance Manager
CA:00004	CA SOI
CA:00005	CA Spectrum Infrastructure Manager
CA:00006	Automation Manager Platform (includes CA Spectrum Automation Manager, CA Virtual Assurance, and CA Virtual Automation)
CA:00007	CA Access Control
CA:00013	CA SiteMinder
CA:00015	CA Workload Automation AE
CA:00018	CA Clarity PPM
CA:00019	CA Service Catalog

CA:00020	CA Service Desk Manager/CA CMDB
CA:00022	CA IT Client Manager
CA:00030	CA Catalyst
CA:00031	Microsoft SCOM
CA:00033	CA Application Configuration Manager
CA:00034	CA NetQoS Performance Center
CA:00035	CA SYSVIEW Performance Manager
CA:00036	Generic SNMP traps
CA:00037	Microsoft Windows Event Log
CA:00038	HP Business Availability Center
CA:00039	Text log files
CA:00040	CA OPS-MVS Event Management and Automation
CA:00041	Web services events
CA:00042	IBM Tivoli Netcool
CA:00043	IBM Tivoli Enterprise Console
CA:00044	IBM Service Request Manager
CA:00045	BMC Atrium
CA:00046	BMC Remedy
CA:00047	CA SOI Modeler
CA:00048	CA Insight Database Performance Manager
CA:00050	CA UIM connector
CA:00051	BMC Remedy and BMC Atrium connector
CA:00056	Service Discovery plugin
CA: 00062	CA SOI Domain connector plugin
CA:09993	Mid-tier connector plugin
CA:09995	Manual CA Catalyst Persistence Store updates
CA:09996	CA SOI web services
CA:09997	Universal Connector Plugin
CA:09998	Sample connector
CA:09999	DB Test Application

The MdrProduct values from CA:00035 to CA:00041 and CA:00052 define sources sent from the Event connector.

The CA SOI MdrProduct values define CA SOI updates as follows:

- **CA:00004**
  Defines CIs published by CA SOI into the Persistence Store, through migration or otherwise.
- **CA:00047**
  Defines CIs manually created from the Service Modeler.
- **CA:09995**
  Defines a manual update of the Persistence Store.
- **CA:09996**
  Defines CIs created using CA SOI web services through the USM Web View.

Generic Connector Documentation

This section provides documentation for generic connectors provided with CA SOI for configuring custom or generic integrations.

Domain Connector

The CA SOI Domain connector enables a connection between a CA SOI deployment and an enterprise CA SOI layer. Each tier consists of a full CA SOI deployment, with the top tier named the Enterprise SA Manager.

The CA SOI Domain connectors provide the following information from a local source SA Manager to the remote Enterprise SA Manager:

- Service CIs and their associated metrics like health, severity, or impact
- Service alerts associated with the imported service CIs that include the quality and risk information
- Changes to services and their alerts, including additions, updates, state changes, and deletions
- Customer information that is associated with services
- Managed CIs associated with alerts in the configured queue
- Alerts in the configured queue
- For Southbound functionality, the alerts that are cleared in Enterprise SA Manager are cleared in Domain SA Manager

You can control the services that appear in the Enterprise SA Manager.

The CA SOI Domain connector forwards the following information from its source SA Manager to the Enterprise SA Manager for analysis in a unified view:

- Service CIs
- Service alerts
- Infrastructure Alerts
- Customer information that is associated with services
- Managed CIs
  The managed CIs or sub-services associated with alerts in the configured queue are published from Domain SA Manager to Enterprise SA Manager. Also, the name of the alerted managed CIs or sub-services appears in the Name column for Alerts in Operations Console of Enterprise SA Manager. The following CI properties are published from Domain SA Manager to Enterprise SA Manager:
- – MdrProduct
  – MdrProdInstance
  – MdrElementID
  – Label
  – ClassName
  – CIReverseURL
  – Description
  – CreationTimestamp
  – LastModTimestamp
  – LastModActivity
  – AdministrativeStatus
  – IsInMaintenance
  – Vendor
  – Custom or User Attribute:
    - CIuserAttribute1 through

CIuserAttribute10
- One or more correlatable properties of each CI Type
- Other necessary properties if any for each CI Type

To publish the managed CIs or sub-services from Domain SA Manager to Enterprise SA Manager, configure **ssaDomain_policy.**xml file.

NOTE

For more information about how to configure the policy file, see Post-Installation Steps.

The CA SOI Domain connector subscribes to changes so that it can automatically forward new services, new alerts in configured queue, and new CIs associated with the alerts or changes to the existing services and alerts to the Enterprise SA Manager. The connector also exposes service metric information for consuming applications.

Install the CA SOI Domain Connector

You install the CA SOI Domain connectors on the Enterprise SA Manager. The Domain connectors obtain data from the different Domain SA Managers and publish to the Enterprise SA Manager through the MQ Bus. Install as many CA SOI Domain connectors as required to tier your deployment to the desired level. Consider the following items during the CA SOI Domain connector deployment phase:

- Multiple CA SOI Domain connectors can connect to one Enterprise SA Manager, but only one CA SOI Domain connector can connect to each Domain SA Manager.
- In firewall environments, the MQ Server port (61616 by default) must be open between the CA SOI Domain connector and the Enterprise SA Manager. Also, the SA Manager port (7090 by default) must be open between the CA SOI Domain connector and the Domain SA Manager.
- The source SA Manager web services host information for the CA SOI Domain connector is stored in the SOI_HOME \resources\Configurations\ssaDomain_*MdrProdInstance.*xml file.

Follow these steps:

1. Run Connector_Domain.exe from the Disk1\SOI folder of the CA SOI installation image.
2. Accept all license agreements, and click Next.
3. Enter the appropriate information in the following fields to configure a web services connection to the domain manager:
   - **SOI Manager Hostname**
     Specifies the name of the domain manager host from which to pull data for use by the remote source SA Manager.
   - **SOI Manager Web Services Port**
     Specifies the port in which to connect to web services on the source SA Manager.
     Default: 7090
   - **Web Service User**
     Specifies the user that has access to the CA SOI web services on the source SA Manager.
     Default: samuser
   - **Web Service Password**
     Specifies a valid password for the Web Service User
   - **Validate SOI Manager**
     Specifies whether to verify the web services configuration information when you click Next.
4. Enter the appropriate information to configure the UI Server settings of the domain manager. Change these values only if the UI Server of the domain manager is remote to the domain manager. The field values allow a proper launch back to the domain manager Operations Console from the Enterprise SA Manager:
   - **Domain Manager UI Server Host**
     Specifies the host on which the source SA Manager's UI Server resides.
     Default: domain manager host
   - **Domain Manager UI Server Port**
     Specifies the source SA Manager's UI Server port.

Default: 7070

5. Specify whether to start the services after installation completes, and click Next.
6. Review your selections, and click Install.
   The CA SOI Domain connector installs on the system and integrates with the specified source SA Manager and Enterprise SA Manager. An installation summary page opens when the installation finishes.
7. Click Done.
   If the installation summary page notes installation errors, view the Domain connector installation log file that is located in the SOI_HOME\log folder to troubleshoot the installation. This file is created when you click Done after the installation finishes.

   NOTE
   You can install another instance of the domain connector to point to an additional SA Manager. Run the Domain Connector installation again and a dialog prompts you to configure an additional instance.

Post-Installation Steps

To publish the managed CIs or sub-services associated with alerts in the configured queue from Domain SA Manager to Enterprise SA Manager, configure the **ssaDomain_policy.xml**policy file.

Follow these steps:

1. Add the following attribute mappings in **<IFW-HOME>\resources\Core\Catalogpolicy\ssaDomain_policy.xml** file.
   a. For CI User Attributes, add the following tags after existing tags of *<Format>* which is available under *<EventClass name="Item">* tag:
      - <Field conditional='ssa_user_attr_1' output='CIuserAttribute1' format='{0}' input='ssa_user_attr_1' />
      - <Field conditional='ssa_user_attr_2' output='CIuserAttribute2' format='{0}' input='ssa_user_attr_2' />
      - <Field conditional='ssa_user_attr_3' output='CIuserAttribute3' format='{0}' input='ssa_user_attr_3' />
      - <Field conditional='ssa_user_attr_4' output='CIuserAttribute4' format='{0}' input='ssa_user_attr_4' />
      - <Field conditional='ssa_user_attr_5' output='CIuserAttribute5' format='{0}' input='ssa_user_attr_5' />
      - <Field conditional='ssa_user_attr_6' output='CIuserAttribute6' format='{0}' input='ssa_user_attr_6' />
      - <Field conditional='ssa_user_attr_7' output='CIuserAttribute7' format='{0}' input='ssa_user_attr_7' />
      - <Field conditional='ssa_user_attr_8' output='CIuserAttribute8' format='{0}' input='ssa_user_attr_8' />
      - <Field conditional='ssa_user_attr_9' output='CIuserAttribute9' format='{0}' input='ssa_user_attr_9' />
      - <Field conditional='ssa_user_attr_10' output='CIuserAttribute10' format='{0}' input='ssa_user_attr_10' />

      For example,

```
<EventClass name="Item">
.
.
.

  <Format>
    <Field output="CAProductIdentifier" input="" format="00004"/>
    <Field output="outfile" input="CAProductIdentifier" format="CA{0}_policy"/>
    <Field conditional='ssa_user_attr_1' output='CIuserAttribute1' format='{0}'
 input='ssa_user_attr_1' />

    .
    .
    <Field conditional='ssa_user_attr_10' output='CIuserAttribute10' format='{0}'
 input='ssa_user_attr_10' />
```

```
.
.
.
  </Format>
```

b. For Alert User Attributes, add the following tags after existing tags of *<Format>* which is available under *<EventClass name="Alert">* tag:
   - <Field conditional='ssa_userattribute_1' output='userAttribute1' format='{0}' input='ssa_userattribute_1' />
   - <Field conditional='ssa_userattribute_2' output='userAttribute2' format='{0}' input='ssa_userattribute_2' />
   - <Field conditional='ssa_userattribute_3' output='userAttribute3' format='{0}' input='ssa_userattribute_3' />
   - <Field conditional='ssa_userattribute_4' output='userAttribute4' format='{0}' input='ssa_userattribute_4' />
   - <Field conditional='ssa_userattribute_5' output='userAttribute5' format='{0}' input='ssa_userattribute_5' />
   - <Field conditional='ssa_userattribute_6' output='userAttribute6' format='{0}' input='ssa_userattribute_6' />
   - <Field conditional='ssa_userattribute_7' output='userAttribute7' format='{0}' input='ssa_userattribute_7' />
   - <Field conditional='ssa_userattribute_8' output='userAttribute8' format='{0}' input='ssa_userattribute_8' />
   - <Field conditional='ssa_userattribute_9' output='userAttribute9' format='{0}' input='ssa_userattribute_9' />
   - <Field conditional='ssa_userattribute_10' output='userAttribute10' format='{0}' input='ssa_userattribute_10' />

   For example,

```
<EventClass name="Alert">
.
.
.
  <Format>
    <Field output="CAProductIdentifier" input="" format="00004"/>
    <Field output="outfile" input="CAProductIdentifier" format="CA{0}_policy"/>
    <Field conditional='ssa_userattribute_1' output='userAttribute1' format='{0}'
 input='ssa_userattribute_1' />
    .
    .
    <Field conditional='ssa_userattribute_5' output='userAttribute5' format='{0}'
 input='ssa_userattribute_5' />
.
.
.
  </Format>
```

2. Restart **CA SAM Integration** Services.

Import Services into the Enterprise SA Manager

You import information from the source SA Manager into the Enterprise SA Manager in either of the following ways:

- Automatically to import all existing services including services added to the source SA Manager
- Manually to import a selection of services

Follow these steps:

1. Open the Operations Console on the Enterprise SA Manager and select Tools, Import Services.
2. Perform *one* of the following actions for each CA SOI Domain connector:
   – Select the CA SOI Domain connector entry, click Auto, and click OK.
     The connector imports all current service CIs and service alerts and automatically imports subsequent service CIs and alerts.
   – Select the CA SOI Domain connector entry and click Import if you want to import only a subset of the existing services.
3. (Manual import only) Select the services to import, move them to the right pane, and click OK.
   The connector imports only the services you selected. If you import manually, then you manually import services that you add to the source SA Manager in the future.
   The service CIs and alerts from the source SA Manager appear in the Enterprise SA Manager Operations Console.
4. (Optional) Right-click a service or service alert that was imported from the source SA Manager and select Launch, CA SOI Domain Manager.
   The source SA Manager Operations Console appears in the context of the selected item so that you can drill into the service for more information.

Only the top-level service CI and any associated service alerts appear for each imported service. You cannot modify services that the Domain connector imports in the Enterprise SA Manager by adding CIs.

To determine if services in the Enterprise SA Manager Operations Console belong to a source SA Manager:

1. Select the top-level Services node in the Services tab of the Navigation pane.
2. Select the Services tab in the Contents pane.

You can add a Tier column to the table on this tab that displays a value of Remote for all services originating in a source SA Manager. All services with a value of Remote in the Tier column cannot be modified in the Enterprise SA Manager. This includes changing the service name, the content of the model, maintenance mode, or any other aspect of the service that is typically modifiable from the Operations Console.

Domain Connector Properties

The following Domain connector properties are available in the CA SOI Administration tab after you install the connector. To view the properties, open the Dashboard and click the Administration tab, and navigate to the Administration, Connector Configuration, *connector_server* folder:

- **alertQueueAsFilterForEnterprise**
  Specifies the name of the alert queue that the domain connector sends to SOI enterprise manager. To specify multiple alert queues, separate the options by comma.
  Default: "-" (hyphen)
- **allServiceAlerts**
  Defines whether to send all service alerts that are generated in the domain level to SOI enterprise manager or not.
  Default: true
- **RetryCount**
  Specifies the number of tries the CA SOI Domain connector gets to connect to the source SA Manager.
- **RetryInterval**
  Specifies the interval (in seconds) the CA SOI Domain connector waits before it tries to reconnect to the source SA Manager.
- **host**
  Specifies the name of the source SA Manager from where you want to collect information.
- **mockMode**
  (Only for debugging purposes) Specifies that the parameter only logs the CA SOI Domain connector behavior.
- **password**

Specifies the password that is associated with the username.

- **pollCustomers**
  Specifies the interval (in seconds) the CA SOI Domain connector waits before it retries to poll for the updated customer information.
- **port**
  Specifies the web services port CA SOI uses to connect to the source SA Manager.
- **username**
  Specifies the web services user name that is required to connect to the source SA Manager.
- **wsTimeout**
  Specifies the web services time-out value (in seconds).
- **batchSize**
  Specifies the number of CIs (associated with the managed alerts) that are fetched in one web service request during startup of domain manager.
  Default: 100

Launch In Context Details

- **Label**
  Specifies the label that is used to support the launch in context from the Enterprise SA Manager to the source SA Manager.
- **Port**
  Specifies the port number that is used to support the launch in context from the Enterprise SA Manager to the source SA Manager.
- **Protocol**
  Specifies the protocol to use when launching a URL to the Enterprise Domain Manager.
- **Type (Read Only)**
  Specifies the Item type to which the Launch in Context Details apply.
- **Host**
  Specifies the Enterprise Domain Manager server.
- **seqNum (Read Only)**
  Internal use only.
  Note: If you add an LIC rule to a configuration file, a unique seqNum must be given.
- **url**
  Specifies the domain managers launch in context URL for each Type.

Enable Southbound Synchronization

This article describes how to enable southbound synchronization for domain connector in Enterprise SA Manager. This functionality clears the alert in domain SA Manager when the alerts are cleared in Enterprise SA Manager.

Follow these steps:

1. From the UI Server page, click **Administration** tab.
2. Expand the **CA Service Operations Insight Manager Configuration** and click **Synchronization Configuration**.
3. Enable the **Alert Synchronization**.
4. Expand **Connector Configuration**, and select the Domain Connector service.
5. Select **isRemotable** option.
   You can also set the option in the domain connector config file by setting the value to 1.
6. Restart **Enterprise SOI Manager service**.

The outbound synchronization for domain connector in Enterprise SA Manager is enabled.

Convert Unmanged CI to Managed CI

This section describes how to convert unmanaged CI to managed CI automatically through escalation policy.

Follow these steps:

Step 1: Configure UnamangedToManagedCI.properties file

1. Navigate to **<SOI_Home>\tomcat\bin** folder.
2. Open **UnmanagedToManagedCI.properties** file and enter the values for the properties as follows:
   a. relationship.mdrProduct=<Connector Id against which relationship between service and unmanaged CI should be published>
   Example: relationship.mdrProduct=CA:09997
   b. relationship.mdrProdInstance=<Connector name (MdrProdInstance attribute value from Connector Config file)>
   Example: relationship.mdrProdInstance=DC-TEST-05.ca.com
   c. relationship.connector.containerName=<Container name (ConnectorName attribute value from IFW Config file)>
   Example: relationship.connector.containerName=DC-TEST-05.ca.com
   d. relationship.semantic=<Relationship to be created between Service and CI to be managed>
   Example: relationship.semantic=HasMember
   You can note the relationship name from Service Modeller or can Create Service editor.
   e. relationship.scopeMdrElementId=<MdrElement ID property value against reconciledSheet projection, for service under which CI should be added, from USM Notebook tab>
   Example: relationship.scopeMdrElementId=eade234b2eb44bb1b8382b34b564e375
   f. relationship.scopeSOIProjectionMdrElementId=<MdrElement ID property value against CA:00047 (SOI) projection, for service under which CI should be added, from USM Notebook tab>
   Example: relationship.scopeSOIProjectionMdrElementId=0x10000000000003

   NOTE

   Ensure that the **scopeMdrElementId** and **scopeSOIProjectionMdrElementId** are MdrElementIDs are of same service.

Step 2: Create Escalation Action

1. Navigate to **Tools**, **Escalation Policies and Actions**, and **Actions tab** on the Operation Console.
2. Click **Add**, and enter the following details:
   a. **Action name*:** specifies the action name.
   b. **Action Type*:** Select Execute Command.
   c. **Command box*:** enter the following command in the Command text box.

```
<SOI_HOME>\tomcat\bin\UnmanagedToManagedCI.bat -h<mgr host>:<mgr port> -u<SOI
 Username> -p<encrypted password> -a$[Alert ID with domain ID]
```

   Where,
   - **<SOI_HOME>** specifies the file path where CA SOI Manager is installed.
   - **<mgr host>** specifies the name of the CA SOI Manager Host.
   - **<mgr port>** specifies the port number for CA SOI Manager.
   - **<SOI Username>** specifies the user name of CA SOI administrator or operator.
   - **<encrypted password>** specifies the encrypted login password for the CA SOI username. You can encrypt the password using WSSamEncryptCmd.bat utility that is located in the tomcat\bin folder.

   Example: C:\Program Files (x86)\CA\SOI\tomcat\bin\UnmanagedToManagedCI.bat -hdc-test-01:7090 -usamuser -pEJC6HRY79+PZirsnrwXJgpG6W0N3emgFQMVCfhvIbuy8 -a$[Alert ID with domain ID]
3. Click **OK**.

Step 3: Create Escalation Policy

1. Open **Alert Escalation Policy Editor**, navigate to **attributes tab**, and define the criteria:
   a. **Attribute:** Unmanaged, **Comparision Type:** Equal To, and **Attributes Value:** Yes, and click **Add.**
   b. **Attribute:** Class, **Comparision Type:** Not Equal To**,** and **Attributes Value:** Entity, and click **Add**.
2. Navigate to **Policy Actions** tab, select the **Escalation Action** that you have created in step 2, and click **OK**.
3. Navigate to **Alert Queue Assignment** tab, and assign policy to Alert Queues. The corresponding CI of the alert is managed when the alert satisfies the two criteria that are mentioned in step a and step b.
4. Click **OK**.

Universal Connector

CA SOI provides a Universal connector in the IFW that can retrieve services, CIs, and events from various CA Technologies and third-party products. The Universal connector provides command line and web services programming interfaces that third-party products can use to publish new services, CIs, and events to the web service, which is polled periodically for these events. These events are normalized to the USM format and made available to the SA Manager.

The Universal connector is appropriate for Level 1 integrations with CA SOI provided any of the following conditions are met:

- The domain manager has a script interface that calls the Universal connector command-line interface to create CIs and alerts in CA SOI.
- The domain manager has a programming interface that the Universal connector web services interface can use to create CIs and alerts in CA SOI.
- Custom events are forwarded to CA SOI.

The Universal connector only forwards events to the SA Manager, which indicates the following:

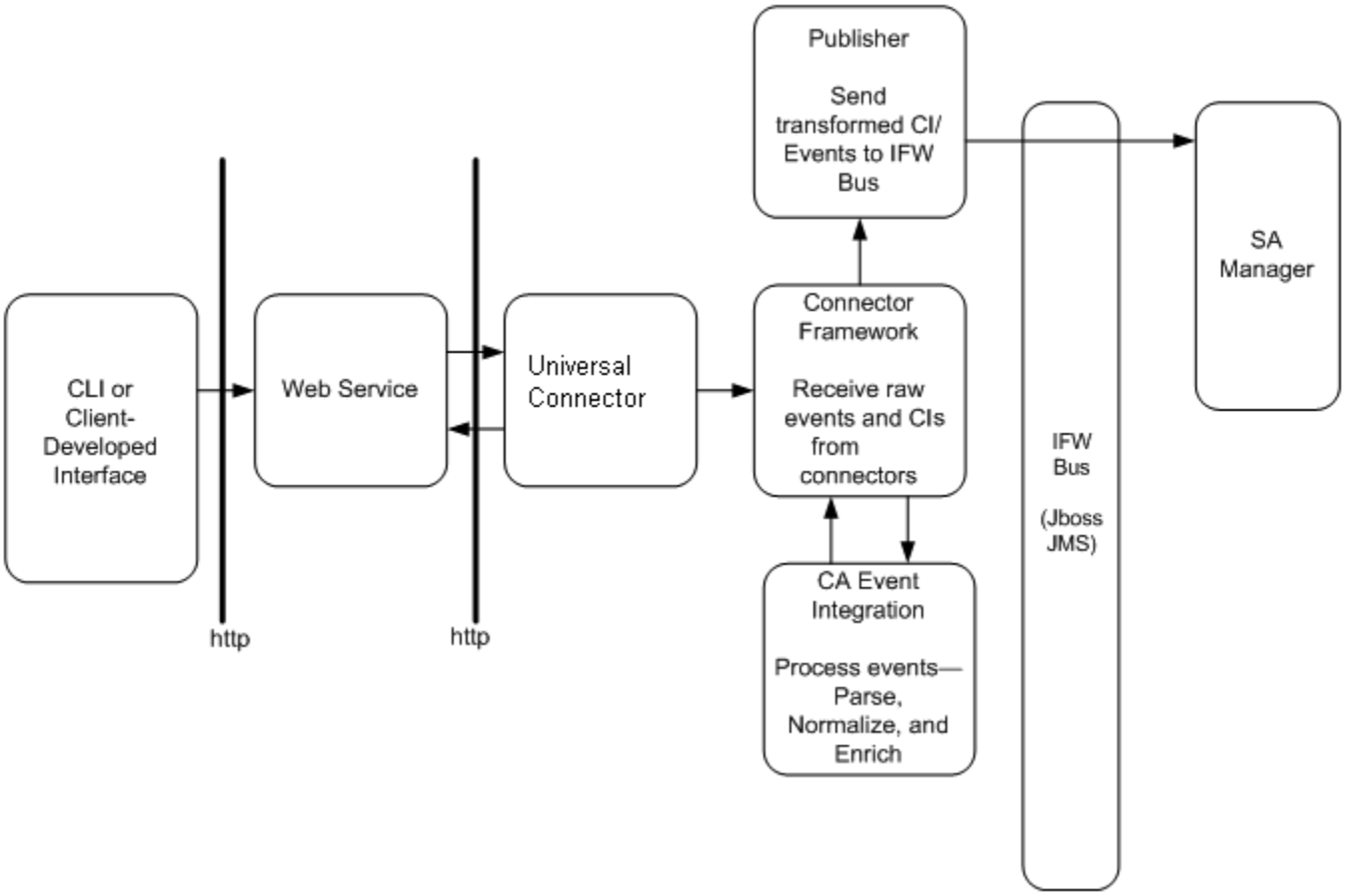
- The integrated application must call the Universal connector interface.
- It is not possible to get information from the integrated application. For example, you cannot import services from the Universal connector in the Operations Console.
- Because CA SOI cannot get the current status for a CI, the status for the Universal connector CIs might not be current after a CA SOI server restart or in newly created service models.

Universal Connector Components

The Universal connector consists of the following main components:

- **Connector**
  Polls the web service periodically for services, CIs, and events; normalizes these items to the USM format; and makes them available to the SA Manager. The connector is part of the IFW.
- **Web Service**
  Lets various CA Technologies and third-party products send services, CIs, and events for the Universal connector to collect and publish to CA SOI. The web service runs on a Tomcat server that by default is on the SA Manager, is named GenericConnectorService, and is located at *http://<SA-Manager>:7090/axis2/services/GenericConnectorService?wsdl*. The server polls for the events that the web service client forwards.
- **Web Service Client**
  Communicates with the web service using the web service interface. It provides a command-line interface (GCEventAddCmd.bat) that implements various web services methods to load services, CIs, and events into the web service. The web service client also provides supporting libraries for user programs to use the web services interface. This component is installed when you select the Universal connector installation option in the CA SOI installation wizard.

The following illustration shows the Universal connector process flow:



Configure the Universal Connector

When you install the Universal connector, it is automatically enabled with default settings. You can change the default settings and controls.

Follow these steps:

1. Access the Dashboard, select the Administration tab, and select the Connector Configuration option.
2. Select the Universal connector in the Connector column.
   The Universal connector details page opens and displays a number of tables with more details about the connector and the server where it is installed.
3. (Optional) Change any of the connector controls in the Connector Controls table and click Save.
4. Click any of the following Value fields in the Connector Details table and click Save:
   – **host**
     Indicates the host name for the Tomcat application server where the web service is running.
   – **port**
     Indicates the port number for the Tomcat application server where the web service is running.
   – **poll interval**
     Specifies the number of seconds to wait between polling for events.
5. Click Stop and wait until the connector status changes to Offline.
6. Click Start and wait until the connector status changes to Online.

NOTE
It may take a few minutes before the connector status displays Online.

WARNING
Do not perform rapid start and stop operations on the connector. Each stop and start sends the corresponding command to the connector. Rapid start and stop operations from the interface can cause these commands to queue on the connector and cause the connector to start and stop until all commands in the queue are processed.

Universal Connector Programming Interface

Contents

You can access the Universal connector web client at the program level. This is especially useful if source control is available for your application and you want to provide a deeper level of integration with CA SOI. To successfully use the programming interface, you must first understand the Universal connector web service methods.

Universal Connector Web Service

The Universal connector provides a one-way interface using web services method calls that let CA Technologies and third-party products and clients send events to CA SOI. These events create new services that can include the following:

- CIs and relationships
- Create, update, and delete CIs
- Add infrastructure alerts using events

The Web Services Descriptor Language (WSDL) for the Universal connector web service is located at http://<SA Manager>:7090/axis2/services/GenericConnectorService?wsdl.

The Universal connector installation downloads all jar files into the SOI_HOME\lib\generic directory that are required to build Java-based web services clients. The file ifw.connectors.universal-4.2.0-client.jar contains all the web services stub classes that the method implements.

The Java package that contains all of the web service Java class definitions is com.ca.sam.ifw.connector.generic.webservice.

Item Class

The Item class creates a Java object containing the following:

- Attributes of each event that you want to pass to the addEvents() method
- Attributes of each CI that you want to pass to the methods addCIs(), updateCIs(), or deleteCIs()
- Attributes of each service that you want to pass to the method addServices()
- Attributes of each service relationship object that you want to pass to the method addServiceRelationships()
- Attributes of each service CI object that you want to pass to the method addServiceCIs()

The Item class is as follows:

```
com.ca.sam.ifw.connector.generic.webservice.Item
public class Item
extends java.lang.Object
```

The constructor summary for the Item class is as follows:

```
Item()
```

The method summary for the Item class is as follows:

- String getClassName()
- void setClassName(String className)
- String getEventtype()
- void setEventtype(String eventtype)
- void setCiProperties(CIProperty[] ciProperties)
- CIProperty[] getCiProperties()

addEvents() Method

The addEvents() method adds an array of events to the web services queue. The Universal connector collects these events for publishing to CA SOI.

```
public boolean addEvents(Item[] statusEvents)
```

throws:

```
org.apache.axis2.AxisFault
```

updateCIs() Method

The updateCIs() method creates a list of CIs with updated attributes that are added to the web services queue. The Universal connector collects the CIs for publishing to CA SOI.

```
public boolean updateCIs(Item[] ciUpdates)
```

throws:

```
org.apache.axis2.AxisFault
```

addCIs() Method

The addCIs() method creates a list of new CIs that are added to the web services queue. The Universal connector collects these items for publishing to CA SOI.

```
public boolean addCIs(Item[] ciAdds)
```

throws:

```
org.apache.axis2.AxisFault
```

deleteCIs() Method

The deleteCIs() method creates a list of CIs that you want to delete from CA SOI and add to the web services queue. The Universal connector collects these items for publishing to CA SOI.

```
public boolean deleteCIs(Item[] ciDeletes)
```

throws:

```
org.apache.axis2.AxisFault
```

addServices() Method

The addServices() method creates a list of new services that are added to the web services queue. The Universal connector collects these items for publishing to CA SOI.

```
public boolean addServices(Item[] services)
```

throws:

```
org.apache.axis2.AxisFault
```

addServiceRelationships() Method

The addServiceRelationships() method creates a list of new service relationships that are added to the web services queue. The Universal connector collects these items for publishing to CA SOI.

```
public boolean addServiceRelationships(Item[] relationships)
```

throws:

```
org.apache.axis2.AxisFault
```

addServiceCIs() Method

The addServiceCIs() method creates a list of new service CIs that are added to the web services queue. The Universal connector collects these items for publishing to CA SOI.

```
public boolean addServiceCIs(Item[] cis)
```

throws:

```
org.apache.axis2.AxisFault
```

Universal Connector Command Line Interface

Contents

Use the GCEventAddCmd batch program (GCEventAddCmd.bat) to manage the Universal connector from a command line. All appropriate files (such as GCEventAddCmd batch file, sample XML files) are installed in the CA SOI root directory, for example C:\Program Files\CA\SOI\generic-client. The batch program lets you create services, CIs, relationships, and alerts in CA SOI from any program that is able to run commands.

> **NOTE**
> Typically, the CA SOI root directory is not in the system search path (Windows %PATH% environment). Therefore, prefix the CA SOI directory, such as C:\Program Files\CA\SOI\generic-client, while running GCEventAddCmd.bat.

You can run the GCEventAddCmd batch program based on the following scenarios:

- For publishing a single event, we recommend that you run the batch program using the appropriate command line parameters. However, if you want, you can also use the XML file as described in the next point.
- For publishing multiple events, CIs, and services, run the batch program using the appropriate XML file. In this case, parameters are stored in the appropriate XML files.

Publish a Single Status Event

The batch program can publish a single event to CA SOI when specified with the appropriate command line parameters. For a single event, you can pass most of the parameters in the argument list. In this case, run the batch program as follows:

```
GCEventAddCmd -h<wsHostName:wsPort> -a<MdrElementID> -i<AlertedMdrElementID> -s<severity> -t<AlertType> -
m<summary>
```

- **-h*<wsHostName:wsPort>***

Specifies the web services host name and port number, which is the same as the SA Manager host name and port number. By default, the SA Manager port is 7090.

- **-a** ***<MdrElementID>***
  Specifies the unique alert ID.
- **-i*<AlertedMdrElementID>***
  Specifies the AlertedMdrElementID, which is the MdrElementID of the CI that the alert is associated with.
- **-s*<severity>***
  Specifies the alert severity. The valid values are Normal, Minor, Major, Critical, Down.
- **-t*<AlertType>***
  Specifies the type of alert. The valid values are Risk and Quality.
- **-m*<summary>***
  Specifies the appropriate message text that you want to display.

Publish Multiple Status Events, CIs, and Services

When the batch program is used with an XML file, the batch program can publish multiple events, CIs, and services. The parameters are stored in the XML file. The XML file acts as a source for the Universal connector.

In this case, run the batch program as follows:

```
GCEventAddCmd -h<wsHostName:wsPort> -f<XML file>
```

- **-h*<wsHostName:wsPort>***
  Specifies the web services host name and port number, which is the same as the SA Manager host name and port number. By default, the SA Manager port is 7090.
- **-f*<XML file>***
  Specifies the XML file name.

The following example shows how you can use an XML file (for example, Add_alerts.xml) to run the batch program:

```
GCEventAddCmd -hSOIserver.ca.com:7090 -fAdd_alerts.xml
```

eventType, MdrElementID, and className are the required USM properties for any XML file that you want to use. All other USM properties depend on the type of CI you are adding. Refer to the USM schema documentation to determine which USM properties are required for each class of CI. For example, the Service CI requires ServiceName and ServiceVersion.

> **NOTE**
> For more information about how to access the USM schema or how to use the USM Web View, see the How to Find USM Properties for a CI section.

To understand how to use and create your XML files, you can review the example scenarios provided in the Integration Scenario Examples section. You can also view the sample XML files available in the Sample XML Files section.

Integration Scenario Examples

Contents

This section provides example scenarios where you can use the GCEventAddCmd batch program to achieve specific tasks.

> **NOTE**
> If you want to know the required USM properties to include in a certain USM type (such as service, computer_system, application, and alert), check the USM schema documentation.

Create a Service

To create a service, run the GCEventAddCmd batch program with an XML file that contains the <Service> tag and appropriate property tags with corresponding values. The template tag structure is as follows:

```
<Services>
 <Service>
   ...
   ...
 </Service>
</Services>
```

> **NOTE**
> For a complete list of USM property tags, see the USM schema to determine which USM properties are required for this class of CI.

The following example lets you create a service called MyService by running the batch program as follows:

```
GCEventAddCmd -hSOIserver.ca.com:7090 -fMyService.xml
```

The example uses the MyService.xml file that includes the following information:

```
<Services>
 <Service>
   <property tag="eventType" value="AddService" />
   <property tag="MdrElementID" value="MyService" />
   <property tag="className" value="Service" />
   <property tag="ServiceName" value="MyService" />
   <property tag="ServiceVersion" value="0" />
   <property tag="Description" value="Adding a new Service" />
     <property tag="AdministrativeStatus" value="Managed" />
 </Service>
</Services>
```

The USM properties used in this example XML file are as follows:

- **eventType**
  Specifies the type of event. For this example, the value is AddService.
- **MdrElementID**
  Specifies a unique identifier for the service. For this example, the value is MyService.
- **className**
  Specifies the USM class name. For this example, the value is Service.
- **ServiceName**
  Specifies the name of the service that distinguishes it from others. For this example, the value is MyService.
- **ServiceVersion**
  Specifies the version level of the service. For this example, the value is 0.
- **Description**
  Specifies a detailed description of the service. For this example, the value is Adding a new Service.
- **AdministrativeStatus**
  Specifies the high-level status of the service. For this example, the value is Managed.

Create a CI

You can manage CIs in two different XML tag contexts<Events> tag and <Services> tag. The combination with other actions determines the XML tag context of your choice. For example, if you want to manage a CI along with a service, use

the <Services>...</Services> tags. If you want to create an event along with the CI creation, use the <Events>...</Events> tags.

- For Events tag, the structure to create a CI is as follows:

```
<Events>
 <Event>
   eventType = AddCIEvent
```

- For Services tag, the structure to create a CI is as follows:

```
<Services>
 <Service>
   eventType = AddCI
```

NOTE
For a complete list of USM property tags, see the USM schema to determine which USM properties are required for the specific class of CI.

The following example lets you create a CI called MyMachine of the USM type ComputerSystem by running the batch program as follows:

```
GCEventAddCmd -hSOIserver.ca.com:7090 -fCreateCI.xml
```

The example uses the CreateCI.xml file that includes the following information about the CI:

```
<Events>
 <Event>
   <property tag="eventType" value="AddCIEvent" />
   <property tag="MdrElementID" value="MyMachine" />
   <property tag="className" value="ComputerSystem" />
   <property tag="Label" value="MyMachine" />
   <property tag="Description" value="Description of a Universal Connector Entity" />
   <property tag="AdministrativeStatus" value="Managed" />
   <property tag="Vendor" value="Dell" />
   <property tag="PrimaryDnsName" value="MyMachine.ca.com" />
   <property tag="ComputerName" value="MyMachine" />
   <property tag="MemoryInGB" value="1024" />
 </Event>
</Events>
```

The parameters used in this example XML file are as follows:

- **eventType**
  Specifies the type of event. For this example, the value is AddCIEvent.
- **MdrElementID**
  Specifies a unique identifier for the CI. For this example, the value is MyMachine.
- **className**
  Specifies the USM class name. For this example, the value is ComputerSystem.
- **Label**
  Specifies a short description for identification of the CI. It is important that the value of the label uniquely distinguishes entities in the interface. For this example, the value is MyMachine.
- **Description**
  Specifies a description of the CI. For this example, the value is "Description of a Universal Connector Entity".
- **AdministrativeStatus**
  Specifies the high-level administrative status of the CI. For this example, the value is Managed.
- **Vendor**

Specifies the hardware vendor name. For this example, the value is Dell.

- **PrimaryDnsName**
  Specifies the fully qualified DNS name of the CI. For this example, the value is MyMachine.ca.com.
- **ComputerName**
  Specifies the name of the computer. For this example, the value is MyMachine.
- **MemoryInGB**
  Specifies the amount of system memory in GB. For this example, the value is 1024.

Add an Alert

To add alerts using an XML file, run the GCEventAddCmd batch program with the XML file that contains the <Event> tag and appropriate property tags with corresponding values. The template tag structure is as follows:

```
<Events>
 <Event>
   ...
   ...
 </Event>
</Events>
```

> **NOTE**
> For a complete list of USM property tags, see the USM schema to determine which USM properties are required for the specific class of CI.

The following example lets you add an alert of Minor severity by running the batch program as follows:

```
GCEventAddCmd -hSOIserver.ca.com:7090 -fAddAlert.xml
```

The example uses the AddAlert.xml file that includes the following information about the alert:

```
<Events>
 <Event>
   <property tag="eventType" value="StatusEvent" />
   <property tag="MdrElementID" value="A00001" />
   <property tag="className" value="Alert" />
   <property tag="AlertedMdrElementID" value="UCServer" />
   <property tag="AlertType" value="Risk" />
   <property tag="Severity" value="Minor" />
   <property tag="Summary" value="UC_Server has an infrastructure alarm.." />
   <property tag="Message" value="The Detailed message associated with this alert.." />
 </Event>
</Events>
```

The parameters used in this example XML file are as follows:

- **eventType**
  Specifies the type of event. In the case of alert, the value is StatusEvent.
- **MdrElementID**
  Specifies a unique alert ID. For this example, the value is A00001.
- **className**
  Specifies the USM class name. For this example, the value is Alert.
- **AlertedMdrElementID**
  Specifies the AlertedMdrElementID, which is the MdrElementID of the CI that the alert is associated with. For this example, the value is UCServer.
- **AlertType**

Specifies the type of alert: Risk, Quality, or Health. For this example, the value is Risk.

- **Severity**
  Specifies the alert severity. For this example, the value is Minor.
- **Summary**
  Specifies the appropriate summary text that you want to display. For this example, the value is "UC_Server has an infrastructure alarm..".
- **Message**
  Specifies the appropriate message text that you want to display. For this example, the value is "The Detailed message associated with this alert..".

Add a CI to a Service

To add a CI to a service, run the GCEventAddCmd batch program with an XML file containing the <Relationship> tag and appropriate required properties with corresponding values.

Note: For a complete list of USM property tags, see the USM schema to determine which USM properties are required for the specific class of CI.

The following example lets you add a CI to a service by running the batch program as follows:

```
GCEventAddCmd -hSOIserver.ca.com:7090 -fMyServiceRelation.xml
```

The MyServiceRelation.xml file includes the following information:

```
<Services>
 <Relationship>
   <property tag="eventType" value="AddRelationship" />
   <property tag="SourceMdrElementID" value="UCService" />
   <property tag="TargetMdrElementID" value="UCServer" />
   <property tag="ScopeMdrElementID" value="UCService" />
   <property tag="Semantic" value="HasMember" />
   <property tag="className" value="BinaryRelationship" />
 </Relationship>
</Services>
```

The parameters used in this example XML file are as follows:

- **eventType**
  Specifies the type of event. For this example, the value is AddRelationship.
- **SourceMdrElementID**
  Specifies the unique source identifier. For this example, the value is UCService.
- **TargetMdrElementID**
  Specifies the unique target identifier. For this example, the value is UCServer.
- **ScopeMdrElementID**
  Specifies the service in which the relationship exists. For this example, the value is UCService.
- **Semantic**
  Relates one USM instance to another. For this example, the value is HasMember.
- **className**
  Specifies the USM class name. For this example, the value is BinaryRelationship.

Update a CI

You can manage CIs in two different XML tag contexts<Events> tag and <Services> tag.

- For Events tag, the structure to update a CI is as follows:
  ```
  <Events>
  ```

```
  <Event>
    eventType = UpdateCIEvent
```

- For Services tag, the structure to update a CI is as follows:

```
  <Services>
   <Service>
    eventType = UpdateCI
```

> **NOTE**
> For a complete list of USM property tags, see the USM schema to determine which USM properties are required for the specific class of CI.

The following example lets you update a CI description by running the batch program as follows:

```
GCEventAddCmd -hSOIserver.ca.com:7090 -fUpdateCI.xml
```

The UpdateCI.xml file includes the following information:

```
<Events>
 <Event>
   <property tag="eventType" value="UpdateCIEvent" />
   <property tag="MdrElementID" value="MyMachine" />
   <property tag="className" value="ComputerSystem" />
   <property tag="Label" value="MyMachine" />
   <property tag="Description" value="Update to the Description" />
   <property tag="AdministrativeStatus" value="Managed" />
   <property tag="Vendor" value="Dell" />
   <property tag="PrimaryDnsName" value="MyMachine.ca.com" />
   <property tag="ComputerName" value="MyMachine" />
   <property tag="MemoryInGB" value="1024" />
 </Event>
</Events>
```

The parameters used in this example XML file are as follows:

- **eventType**
  Specifies the type of event. For this example, the value is UpdateCIEvent.
- **MdrElementID**
  Specifies a unique identifier for the CI. For this example, the value is MyMachine.
- **className**
  Specifies the USM class name. For this example, the value is ComputerSystem.
- **Label**
  Specifies a short description for identification of the CI. It is important that the value of the Label uniquely distinguishes entities in the interface. For this example, the value is MyMachine.
- **Description**
  Specifies a description of the CI. For this example, the value is "Update to the Description".
- **AdministrativeStatus**
  Specifies the high-level administrative status of the CI. For this example, the value is Managed.
- **Vendor**
  Specifies the hardware vendor name. For this example, the value is Dell.
- **PrimaryDnsName**
  Specifies the fully qualified DNS name of the entity. For this example, the value is MyMachine.ca.com.
- **ComputerName**
  Specifies the name of the computer. For this example, the value is MyMachine.
- **MemoryInGB**

Specifies the amount of system memory. For this example, the value is 1024.

Delete a CI

You can manage CIs in two different XML tag contexts<Events> tag and <Services> tag.

- For <Events> tag, the structure to delete a CI is as follows:

```
<Events>
 <Event>
   eventType = DeleteCIEvent
```

- For <Services> tag, the structure to delete a CI is as follows:

```
<Services>
 <Service>
   eventType = DeleteCI
```

> **NOTE**
> For a complete list of USM property tags, see the USM schema to determine which USM properties are required for the specific class of CI.

You can delete only those CIs (by using the Universal connector) that the Universal connector has created. You cannot delete a CI from a different connector.

The following example lets you delete a CI called MyMachine of USM type ComputerSystem by running the batch program as follows:

```
GCEventAddCmd -hSOIserver.ca.com:7090 -fDeleteCI.xml
```

The DeleteCI.xml file includes the following information:

```
<Events>
 <Event>
   <property tag="eventType" value="DeleteCIEvent" />
   <property tag="MdrElementID" value="MyMachine" />
   <property tag="className" value="ComputerSystem" />
   <property tag="Label" value="MyMachine" />
   <property tag="Description" value="Delete the ComputerSystem" />
   <property tag="AdministrativeStatus" value="Managed" />
   <property tag="Vendor" value="Dell" />
   <property tag="PrimaryDnsName" value="MyMachine.ca.com" />
   <property tag="ComputerName" value="MyMachine" />
   <property tag="MemoryInGB" value="1024" />
 </Event>
</Events>
```

The parameters used in this example XML file are as follows:

- **eventType**
  Specifies the type of event. For this example, the value is DeleteCIEvent.
- **MdrElementID**
  Specifies a unique identifier for the CI. For this example, the value is MyMachine.
- **className**
  Specifies the USM class name. For this example, the value is ComputerSystem.
- **Label**
  Specifies a short description for identification of the CI. It is important that the value of the Label uniquely distinguishes entities in the interface. For this example, the value is MyMachine.
- **Description**

Specifies a description of the CI. For this example, the value is "Delete the ComputerSystem".

- **AdministrativeStatus**
  Specifies the high-level administrative status of the CI. For this example, the value is Managed.
- **Vendor**
  Specifies the hardware vendor name. For this example, the value is Dell.
- **PrimaryDnsName**
  Specifies the fully qualified DNS name of the entity. For this example, the value is MyMachine.ca.com.
- **ComputerName**
  Specifies the name of the computer. For this example, the value is MyMachine.
- **MemoryInGB**
  Specifies the amount of system memory. For this example, the value is 1024.

Sample Universal Connector XML Files

Installing the Universal connector also installs sample XML files that help you understand the XML file structure and various property tags. These XML files provide examples of the format required for the type of items that you want to apply to CA SOI. The following are the sample XML files that you can find at the CA SOI root location:

- universalAdd_Alerts
- universalAdd_Application
- universalAdd_ApplicationServer
- universalAdd_ComputerSystem
- universalAdd_Database
- universalAdd_FileSystem
- universalAdd_Memory
- universalAdd_Processor
- universalAdd_Service
- universalAdd_ServiceModel
- universalDelete_ComputerSystem
- universalUpdate_ComputerSystem

universalAdd_Alerts

This sample XML file helps you add an alert. The structure of this sample file is as follows:

```
<Events>
 <Event>
   <property tag="eventType" value="StatusEvent" />
   <property tag="MdrElementID" value="A00001" />
   <property tag="className" value="Alert" />
   <property tag="AlertedMdrElementID" value="UCServer" />
   <property tag="AlertType" value="Risk" />
   <property tag="Severity" value="Minor" />
   <property tag="Summary" value="UC_Server has an infrastructure alarm.." />
   <property tag="Message" value="The Detailed message associated with this alert.." />
 </Event>
</Events>
```

universalAdd_Application

This sample XML file helps you add a new application CI. The structure of this sample file is as follows:

```
<Events>
 <Event>
   <property tag="eventType" value="AddCIEvent" />
   <property tag="MdrElementID" value="Application:Login" />
   <property tag="Label" value="Application:Login" />
   <property tag="className" value="Application" />
   <property tag="Description" value="Adding a new Application" />
   <property tag="Vendor" value="CA" />
   <property tag="ProductName" value="Login" />
   <property tag="DeviceSysName" value="CA:Login" />
 </Event>
</Events>
```

universalAdd_ApplicationServer

This sample XML file helps you add a new application server CI. The structure of this sample file is as follows:

```
<Events>
 <Event>
   <property tag="eventType" value="AddCIEvent" />
   <property tag="MdrElementID" value="ExchangeServer" />
   <property tag="className" value="ApplicationServer" />
   <property tag="Description" value="Adding a new Application Server" />
   <property tag="category" value="Exchange Services" />
   <property tag="Label" value="ExchangeServer" />
   <property tag="Vendor" value="CA" />
   <property tag="ProductName" value="Exchange" />
   <property tag="ProcessID" value="01" />
   <property tag="AccessedViaTcpPort" value="01" />
   <property tag="ProcessDistinguishingID" value="01" />
   <property tag="DeviceSysName" value="CA:ExchangeServer" />
 </Event>
</Events>
```

universalAdd_ComputerSystem

This sample XML file helps you add a computer system CI. The structure of this sample file is as follows:

```
<Events>
 <Event>
   <property tag="eventType" value="AddCIEvent" />
   <property tag="MdrElementID" value="MyMachine21" />
   <property tag="className" value="ComputerSystem" />
   <property tag="Label" value="MyMachine21" />
   <property tag="Description" value="Description of a Universal Connector Entity" />
   <property tag="AdministrativeStatus" value="Managed" />
   <property tag="Vendor" value="Dell" />
   <property tag="PrimaryDnsName" value="MyMachine21.ca.com" />
   <property tag="ComputerName" value="MyMachine21" />
   <property tag="MemoryInGB" value="1024" />
 </Event>
</Events>
```

universalAdd_Database

This sample XML file helps you add a database CI. The structure of this sample file is as follows:

```
<Events>
 <Event>
   <property tag="eventType" value="AddCIEvent" />
   <property tag="MdrElelentID" value="ReplicationDatabase01" />
   <property tag="className" value="Database" />
   <property tag="Description" value="Adding a new Database" />
   <property tag="category" value="Replication Services" />
   <property tag="Label" value="ReplicationDatabase01" />
   <property tag="DBInstanceName" value="ReplicationDatabase01" />
   <property tag="DeviceSysName" value="ReplicationDatabase01" />
   <property tag="DatabaseName" value="ReplicationDatabase01" />
 </Event>
</Events>
```

universalAdd_FileSystem

This sample XML file helps you add a new file system CI. The structure of this sample file is as follows:

```
<Events>
 <Event>
   <property tag="eventType" value="AddCIEvent" />
   <property tag="MdrElemmentID" value="FileSystem" />
   <property tag="className" value="File" />
   <property tag="Description" value="Adding a new FileSystem" />
   <property tag="category" value="Exchange Services" />
   <property tag="Label" value="FileSystem" />
   <property tag="StoreName" value="FileSystem" />
   <property tag="FilePathUrl" value="file://c" />
 </Event>
</Events>
```

universalAdd_Memory

This sample XML file helps you add a new memory CI. The structure of this sample file is as follows:

```
<Events>
 <Event>
   <property tag="eventType" value="AddCIEvent" />
   <property tag="MdrElementID" value="Memory" />
   <property tag="className" value="Memory" />
   <property tag="DEscription" value="Adding a new Memory" />
   <property tag="category" value="Exchange Services" />
   <property tag="Label" value="Memory" />
   <property tag="MemoryType" value="Memory" />
   <property tag="OSNumeric" value="01" />
   <property tag="ContaingIndex" value="01" />
   <property tag="DeviceSysName" value="Memory" />
 </Event>
</Events>
```

universalAdd_Processor

This sample XML file helps you add a new processor CI. The structure of this sample file is as follows:

```
<Events>
 <Event>
   <property tag="eventType" value="AddCIEvent" />
   <property tag="MdrElementID" value="CPU" />
   <property tag="className" value="Processor" />
   <property tag="Description" value="Adding a new Processor" />
   <property tag="category" value="Exchange Services" />
   <property tag="Label" value="CPU" />
   <property tag="ProcessorType" value="CPU" />
   <property tag="OSNumeric" value="01" />
   <property tag="ContainingIndex" value="01" />
   <property tag="DeviceSysName" value="CPU" />
 </Event>
</Events>
```

universalAdd_Service

This sample XML file helps you add a new service. The structure of this sample file is as follows:

```
<Events>
 <Event>
   <property tag="eventType" value="AddCIEvent" />
   <property tag="MdrElementID" value="MyService" />
   <property tag="className" value="Service" />
   <property tag="Label" value="MyService" />
   <property tag="Description" value="Description of a new Service" />
   <property tag="AdministrativeStatus" value="Managed" />
   <property tag="ServiceName" value="MyService" />
   <property tag="ServiceVersion" value="1" />
 </Event>
</Events>
```

universalAdd_ServiceModel

This sample XML file helps you add a service model. The structure of this sample file is as follows:

```
<Services>
 <Service>
   <property tag="eventType" value="AddService" />
   <property tag="MdrElementID" value="UCService" />
   <property tag="className" value="Service" />
   <property tag="ServiceName" value="UCService" />
   <property tag="ServiceVersion" value="0" />
   <property tag="Description" value="Adding a new Service" />
   <property tag="AdministrativeStatus" value="Managed" />
 </Service>
 <CI>
   <property tag="eventType" value="AddCI" />
   <property tag="MdrElementID" value="UCServer" />
   <property tag="className" value="ComputerSystem" />
   <property tag="Label" value="UCServer" />
```

```
    <property tag="Description" value="Description of a new ComputerSystem" />
    <property tag="AdministrativeStatus" value="Managed" />
    <property tag="Vendor" value="Dell" />
    <property tag="PrimaryDnsName" value="UCServer.ca.com" />
    <property tag="ComputerName" value="UCServer" />
    <property tag="MemoryInGB" value="1024" />
    <property tag="CIuserAttribute1" value="Attr1" />
      <property tag="CIuserAttribute2" value="Attr2" />
      <property tag="CIuserAttribute3" value="Attr3" />
      <property tag="CIuserAttribute4" value="Attr4" />
      <property tag="CIuserAttribute5" value="Attr5" />
  </CI>
  <Relationship>
    <property tag="eventType" value="AddRelationship" />
    <property tag="SourceMdrElementID" value="UCService" />
    <property tag="TargetMdrElementID" value="UCServer" />
    <property tag="ScopeMdrElementID" value="UCService" />
    <property tag="Semantic" value="HasMember" />
    <property tag="className" value="BinaryRelationship" />
  </Relationship>
 </Services>
```

universalDelete_ComputerSystem

This sample XML file helps you delete a computer system CI. The structure of this sample file is as follows:

> **TIP**
>
> Although this example shows more attributes, only the MDR Element ID and Class Name attributes are required for CI deletes.

```
 <Events>
  <Event>
    <property tag="eventType" value="DeleteCIEvent" />
    <property tag="MdrElementID" value="MyMachine21" />
    <property tag="className" value="ComputerSystem" />
    <property tag="Label" value="MyMachine21" />
    <property tag="Description" value="Delete the ComputerSystem" />
    <property tag="AdministrativeStatus" value="Managed" />
    <property tag="Vendor" value="Dell" />
    <property tag="PrimaryDnsName" value="MyMachine21.ca.com" />
    <property tag="ComputerName" value="MyMachine21" />
    <property tag="MemoryInGB" value="1024" />
  </Event>
 </Events>
```

universalUpdate_ComputerSystem

This sample XML file helps you update a computer system CI. The structure of this sample file is as follows:

```
 <Events>
  <Event>
    <property tag="eventType" value="UpdateCIEvent" />
    <property tag="MdrElementID" value="MyMachine21" />
    <property tag="className" value="ComputerSystem" />
```

```
    <property tag="Label" value="MyMachine21" />
    <property tag="Description" value="Update to the Description" />
    <property tag="AdministrativeStatus" value="Managed" />
    <property tag="Vendor" value="Dell" />
    <property tag="PrimaryDnsName" value="MyMachine21.ca.com" />
    <property tag="ComputerName" value="MyMachine21" />
    <property tag="MemoryInGB" value="1024" />
  </Event>
 </Events>
```

How to Map Old Schema Properties to USM Properties with Universal Connector

As an integration developer, you can map the old schema properties to USM properties to use the old XML files with the Universal connector. The Universal connector now supports only USM-based properties. XML files using the old schema properties, therefore, do not work with the Universal connector. If you want to use the old XML files, map the old schema properties to USM properties based on the eventType value.

You can do this mapping based on your specific requirements as follows:

- Map Properties for AddCIEvent, AddService, or AddCI.
- Map Properties for StatusEvent.
- Map Properties for AddRelationship.

Map Properties for AddCIEvent, AddCI, or AddService

When you use AddCIEvent, AddCI, or AddService as the value for the eventType tag, you map the mandatory old properties to the corresponding USM properties as follows:

Old Properties	USM Properties
instanceID	MdrElementID
className	className (must represent a valid USM class)
situationMessage	Description
deviceID	Not Applicable
severity	Not Applicable

Map Properties for StatusEvent

When you use StatusEvent as the value for the eventType tag, you map the mandatory old properties to the corresponding USM properties as follows:

Old Properties	USM Properties
siloAlarmID	MdrElementID
instanceID	AlertedMdrElementID
className	className (must be Alert in this case)
situationMessage	Summary
severity	Severity (value represents Normal, Minor, Major, Critical, or Down)
situationType	AlertType
deviceID	Not Applicable

Map Properties for AddRelationship

When you use AddRelationship as the value for the eventType tag, you map the mandatory old properties to the corresponding USM properties as follows:

Old Properties	USM Properties
aNodeCI	SourceMdrElementID
bNodeCI	TargetMdrElementID
serviceCI	ScopeMdrElementID
associationType	Semantic
className	className (must be BinaryRelationship)

Product Connector Documentation

This section provides the documentation for all product connectors.

To access current product connector documentation, click here.

Connector Development

This section describes how an integration developer can develop a custom connector.

Sample Connector

Contents

The Sample connector uses test data and a mock domain manager to simulate data retrieval. The connector interacts with the IFW and the CA Catalyst Synchronizer components. You can install the connector to integrate sample data into CA SOI, test CA SOI functionality, and illustrate how a connector operates.

What is Sample Connector

The Sample connector is a fully functional connector that creates sample CIs, alerts, and relationships in CA SOI to demonstrate and test connector functionality. The Sample connector uses a mock domain manager based on sample data files, which you can easily manipulate to change the connector's behavior. It also provides the materials to build a custom Level 4, 5, or 6 connector, including Java code, connector policy, configuration, installer kit, and test data.

Running the Sample connector or using the Sample connector to create a connector helps you understand how a custom connector works and how the methods and policy are defined.

How the Sample Connector Works

The Sample connector uses a mock domain manager interface to retrieve sample data and demonstrate connector functionality. The connector gets the data to create CIs, relationships, and alerts based on XML files in the <SOI_HOME>\resources\SampleConnector\data directory. The data in the files is represented in both the USM format and a Sample connector format that simulates a domain manager format.

The following files supply data to the Sample connector through the mock domain manager:

- **sample-cis.xml**
  Contains sample CIs and relationships that are imported when the connector starts. The sample data demonstrates the following CI types:

  - ComputerSystem
  - Router
  - InterfaceCard
  - Processor
  - File
  - BackgroundProcess
  - DatabaseInstance
  - BinaryRelationship
  - Service
- **sample-alerts.xml**
  Contains sample open alerts that are loaded at startup.
- **sample-ci-changes.xml**
  Contains sample data to simulate changes in the Sample connector domain manager such as create, update, and delete operations. Any changes recorded in this file apply at the connector startup and occur while the connector is running.
- **sample-metrics.xml**
  Contains sample metric data.

The Sample connector contains policy files that transform Sample connector data to the USM format for transmittal to CA SOI and transform USM data to the Sample connector format for invoking inbound operations on the mock domain manager data.

This provided Sample connector functionality lets you view sample data in CA SOI, simulate CI changes, and test inbound operations on other connectors.

Use Cases

You can use the Sample connector in *any* of the following ways:

- **To verify a CA SOI installation**
  The Sample connector can help you verify the correct operation of the CA SOI IFW to ensure that the product is running connectors and displaying connector data.
  For more information, see Running the Sample Connector.
- **As a base for creating a custom connector**
  The Sample connector contains the materials necessary to build a custom Level 4, 5, or 6 connector. It demonstrates the implementation of all required Java interfaces and the connector policy. It also demonstrates how to use the test harness provided with the system so that you can verify the connector's functionality before deploying the connector to CA SOI. The Sample connector implementation works with mock data that simulates a domain manager. You can use this interface for testing or when the domain manager data is not easily available.
  For more information, see Building a Custom Connector.
- **To test advanced custom connector functionality**
  The Sample connector can serve as a trigger to test the inbound operations that enable a custom connector to create, update, and delete data in its domain manager based on CI changes in CA SOI. This testing technique forces a change to an instance of a USM type managed by the Sample connector. The CA Catalyst Synchronizer then sends that change to all connectors enabled for the USM type. If a custom connector supports inbound operations for the USM type, the connector's create, update, or delete method is called and tested.
  For more information, see Update Sample Connector Data.

Install the Sample Connector

Install the Sample connector on any supported operating system using the installer provided with CA SOI. The connector must be able to connect to an SA Manager system, but does not have to be on the same system as the SA Manager.

NOTE
If you want to build a custom connector using the Sample connector framework, install the Sample connector on the SA Manager.

Follow these steps:

1. Run the Connector_Sample.exe file at the root of the CA SOI installation image and click Next.
2. Accept the license agreement, then accept the third-party license agreements, and click Next.
   The Choose Install Folder screen opens.

   NOTE
   This page does not open if CA SOI components already reside on the system.
3. Accept, enter, or choose the installation folder.

   NOTE
   The maximum installation path length is 150 characters. The installer blocks paths with more than 150 characters.

   If you are not installing on the SA Manager server, the Integration Services Configuration page opens; otherwise, skip to Step 5.
4. Specify the required information to connect to the appropriate SA Manager and configure connector preferences.

   NOTE
   Refer to your Installation Worksheet in the SA Manager section for these values.
5. Select whether to start the CA SOI services after installation and click Next.
6. Click Install.
   The connector installs on the system and integrates with the appropriate CA SOI instance. An installation summary page opens when the installation finishes.
7. (Optional) Review the installation log file (Sample_Install_*releasenumber*) that is in the <SOI_HOME>\log folder to check for installation errors.

Installed Components

The Sample connector installs the following components:

NOTE
The string <SOI_HOME> refers to the CA SOI installation directory. The default is C:\Program Files (x86)\CA \SOI. If you install the Sample connector on a machine that already contains CA SOI components, the connector is installed into the existing directory.

- **<SOI_HOME>\lib\sample.catalyst.connector.jar**
  Contains the code that runs the Sample connector in CA SOI.
- **<SOI_HOME>\resources\Configurations**
- sampleConnector_*connectorserver*.xml
  Contains the following Sample connector configuration information for use by the SA Manager:
  – Java class that implements the connector
  – Connector policy files
  – Indication if the connector is enabled

  The file also contains configuration information used by the Sample connector, such as the names of the provided sample data files and connection information for the mock domain manager.
- **<SOI_HOME>\resources\Core\Catalogpolicy\sampleconnector_policy.xml and sampleconnector_policySB.xml**
  Contains the policy for transforming Sample connector data to and from the USM format. The sampleconnector_policy.xml file transforms outbound data from the format produced by the Java code to the USM

format. The sampleconnector_policySB.xml file transforms inbound data from the USM format to the format consumed by the connector Java code.

- **<SOI_HOME>\resources\SampleConnector\data**
  Contains sample data files used by the Sample connector.
- **<SOI_HOME>\SampleConnector**
  Contains the files that implement the Sample connector. The SampleConnector folder contains the Eclipse project files (.project and .classpath) and the following subfolders:
  - **src\java**
    Contains the Java source files.
  - **src\resources**
    Contains same sample data and policy files included in the <SOI_HOME>\resources directory.
  - **test\java**
    Contains SampleConnectorUTest, the main class for testing the connector using the test harness.
  - **test\resources**
    Contains data files used by the Sample connector when running with the test harness.
  - **lib**
    Contains libraries required to run connectors with the test harness that are not required to run connectors with CA SOI.
  - **docs**
    Contains the Javadoc generated from the Sample connector Java source.
  - **InstallKit**
    Contains template files for creating a connector configuration file, connector policy files, and a connector installer.

    WARNING
    If you make changes to the sample data or policy files and want to view the effect on a Sample connector running with CA SOI, you must make the changes in the <SOI_HOME>\resources\SampleConnector\data directory and restart the CA SAM Integration Services service (you do not have to restart the service if changing the sample-ci-changes.xml file). Changing the files in <SOI_HOME>\SampleConnector has no effect on a Sample connector running with CA SOI. However, when running the Sample connector with the test harness, the files in <SOI_HOME>\SampleConnector are used.

Enable the Sample Connector

When you install the Sample connector, it is disabled by default. Enable the connector in its connector configuration file to run the connector in CA SOI.

Follow these steps:

1. Open the sampleConnector_*connectorserver*.xml file located at <SOI_HOME>\resources\Configurations on the connector system.
2. Set the State attribute in the element '<Silo>' to "Enabled".
3. Save and close the file.
4. Restart the CA SAM Integration Services service.

Verify that the Sample Connector is Running

You verify that the Sample connector is running and test its basic functionality to ensure that it is working correctly.

Follow these steps:

1. Open the Dashboard.
2. Select the Administration tab.
3. Expand Connector Configuration and the server where the Sample connector resides.
4. Select the CA:09998_*domainserver@connectorserver* entry.

NOTE
Because the Sample connector works with an embedded mock domain manager, the domain manager server is always the same as the connector server.

Information about the Sample Connector displays in the right pane. The connector displays as Online after it finishes initializing.

If the Sample connector entry does not appear, ensure that you have enabled the connector and restarted the CA SAM Integration Services service.

Verify Service, CI, and Alert Creation

You verify that the Sample connector can create services, CIs, and alerts by importing a provided sample service into the Operations Console.

Follow these steps:

1. Open the Operations Console.
2. Select Tools, Import Services.
   The Configure Data Sources dialog opens.
3. Select the CA:09998_*domainserverr@connectorserver* entry and click Import.
   The Import Services dialog opens.
4. Select SampleService in the left pane, click the right arrow to move SampleService to the right pane, and click OK.
   The service imports. The service displays as SampleService in the Operations Console and should contain CIs and alerts.
   The service, CIs, and alerts are verified.

Update Sample Connector Data

You can update or delete CIs, alerts, and services created by the Sample connector or create new ones by modifying the sample-ci-changes.xml file. You can view the changes to these objects in CA SOI. In a custom connector, you perform this operation by creating, updating, or deleting data in the source domain manager.

You use this functionality to test inbound create, update, and delete operations in other connectors including a custom connector. For example, when you create a CI in the Sample connector, the CA Catalyst Synchronizer sends that creation to all connectors enabled for the USM type that was created.

Follow these steps:

1. Save a backup copy of the <SOI_HOME>\resources\SampleConnector\data\sample-ci-changes.xml file and open the file.

   NOTE
   Make sure that you are accessing this file from the correct location. Changing the files in the <SOI_HOME>\SampleConnector directory has no effect on a running Sample connector.

   The file lists several predefined change operations.
2. Modify the first <event> element in the file as follows:
   – Modify the action and entitytype attributes as follows to add a new operation that adheres to the established schema in the file:

     ```
     <event xsi:type="usm:SiloDataChange" action="" entityType="">
     ```

     - **action**
       Defines the type of action to perform. Valid values are create, update, and delete.
     - **entitytype**

Defines the type of entity to create, update, or delete. Valid values are Relationship, Alert, and Item. Use the same value that you used in this attribute in the nested element <property_name="entitytype" />.

– Modify the <property> elements nested in the <silodata> element. Each <property> element defines one property name and value. The properties can be USM properties or a more compact set of Sample connector properties. Define the USM format by setting the value of the eventtype property to USM-Entity as follows:

```
<property name="eventtype" value="USM-Entity" />
```

Omit this property to use the Sample connector properties.

– Specify the type of entity in the entitytype property. Valid values are Item, Alert, and Relationship. This value should match the value of the entitytype attribute in the <event> element.
Example: <property name="entitytype" value="Item" />

– Define the object type in the class property for the Sample connector format or the ClassName property for the USM format as follows:
Example: <property name="class" value="Router" />
Example: <property name="ClassName" value="Service" />
Examples of USM and Sample connector formats are included in the file.

> **NOTE**
> The Sample connector supports only a subset of the USM object types.

– When the action is update or delete, you must specify the correct value for id (Sample connector format) or MdrElementID (USM format). If you are referring to a CI created in a prior update to sample-ci-changes.xml, use the value of the generated ID. Find the value of MdrElementID for existing objects by examining the USM properties for the CI in the USM Web View.

> **NOTE**
> For more information about the USM Web View, see USM Web View for PC and USM Web View for Mobile Devices.

When the action is create, the value of id or MdrElementID is ignored, because the item does not yet exist in CA SOI. The Sample connector assigns a value automatically.

3. Save and close the file.
The connector detects that the file has changed and sends change events to CA SOI. Verify updates by checking the CIs in the Operations Console.
The CA Catalyst Synchronizer (if enabled) sends change events back to the configured connectors that have inbound operations enabled. Sample connector data is updated.

> **NOTE**
> The CA Catalyst Synchronizer is disabled by default. Synchronization is only supported for specific use cases. For more information about enabling the Synchronizer and supported use cases, see Synchronization.

Example: Create a new Router CI in Sample connector format

The following example creates a new Router class CI in Sample connector format.

```
<event xsi:type="usm:SiloDataChange" action="create" entitytype="Item">
    <silodata xsi:type="usm:SiloData" entitytype="Item">
        <properties>
            <property name="entitytype" value="Item" />
            <property name="id" value="1" />
            <property name="name" value="testrouter" />
            <property name="ip_address" value="13.1.1.1" />
            <property name="class" value="Router" />
            <property name="description" value="Cisco Router" />
            <property name="sysname" value="testrouter" />
            <property name="dnsname" value="testrouter.ca.com" />
```

```
        </properties>
    </silodata>
</event>
```

The create action specifies to create a new entity. The absence of the property eventtype=USMEntity indicates that the properties are in the Sample connector format. The class of Router indicates the entity type. Although an id value is provided, this value is ignored, and the Sample connector automatically assigns a unique value.

Example: Update an existing Alert in USM format

The following example updates an alert that already exists in the Operations Console in USM format:

```
<event xsi:type="usm:SiloDataChange" action="update">
    <silodata entitytype="Alert">
        <properties>
            <property name="eventtype" value="USM-Entity" />
            <property name="entitytype" value="Alert" />
            <property name="ClassName" value="Alert" />
            <property name="MdrElementID" value="1008" />
         <property name="AlertedMdrElementID" value="100" />
            <property name="AlertedMdrProduct" value="CA:09998" />
            <property name="UrlParams" value="http://localhost?id=8" />
            <property name="Message" value="Service is stopped - Again" />
            <property name="Summary" value="Service is stopped - Again" />
            <property name="Severity" value="Critical" />
            <property name="AlertType" value="Quality" />
            <property name="OccurrenceTimestamp"
            value="0001-05-01T00:00:00-00:00" />
            <property name="ReportTimestamp"
            value="0001-07-01T00:00:00-00:00" />
        </properties>
    </silodata>
</event>
```

The action of update specifies to update an existing object, the eventtype of USM-Entity indicates the USM format, and the entitytype of Alert indicates that the object is an existing alert. The connector uses the value of the AlertedMdrElementID property to match the information with an existing object in CA SOI.

How to Build a Custom Connector

Contents

As an integration developer, you can build a custom connector using the Sample connector as a template. Using the Sample connector as a template is the recommended method for developing a custom connector.

Complete the following process to build a custom connector based on the Sample connector:

1. Initialize an Eclipse project for the connector.
2. Implement the required interfaces and methods.
3. Create the connector configuration file.
4. Create a connector policy.
5. Test the connector.
6. Deploy the connector.

Connector Writing Basics

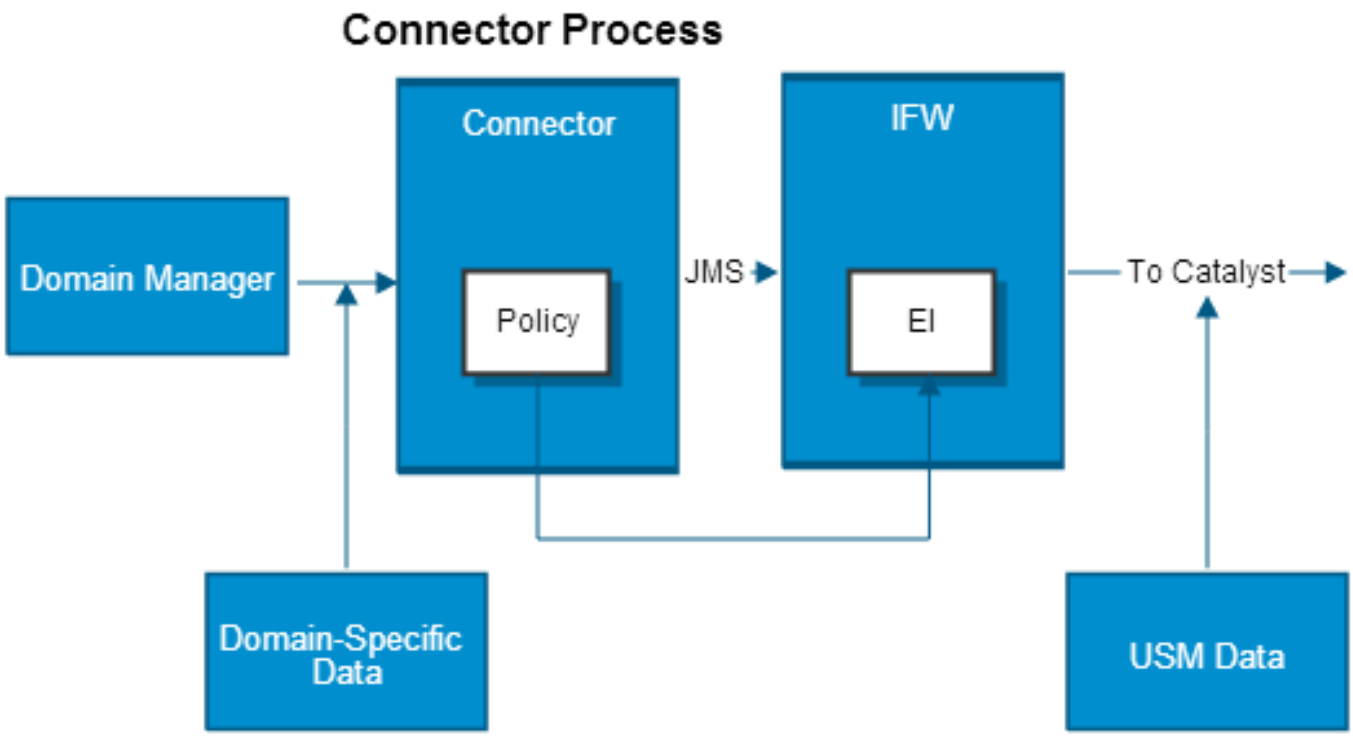
You can write custom connectors to create Level 4, 5, or 6 connector integrations for any domain manager product. Connectors require the following main components:

- Java code to implement the required interfaces for the connector
- Methods to retrieve raw data from a domain manager
- Connector policy that specifies how to transform the outbound raw domain manager data into a standard USM format
- Optional methods and policy that allow for inbound create, update, and delete operations in the domain manager

Connectors retrieve raw domain manager data such as CIs, alerts, and relationships as property-value pairs and normalize properties and values so that they adhere to the USM schema, which is the internal schema of the CA Catalyst framework and SA Manager.

The following illustration shows the process:

Figure 61: Connector Process



1. The connector methods interface with the domain manager and retrieve entities in the format of the domain manager.
2. The connector sends domain-specific data to the IFW through JMS.
3. The IFW processing module (pictured above as EI) uses connector policy to transform the entities to the USM format if necessary.
4. The IFW transmits the USM entities through the ActiveMQ server to CA Catalyst and the SA Manager for display on the CA SOI interfaces as CIs, alerts, or relationships.

Connector Considerations

There are some basic questions to consider before developing a connector. Depending on the domain manager and the complexity of the data, you can take different approaches, such as the following:

- Get every property for every entity and then determine how to normalize them
- Identify the data needed to fill all the USM property values and determine what domain manager data you need for those properties

Usually, the process falls somewhere in between and becomes an iterative approach as you start seeing what the domain manager data looks like and try to determine how to map it to USM types.

Some common questions to address when planning a custom connector are as follows:

- What entities and associated properties can the domain manager provide and how do I get them?
- Which of these entities can be represented in USM?
- Does the domain manager provide a group or service concept?
- How is an alert represented?
- How can you access the domain manager CI and alert data?
- Does the domain manager provide launch in context capabilities?
- Will CIs be updated or deleted?
- What USM types best describe the types of resources these entities represent?
- What USM properties require data and what is their format?
- What properties of the domain manager entities can fill the required properties of USM?
- What normalization is necessary to convert the domain manager property values to the USM format?

For more information about the USM types to which you can normalize all domain manager resources, see the USM schema documentation. For information about how to access the USM documentation,

System Prerequisites

The system that you use to build a connector using the Sample connector requires the following:

- SA Manager installed
- Sample connector installed (to gain access to the project files)
- AdoptOpen JRE 1.8.0.212
- Eclipse 3.5
- TestNG plug-in for Eclipse

How to Set Up the Sample Connector in Eclipse IDE

Contents

Load the Sample connector project into Eclipse to build a connector and run the test harness. The Sample connector project in Eclipse provides a framework for building and testing a connector, including the required interfaces and optional test harness and mock data interfaces.

You set up a Sample connector project in Eclipse as follows:

1. Create a workspace directory in Eclipse and specify the workspace directory when you start the application for the first time.
2. Configure Eclipse to use the JRE used by CA SOI.
3. Import the Sample connector project into Eclipse.
4. Install the TestNG plug-in.
5. Configure the project launch settings.
6. Start Eclipse with a new workspace.

Configure Eclipse to Use Correct JRE

Configure Eclipse to use the JRE that CA SOI uses before you import the Sample connector.

Follow these steps:

1. Open Eclipse and select Window, Preferences.
   The Preferences dialog opens.
2. Expand Java and select Installed JREs.
   The Installed JREs page opens.
3. Click Add.
   The Add JRE dialog opens.
4. Select Standard VM and click Next.
   The JRE Definition page opens.
5. Select Directory, select the SOI_HOME\jre folder, and click OK.
   The JRE information appears in the JRE Definition page.
6. Click Finish.
   The JRE appears in the Installed JREs page.
7. Select the check box next to the CA SOI JRE and click OK.
   The JRE preference is saved and Eclipse is configured.

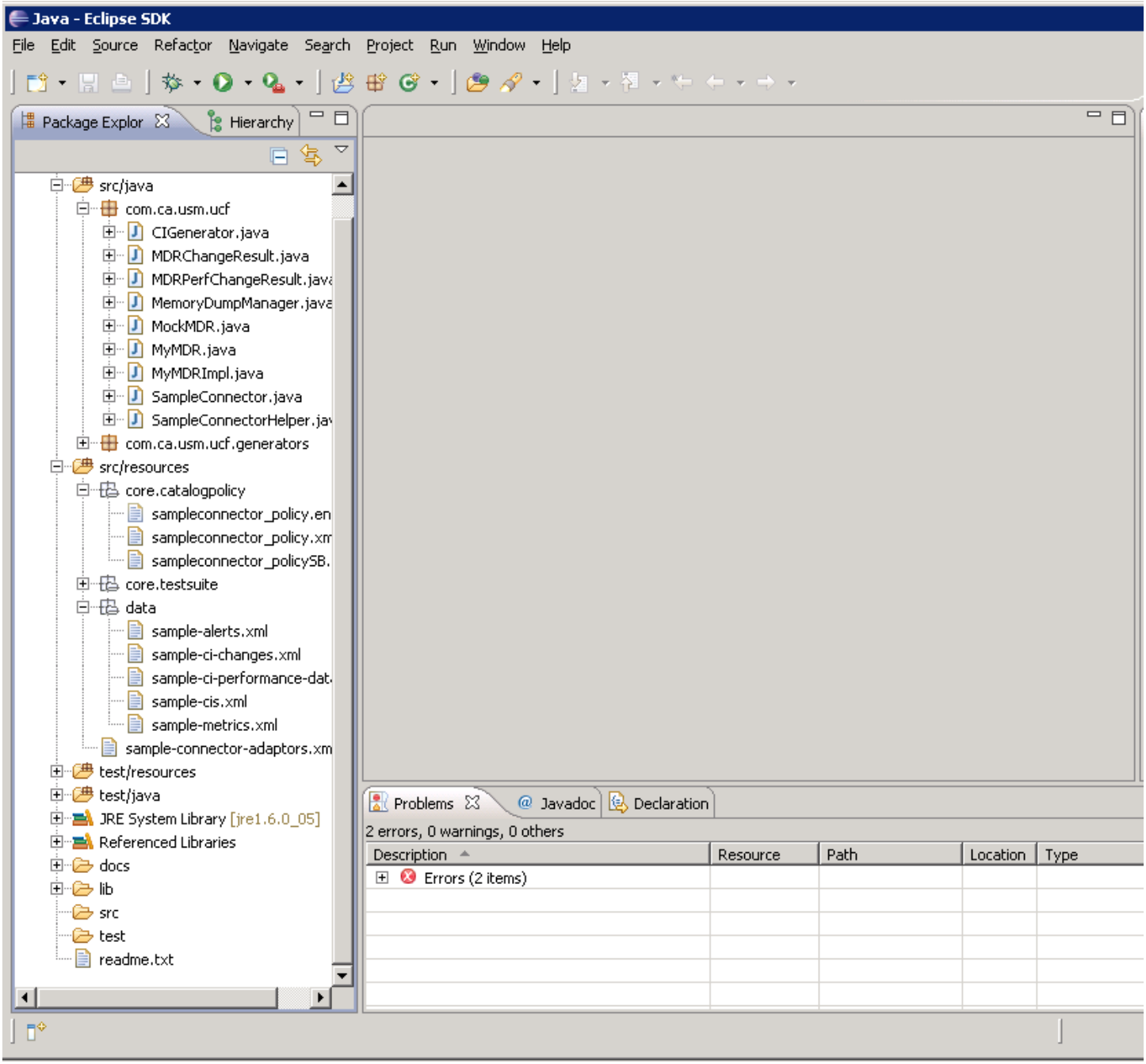
Import Sample Connector Project

You import the Sample connector project into Eclipse in a new workspace.

Follow these steps:

1. Start Eclipse with a new workspace and Select File, Import.
   The Import page opens.
2. Expand General, select Existing Projects into Workspace, and click Next.
   The Import Projects page opens.
3. Choose Select root directory, click Browse, navigate to the SOI_HOME\SampleConnector directory, and click OK.
   The SampleConnector project displays as selected in the Projects pane.
4. Click Finish.
   The Sample connector project is imported.

The following screen shows the imported SampleConnector project expanded to display the provided interfaces, policy, and sample data:



Install TestNG Eclipse Plug-in

To run connector unit tests, install the TestNG plug-in in your Eclipse environment.

Follow these steps:

1. Uninstall any existing previous version of TestNG.
2. Start Eclipse and select Help, Install New Software.
   The Install page opens.
3. Click Add.
   The Add Site page opens.

4. Enter TestNG in the Name field, enter http://beust.com/eclipse in the Location field, and click OK.
   Information for the plug-in displays in the table on the Install dialog.
5. Expand TestNG, select org.testng.eclipse, and click Next twice.
   The Review Licenses page opens.
6. Accept the License Agreement and click Finish.
   TestNG installs, and Eclipse asks for the permission to restart.
7. Click Yes.
   Eclipse restarts, and the TestNG plug-in is now available for use.

Configure Project Launch Settings

You configure the project launch settings to specify where the Sample connector should start.

Follow these steps:

1. Open Eclipse.
2. Select Run, Debug Configurations.
   The Debug Configurations dialog opens.
3. Right-click TestNG and select New.
   The right pane is populated with the TestNG configuration information.
4. Enter a meaningful name in the Name field and click Browse in the Project field.
   The Project Selection dialog opens.
5. Select SampleConnector and click OK.
   SampleConnector displays in the Project field.
6. Select Class in the Run pane and click Browse next to the field.
   The TestNG dialog opens.
7. Select SampleConnectorUTest and click OK.
   The class displays in the Class field.
8. Click Apply.
   The project launch settings are configured. Complete Steps 9-11 to run a unit test.
9. Click Debug.
   The unit test runs.
10. Select the TestNG tab.
    The unit test results display.
11. Click the icon labeled Open TestNG Report.
    The execution results report displays in HTML format. All tests pass if the environment is configured correctly.

How You Implement a Custom Connector

Contents

After you have loaded the Sample connector project into Eclipse, you can begin implementing the domain manager integration for a custom connector.

> **NOTE**
> You must install the Sample connector on the SA Manager to implement the connector interfaces and methods.

Connector Interfaces and Methods

The Sample connector extends the USMBaseConnector class, which implements some of the common connector framework code. The Sample connector also implements the following interfaces that define connector functionality. When writing a custom connector, you can code the necessary interfaces and methods using the Sample connector framework to properly integrate with a domain manager:

NOTE
View these interfaces in the Eclipse project in the SampleConnector.java class located in the src/java folder.

- **com.ca.connector.runtime.Connector**
  Contains method declarations to support the connector lifecycle.
- **com.ca.connector.runtime.EntityManager**
  Contains method declarations for retrieving data from the domain manager and creating, update, and deleting data in the domain manager.
- **com.ca.connector.runtime.EntityChangePublisher**
  Contains method declarations for events related to changes in CIs.
- **com.ca.connector.runtime.ConnectorEventPublisher**
  (Not supported) Contains method declarations for non-CI related events.
- **com.ca.connector.runtime.EntityOperationRunner**
  (Not supported) Contains method declarations to support custom operations related to specific entities.
- **com.ca.connector.runtime.ConnectorOperationRunner**
  (Not supported) Contains method declarations to support connector custom operations.
- **com.ca.connector.runtime.OperationRunner**
  (Not supported) Contains method declarations required to run and monitor the progress of synchronous and asynchronous operations.

  WARNING
  The ConnectorEventPublisher, ConnectorOperationRunner, EntityOperationRunner, and OperationRunner interfaces in the connector code are placeholders with no current implementation. The operations provided by these interfaces are not supported in this release of CA SOI.

Connector Interface

The com.ca.connector.runtime.Connector interface contains the methods that control the connector lifecycle with the domain manager. This interface includes the following methods:

- public void initialize(UUID *uuid*, DataObject *config*, Properties *properties*) throws UCFException;
  Initializes the connector, based on a configuration file that you define in the SOI_HOME\resources\Configurations directory. The integration framework reads any configuration parameters that are listed in the connector configuration file (*ConnectionInfo* tag), and pass them as input parameters to the initialize() method. As the name indicates, this method completes any tasks required for the connector to become ready for fulfilling requests for data.
  - **uuid**
    Uniquely identifies this instance of the connector.
  - **config**
    Created from an instance of the ConnectorConfigDesc associated with the connector. The config object is edited by installers, the administration user interface, and other means.
  - **properties**
    Defines properties that are provided to the instance and are typically used by proxy client implementations to connect to the domain manager.
- **public boolean isSystemUp();**
  Determines whether the domain manager is available. This method checks whether the system is running and returns a flag indicating whether the underlying system is up or down. The IFW uses this method to help detect error states. If this method does not return *true*, the connector is not able to come online.
- **public void restart();**
  Restarts the connector. This is an equivalent to a shutdown and reinitialize.
- **public void shutdown();**
  Shuts down the connector and cleans up the connection with the underlying system.

EntityManager Interface

The com.ca.connector.runtime.EntityManager interface contains the methods that support the data retrieval from the domain manager and inbound operations in the domain manager. Only the get() method is required if you do not want to enable inbound operations in the domain manager. This interface includes the following methods:

- public Collection<DataObject> get(DataObject *selector*) throws UCFException;
  Retrieves the entities that match the provided selector. The method reads data from the domain manager in response to a request from the IFW. The get() method accepts a selector of type DataObject that instructs the connector about the type of data that is being requested. In CA SOI, the underlying type for this selector DataObject is SiloDataFilter . The return type for the get() method is a Collection of DataObjects. Each DataObject in the return collection is a single entity from the domain manager, having the CA SOI underlying type of SiloData.
  - **selector**

    Defines an instance of the selector type associated with one of the entity types managed by the connector.
- public DataObject create(DataObject *newValue*) throws UCFException;
  (Optional) Creates the specified entity in the domain manager and returns the created entity.
  - **newValue**
    Defines the new value for the managed entity to be created in the domain manager.
- public DataObject update(DataObject *newValue*) throws UCFException;
  (Optional) Updates the specified entity in the domain manager and returns the updated entity.
  - **newValue**
    Defines the new value for the managed entity to be updated in the domain manager.
- public void delete(DataObject *selector*) throws UCFException;
  (Optional) Deletes the entity in the domain manager that matches the provided selector.
  - **selector**

    Defines an instance of the selector type associated with one of the entity types managed by the connector.
- public void discover(DataObject *selector*) throws UCFException;
  (Not supported) Discovers the entities that match the provided selector.
  - **selector**

    Defines an instance of the selector type associated with one of the entity types managed by the connector.

    If the selector is found, it is reported as a normal onCreate event to CI change subscribers.
- public Enumeration<DataObject> enumerate(DataObject *selector*, UUID *enumID*) throws UCFException;
  (Not supported) Enumerates the entities that match the provided selector and returns an enumeration of the entity type that matches the selector. While this method is semantically equivalent to a (Java) enumeration over the results of the get(EntitySelector sel) method, this call provides the opportunity to iteratively build/transfer the results. For example, an implementation may compute/retrieve matching entities from the managed system in batches.
  - **selector**

    Defines an instance of the selector type associated with one of the entity types managed by the connector.
  - **enumID**

    Defines a unique ID that is subsequently used by the caller in a call to closeEnumeration() to indicate that it is no longer interested in the enumeration.
- public void closeEnumeration(UUID *enumID*) throws UCFException;
  (Not supported) Indicates that callers are no longer interested in the remaining values of an enumeration started earlier with the same ID.
  - **enumID**

    Defines a unique ID that is subsequently used by the caller in a call to closeEnumeration() to indicate that it is no longer interested in the enumeration.

WARNING
While the connector framework supports the discover(), enumerate(), and closeEnumeration() methods, CA SOI does not currently use the functionality enabled by these methods. While these methods are available for implementation, the Sample connector does not contain sample code or implement any of these methods by default.

SiloDataFilter

SiloDataFilter is the primary filter that CA SOI uses when calling interfaces that the connector implements. SiloDataFilter supports the following filter elements:

- **entitytype**
  Represents the category of the USM data types in which a user is interested. The data type of the entitytype filter element is String. The valid values for entitytype are as follows:
  - **Item**
    Represents all entities that a domain manager manages, such as routers, applications, services, and so on. It does not include alerts and relationships.
  - **Relationship**
    Represents all USM-supported BinaryRelationships.
  - **Alert**
    Represents all USM alerts that a domain manager generates.
- **itemtype**

  NOTE
  The itemtype filter element is valid only when the entitype is set to Item. You cannot use itemtype if entitytype is set to Relationship or Alert.

  Represents the specific item type that is requested. The specific USM type names define the valid itemtype values, such as ComputerSystem, Service, Router, and so on. You can use NULL if you do not want to set any refinement condition for itemtype. The data type of the itemtype filter element is String.
- **recursive**
  Specifies whether the connector recursively collects the item and its constituent children and relationships. The data type of the recursive filter element is Boolean. The valid values are true and false. True retrieves complete definition of an item, including its subitems, constituent items and the relationship between these items. False retrieves only the top-level items.
  For example, if entitytype is set to Item, itemtype to Service, and recursive to true, all relationships and referenced items for a service are retrieved. If recursive is set to false, only the top-level services are retrieved.
- **Id**
  Specifies the entity instance ID as represented within the domain manager. This ID is the same as MdrElementID. When set to NULL, all entities of type set in the entitytype and itemtype filter elements are returned. The data type of the Id filter element is String.
- **updatedAfter**
  Specifies all entities of the type set in entitytype and itemtype that were updated after the specified date. The valid format is YYYY-MM-DDThh: mm: ss.sTZD. The data type of the updatedAfter filter element is DateTime.

  NOTE
  When filter is NULL; that is, no filter element is present, all entities are returned. Additionally, when multiple filter elements are present, they are combined with an AND operator.

SiloData

SiloData is the DataObject type that a connector uses to report all data to CA SOI. SiloData supports the following data elements:

- **entitytype**

Represents the category of the USM data types that the connector returns. The valid entitytype values are as follows:
  - **Item**
    Represents all entities that a domain manager manages, such as routers, applications, services, and so on. It does not include alerts and relationships.
  - **Relationship**
    Represents all USM-supported BinaryRelationships.
  - **Alert**
    Represents all USM alerts that a domain manager generates.
- **Properties**
  Stores the *attribute=value* pairs that the connector returns. The Properties element is of type KeyWordValuePairs.
- **Details**
  (Optional) Provides details on the USM entity being returned.

EntityChangeEventPublisher Interface

The com.ca.connector.runtime.EntityChangePublisher interface contains the method declarations necessary to subscribe to changes in domain manager data. This interface contains the following methods:

- public synchronized String subscribeToChanges(DataObject *selector*, EntityChangeSubscriber *subscriber*);
  Subscribes to entity change events. Returns a String containing a subscription ID which may later be used to unregister the subscription.
  After the data is retrieved and returned to CA SOI in a synchronous manner, using the get() method, it is desirable to keep the domain manager view of that data in synch with the CA SOI view. To provide this capability, a connector must implement the subscribeToChanges() method. This method launches a thread to provide asynchronous updates to the IFW through a callback for any changes to the domain manager data. The method accepts two parameters as input: a DataObject selector (SiloDataFilter) and an EntityChangeSubscriber object. To manage subscriptions, the connector creates an instance of EntityChangeSubscriptionManager. The incoming subscription is added to the manager instance, which the connector can then use to return changed data to CA SOI.
  - **selector**
    Defines a filter on entities to which to subscribe.
  - **subscriber**
    Defines the subscriber to entity change events and returns the subscription ID.
- public synchronized void unsubscribeFromChanges(String *subscriptionId*);
  Unsubscribes from entity change events.
  - **subscriptionID**

    Defines the subscription ID to unsubscribe.

ConnectorEventPublisher Interface

The com.ca.connector.runtime.EventPublisher interface contains the methods that control how the connector interacts with generic events not related to CIs from the domain manager. This interface includes the following methods:

> **WARNING**
> While the connector framework supports this interface, CA SOI does not currently use the functionality provided by these methods in integrated connectors. While these methods are available for implementation, the Sample connector does not contain sample code or implement any of these methods by default.

- **public Object getEarliestAvailableBookmark;**
  Gets a unique bookmark associated with the earliest event available to the connector.
- public String subscribeToEvents(String *eventTypeID*, DataObject *filter*, ConnectorEventSubscriber *subscriber*);
  Subscribes to general events published by the connector.
  - **eventTypeID**

Defines the ID of a type of event published by the connector.

– **filter**

Defines restrictions of events to which the connector subscribes. The object type is the filter type supported by the event type in the eventTypeID parameter.

– **subscriber**

Defines the subscriber to connector events and returns a subscription ID.

- public String subscribeToEvents(String *eventTypeID*, DataObject *filter*, Object *bookmark*, ConnectorEventSubscriber *subscriber*);
  Subscribes to general events published by the connector. This variant of the subscription method lets subscribers ask for events (that the connector may have missed, for example) generated since a previously received event. The start event is identified by an ID associated with the event, or a bookmark (see the ConnectorEventSubscriber API for details). The bookmark is opaque to the subscriber. This mechanism is derived from WS-MAN and provides some level of support for recovery.
  – **eventTypeID**

  Defines the ID of a type of event published by the connector.

  – **filter**

  Defines restrictions of events to which the connector subscribes. The object type is the filter type supported by the event type in the eventTypeID parameter.

  – **bookmark**
  Identifies a previously received event. Any event that occurred after the bookmarked event is regenerated for the subscriber. Subscribers use a special value of the bookmark to request all available events returned by the getEarliestAvailableBookmark() method.

  – **subscriber**

  Defines the subscriber to connector events and returns a subscription ID.

- public void unsubscribeFromEvents(String *subscriptionID)*;
  Subscribes to entity change events.
  – **subscriptionID**

  Defines the subscription ID to unsubscribe.

EntityOperationRunner Interface

The com.ca.connector.runtime.EntityOperationRunner interface contains the method declarations to implement custom operations for specific entities within the domain manager. All methods in this interface are optional.

> **WARNING**
> While the connector framework supports custom operations, CA SOI does not currently use custom operations in integrated connectors. While these methods are available for implementation, the Sample connector does not contain sample code or implement any of these methods by default.

This interface contains the following methods:

- public DataObject execute(String *operID*, DataObject *entityKey*, DataObject *operParams*) throws UCFException;
  Executes a synchronous operation against managed entities and returns the result of the operation.
  – **operID**

  Defines the name of the operation.

  – **entityKey**

  Defines the entity instances to which the operation applies.

  – **operParams**

Defines the operation input parameters.

- public String start(String *operID*, DataObject *selector*, DataObject *operParams*, OperationListener *listener*, String *refID*) throws UCFException;
  Starts an asynchronous operation against managed entities and returns an operation unique instance ID that you can use to refer to the operation.
  - **operID**

    Defines the name of the operation.
  - **entityKey**

    Defines the entity instances to which the operation applies.
  - **operParams**

    Defines the operation input parameters.
  - **listener**

    Listens for state changes related to the operation instance. This parameter can be null, and a listener can also register for the operation later.
  - **refID**

    Defines an ID understood by the caller.

ConnectorOperationRunner Interface

The com.ca.connector.runtime.ConnectorOperationRunner interface contains the method declarations to support connector custom operations for specific entities within the domain manager. All methods in this interface are optional.

> **WARNING**
> While the connector framework supports custom operations, CA SOI does not currently use custom operations in integrated connectors. While these methods are available for implementation, the Sample connector does not contain sample code or implement any of these methods by default.

This interface contains the following methods:

- public DataObject execute(String *operID*, DataObject *operParams*);
  Executes a synchronous operation against a connector and returns the result of the operation.
  - **operID**

    Defines the name of the operation.
  - **operParams**

    Defines the operation input parameters.
- public String start(String *operID*, DataObject *operParams*, OperationListener *observer*, String *refID*);
  Starts an asynchronous operation against the connector and returns an operation unique instance ID that you can use to refer to the operation.
  - **operID**

    Defines the name of the operation.
  - **operParams**

    Defines the operation input parameters.
  - **observer**
    Listens for state changes related to the operation instance. This parameter can be 'null' and a listener can also register for the operation later.
  - **refID**

    Defines an ID understood by the caller.

OperationRunner Interface

The com.ca.connector.runtime.OperationRunner interface contains the method declarations to monitor the progress of synchronous and asynchronous custom operations on the domain manager and connector. All methods in this interface are optional.

> **WARNING**
> While the connector framework supports custom operations, CA SOI does not currently use custom operations in integrated connectors. While these methods are available for implementation, the Sample connector does not contain sample code or implement any of these methods by default.

This interface contains the following methods:

- public void abort(String *operInstID*) throws Exception;
  Aborts the operation.
  - **operInstID**

    Defines the unique instance ID of a running operation obtained through an initial call to start().
- public void forget(String *operInstID*) throws Exception;
  Forgets the operation by indicating to the underlying system that the caller is no longer interested in the results, but that the operation does not necessarily need to be aborted.
  - **operInstID**

    Defines the unique instance ID of a running operation obtained through an initial call to start().
- public void recover(String *operInstID*, DataObject *lastState*, OperationListener *listener*, String *refID*) throws UCFException;
  Recovers an operation by restarting it from the state previously saved in an intermediateState() callback to the listener. Some operations may simply restart, others may abort as a result, and others may be able to restart from the obtained state.
  - **operInstID**

    Defines the unique instance ID of a running operation obtained through an initial call to start().
  - **lastState**
    Defines the last known operation state.
  - **listener**

    Listens for state changes related to the operation instance. This parameter can be null, and a listener can also register for the operation later.
  - **refID**

    Defines an ID understood by the caller.
- public void setOperationListener(String *operInstID*, String *refID*, OperationListener *listener*) throws UCFException;
  Listens to ongoing operation state changes. The interface supports a single listener per runner. If more are required, you can use a simple multiplexer pattern.
  - **operInstID**

    Defines the unique instance ID of a running operation obtained through an initial call to start().
  - **refID**

    Defines an ID understood by the caller.
  - **listener**

    Listens for state changes related to the operation instance. This parameter can be null, and a listener can also register for the operation later.

Sample Connector Interface Implementation

When you initialize the provided Sample connector, it does the following:

- Determines the names of the data files from the configuration properties and sends them to the constructor of the MockMDR instance
- Gets the configuration properties related to MockMDR
- Connects to MockMDR
- Starts an event simulation thread that periodically checks for changes in the sample-ci-changes.xml file and generates events to send to CA SOI

In response to the get() method call, the Sample connector filtering logic first reads all of the data from the XML file into memory and then applies the filter criteria to select the objects that qualify. This may not be practical for a custom connector with a very large number of objects. The connector should translate the filter into one or more native queries to directly select only the objects that satisfy the filter criteria.

The connector methods create(), update(), and delete() let CA SOI change the contents of the connector's data repository. This enables the connector's data repository to be synchronized to the reconciled entities persisted by CA Catalyst. In the Sample connector, there is only an in-memory data repository (list of CIs). The implementation of the create(), update(), and delete() methods make corresponding changes to the in-memory list of CIs.

To support the EntityChangeEventPublisher interface, the Sample connector makes use of a shared helper class to manage multiple CI change event subscribers (EntityChangeSubscriptionManager). The helper class dispatches events based on matching subscription filters.

Mock Domain Manager and Sample Data

A connector integrates with a domain manager to retrieve data. The Sample connector uses a mock domain manager, called the mock MDR, to simulate data creation and event generation. Even when you are building a custom connector to integrate with a real domain manager, in some cases you cannot automate unit tests with the actual domain manager. In this case, you can use the mock MDR interfaces to populate your connector with sample data for testing purposes.

The Sample connector contains the following interfaces for developing or using a mock domain manager:

- **com.ca.usm.ucf.MyMDR**
  Contains method declarations to implement a mock domain manager. This interface lets you switch between retrieving mock data and real data. Two implementations of MyMDR are included in the Sample connector: MockMDR and MyMDRImpl. Since the Sample connector always uses a mock domain manager (implemented in MockMDR.java), the implementation of retrieving data from the real domain manager is merely a placeholder (MyMDRImpl.java).
- **com.ca.usm.ucf.MockMDR**
  Contains method declarations for a full implementation of a mock domain manager by the Sample connector. The Sample connector uses this interface and the provided sample data as its primary data repository. You can use this class as the mock domain manager for your custom connector, but you must modify the methods in this class so that the mock domain manager and sample data mirrors the integration method and data of your domain manager.
- **com.ca.usm.ucf.MyMDRImpl**
  Contains method declarations for maintaining a shell of the real domain manager instance. All methods in this class throw an exception indicating that the operation is not supported because the Sample connector does not have a real domain manager. Any custom connector would have to replace the method implementation in this class with the code to access the real domain manager.

Using a mock domain manager in a custom connector is optional. If you want to do so, you can use the MockMDR interface to create a mock domain manager that simulates data from your domain manager. See the code in the MyMDR interface for an example of how the Sample connector uses MockMDR to create its data repository.

The Sample connector configuration contains a mockMode parameter that controls whether it is retrieving mock data. Because the Sample connector can only retrieve mock data, this parameter is true by default. If you create a mock domain manager for your custom connector, you can include this parameter as false by default, so that you can switch to mock data whenever necessary.

Create Connector Policy

Create a connector policy for your custom connector to transform data from the source domain manager format to the USM format and optionally to transform USM data back to the domain manager format to push changes to the domain manager.

Policy for each connector must be present in the SOI_HOME\resources\Core\Catalogpolicy directory for any processing to occur on objects retrieved from the domain manager. Without policy, domain manager data cannot be displayed in CA SOI.

For information about writing connector policy, including detailed syntax and examples for all available policy operations, see Writing Connector Policy.

All outbound policy must transform data to adhere to the USM schema for the data to appear correctly in CA SOI. For more information about the USM schema, see Unified Service Model. For more information about the types and properties supported in the USM schema, see the USM schema documentation.

Sample Connector Policy Implementation

The Sample connector implements the following policy files in the SOI_HOME\resources\Core\Catalogpolicy directory:

- **sampleconnector_policy.xml**
  Transforms outbound data from the source format defined in the Sample connector mock domain manager to the USM format. This is the outbound policy file that transforms all objects from the connector as outputs from the get(), create(), and update() methods. The Sample connector outbound policy leverages global policy, which eliminates the need for the connector to handle actions common to all connectors. For more information about global policy and how to use it in outbound policy, see Global Policy.
- **sampleconnector_policySB.xml**
  Transforms inbound data from the USM format to the source format defined in the Sample connector mock domain manager. This is the inbound policy file that transforms all objects stored in CA SOI as inputs to the create(), update(), and delete() methods.

The Sample connector supports both USM and non-USM formats and has a mixture of both entity types in the sample data files. Therefore, the provided connector policy handles entities that require transformation and entities that are already in the USM format and pass through the policy with no further processing required. The test data sets the eventtype property to USM-Entity to indicate that the entity is already in the USM format and does not need transformation.

The Sample connector sample data includes examples of several CI types. These types are covered in the connector policy. The mappings from the Sample connector types to USM types in the Sample connector policy are as follows:

Sample Connector Type	USM Type
System	ComputerSystem
Windows Server	BackgroundProcess
Database	DatabaseInstance
Disk Partition	File
LAN Interface	InterfaceCard
Processor	Processor
Router	Router
Service	Service

The Sample connector supports the native CI types listed in the Sample Connector Type column. However, it supports any USM CI type if the data is in USM format for outbound and inbound operations.

You can use the provided Sample connector policy as a point of reference or the framework from which you create a policy for your custom connector.

Outbound Policy Example

The following policy example shows how the provided Sample connector outbound policy transforms sample data of the System class to the ComputerSystem USM CI type:

```
<EventClass name='Item' >
  < Classify>
    <Field input='class' pattern='^System$' output='eventtype'
    outval='ComputerSystem' />
  </Classify>
```

The classify operation classifies CIs with a class of System to the USM type of ComputerSystem. This operation directs all specialized processing to occur under the policy for the ComputerSystem event class. The object also inherits policy operations from the parent Item event class.

```
  < Normalize>
    <Field output='MdrProdInstance' outputtype='ref' type='map'
    input='MdrProdInstance' >
      <mapentry mapin='^MdrProdInstance$' mapout='{fqdn(localhost)}' />
      <mapentry mapin='localhost' mapout='{fqdn(localhost)}' />
    </Field>
  </Normalize>
```

The normalize operation in the generic Item event class derives the name of the local host to include as the source of the information.

```
  < Format>
    <!-- Five digit CA Product identifier as defined by the USM
    MdrProductEnum -->
    <Field output='CAProductIdentifier' format='09998' input='' />
    <Field output='MdrProduct' format='CA:{0}' input='CAProductIdentifier' />
    <!-- MdrProdInstance setup above in normalize section
    <Field output='MdrProdInstance' format='{0}' input='{fqdn(localhost)}' /-->
    <Field conditional='id' output='MdrElementID' format='{0}' input='id' />
    <Field output="UrlParams" format="http://{0}:8080?id={1}"
    input="MdrProdInstance,MdrElementID" />
    <Field conditional='description' output='Description' format='{0}'
    input='description' />
    <Field conditional='name' output='Label' format='{0}' input='name' />
  </Format>
```

The format operations in the generic Item event class add the '09998' identifier to any data retrieved from the Sample connector. Every connector has a similar identifier that is included in the final USM entity. Other operations configure the basic CI properties related to the data source.

```
<EventClass name='ComputerSystem' extends='Item' >
  < Format>
  <!-- Assign class name -->
    <Field output='ClassName' format='ComputerSystem' input='' />
    <!-- Correlatable properties, must populate at least one -->
    <Field conditional='dnsname' output='PrimaryDnsName' format='{0}'
    input='dnsname' />
    <Field conditional='sysname' output='SysName' format='{0}' input='sysname' />
```

```
    <Field conditional='macaddress' output='PrimaryMacAddress' format='{0}'
    input='macaddress' />
    <Field conditional='ip_address' output='PrimaryIPV4Address' format='{0}'
    input='ip_address' />
    <!-- Non-Correlatable properties -->
    <Field conditional='name' output='ComputerName' format='{0}' input='name' />
  </Format>
```

The format operation in the ComputerSystem event class converts the properties from the source Sample connector CI format to the USM format. Properties such as DNS name, sysname, and MAC address are converted to the standard USM format. After all format operations complete, the CI is a USM entity.

Inbound Policy Examples

The following policy example shows how the provided Sample connector inbound policy transforms USM data of the Router type class to the Router class specifications in the Sample connector mock domain manager:

```
< EventClass name='Item' >
  < Classify>
    <Field input='ClassName' pattern='^Router$' output='eventtype'
    outval='Router' />
  </Classify>
```

The classify operation classifies CIs with a USM type of Router to the Sample connector class of Router. This operation directs all specialized processing to occur under the policy for the Router event class. The object also inherits policy operations from the parent Item event class.

```
  < Format>
    <Field input='LastModActivity' format='{0}' output='action' />
    <Field output='CAProductIdentifier' format='09998' input='' />
    <Field conditional='MdrElementID' input='MdrElementID' format='{0}'
    output='id' />
    <Field conditional='Description' input='Description' format='{0}'
    output='description' />
    <Field conditional='Label' input='Label' format='{0}' output='name' />
    <!-- correlated properties -->
    <Field conditional='SysName' input='SysName' format='{0}' output='sysname' />
    <Field conditional='PrimaryDnsName' input='PrimaryDnsName' format='{0}'
    output='dnsname' />
    <Field conditional='PrimaryMacAddress' input='PrimaryMacAddress' format='{0}'
    output='macaddress' />
    <Field conditional='PrimaryIPV4Address' input='PrimaryIPV4Address'
    format='{0}' output='ip_address' />
  </Format>
```

The format operations in the generic Item event class convert USM properties to the source format in the Sample connector mock domain manager.

```
  < Write>
    <Field type='file' name='outfile' properties='*' />
    <Field type='publishcache' properties='name,description,id,action,sysname,
    dnsname,macaddress,ip_address' />
  </Write>
</EventClass>
```

The write operation in the generic Item event class writes the derived properties to the output CI. The CI is then processed further by the specialized policy under the Router event class.

```
<EventClass name='Router' extends='Item' >
  < Format>
    <Field output='class' format='Router' input='' />
    <Field conditional='AdministrativeStatus' input='AdministrativeStatus'
    format='{0}' output='adminstatus' />
    <Field conditional='IsInMaintenance' input='IsInMaintenance'
    format='{0}' output='maintenanceFlag' />
  </Format>
```

The format operation in the Router event class converts the class property in the output CI to Router, the AdministrativeStatus property to adminstatus, and the IsInMaintenance property to maintenanceFlag.

```
  < Write>
    <Field type='file' name='outfile' properties='*' />
    <Field type='publishcache' properties='class,adminstatus,maintenanceFlag' />
  </Write>
</EventClass>
```

The write operation in the Router event class writes the final CI to an outfile for transmission to the Sample connector mock domain manager sample data.

In addition to this example, all inbound policy must include an event class of SiloDataFilter. All DataObject instances passed to the create(), update(), and delete() methods are transformed before the methods are invoked. Since the top-level type name is SiloDataFilter for the delete() method, this event class is used as the value for the property eventtype when the filter properties are converted. Therefore, SiloDataFilter must match the name of an event class. The Sample connector policy implements this event class as follows:

```
<EventClass name='SiloDataFilter' >
    <!-- properties are as follows (with example below): -->
    <!-- + id (xs:string) -->
    <!-- + updatedAfter (xs:dateTime) -->
    <!-- + entitytype (xs:string) Item, Relationship, or Alert -->
    <!-- + itemtype (xs:string) -->
    <!-- + recursive (xs:boolean) -->
    <!-- Format>
       <Field output='mdrelementid' input='id' format='{0}' />
    </Format-->
    < Write>
      <Field type='file' name='outfile' properties='*' />
      <Field type='publishcache' properties='id,updatedAfter,
      entitytype,itemtype,recursive' />
    </Write>
</EventClass>
```

USM Metrics Support

The Sample connector provides an example of how to support USM metrics. It implements the following operations as defined in USM:

> **NOTE**
> CA SOI currently does not support viewing metric information (that connectors collect) in its user interface.

- **GetAvailableMetrics**
  Retrieves the metrics that can be queried as follows:

  - Identifies the CI for which metrics are requested by its MdrProduct, MdrProdInstance, and MdrElementID values
  - Returns all possible metrics (standard and domain-specific) for the CI and its domain-specific identifier for each metric
- **GetMetricsValues**
  Determines the metric values for the supplied metric names. Metric values can be either the historical values or the current live value.

The operations are implemented as custom operations using the execute() method of the EntityOperationRunner interface.

> **NOTE**
> For more information about the USM metric collection capabilities, see the USM schema documentation.

Metric Data Repository

The SOI_HOME\resources\SampleConnector\data\sample-metrics.xml file is the data repository for metric data. You can add new metric definitions and values to the file, and the Sample connector uses all metric definitions and values for sample metric collection.

The following example shows how metrics are defined in the sample-metrics.xml file:

```
<silodata xsi:type="usm:SiloData">
   <properties>
       <property name="MdrElementID" value="3" />
       <property name="MetricName" value="total_accounts" />
       <property name="MetricDescription" value="total_accounts" />
       <property name="MetricType" value="Gauge" />
       <property name="MetricUnitDefinition" value="Number" />
       <property name="MetricDataType" value="Int" />
       <property name="IsOnlyLiveData" value="false" />
       <property name="MinGranularityInSecs" value="3600.0" />
       <property name="Value" value="100" />
       <property name="Timestamp" value="2010-08-05T11:00:00-06:00" />
       <property name="Value1" value="200" />
       <property name="Timestamp1" value="2010-08-17T11:00:00-06:00" />
   </properties>
</silodata>
```

This syntax defines the total_accounts sample metric as follows:

- It is a gauge metric type and integer data type expressed as a number.
- It tracks historical data (because the IsOnlyLiveData property is false).
- It has two historical value entries with corresponding timestamps.

To indicate a historical data metric, define multiple Value properties incremented by integers, and define a timestamp for each value.

Connector Configuration Files

Contents

To deploy and use a custom connector in the CA SOI IFW, you must define a configuration file for the connector in the SOI_HOME\resources\Configurations directory. This configuration file is an XML file that helps perform various tasks similar to any other connector configuration file. For example, it contains data pertaining to how to connect to the domain manager, the location of the connector class, the name of the connector, and so on.

You can create the connector configuration file manually by following the template file *myMDR_template.xml* provided with the Sample connector installer kit. Comments marked within this template file explain the properties that help you create your configuration file. You must create a copy of the template file, edit it as required, and rename it based on the naming convention (<MdrProduct>_template.xml). Additionally, while creating your configuration file, you must consider the following points:

- Some entries in the configuration files begin and end with the @ symbol. These are tokenized fields that the installer replaces with data that the user provides during installation. Any data that you want to collect from the user must have an entry with a tokenized value in the configuration file.
- Connector configuration files must begin and end with the main ConnectorConfig tag. Additionally, files must include the following tags and properties:
  - Must begin and end with the <Silo> tag
  - Must contain the fields *State* and *name*
  - Must contain the <ImplementationClass> tag and *name* and *policy* fields
  - Must contain the <ConnectionInfo> tag
  - Must contain the <ConnectorControls> tag
- The installer does not change the string literal values in the template file; therefore, they remain *as is* after installation. You can change these values while creating the configuration file, by directly editing the file after the connector installation, or using the Administration UI.

During installation, the installer replaces the word *template* in <MdrProduct>_template.xml with the location of the domain manager server to which the connector is pointing (MdrProdInstance). For example, the sample template is called *myMDR_template.xml*. During installation, the user is prompted for the location of the domain manager server. If the user enters the server name as myServerX, the configuration file is renamed to myMDR_myServerX.xml.

> **NOTE**
> The connector configuration file becomes available under the SOI_HOME\ resources\Configurations folder when the connector installation is done.

Create the Connector Configuration File

To create the connector configuration file, you must use the template file *myMDR_template.xml* provided with the Sample connector installer kit. The template file includes various comments about properties that help you create your configuration file.

Follow these steps:

1. Unzip the file *connector-installkit.zip* available at SOI_HOME\SampleConnector\InstallKit.
   The connector-installkit folder is extracted into the SOI_HOME\SampleConnector\InstallKit folder.
2. Open the ..\connectors\resources\ConnectorConfigTemplates folder, and locate the template XML file *myMDR_template.xml.*
3. Create a copy of the template file, rename the copied file (for example, <MdrProduct>_template.xml), and open the renamed file in a text editor.

   > **NOTE**
   > During installation, the installer replaces the word *template* in <MdrProduct>_template.xml with the location of the domain manager server to which the connector is pointing (MdrProdInstance).

   The XML file opens in the text editor.
4. Complete the following parameters, and save the file:

   > **NOTE**
   > Notice that several properties already have default tokenized values in the template file. These properties typically require custom values that users supply during installation.

   - **MdrProdInstance**

Identifies the domain manager instance. For example, ABC123.ca.com.
This parameter is helpful when you have multiple instances of a product installed in your enterprise. You can enter a tokenized value so that the user can define the domain manager instance during installation.

– **MdrProduct**
Specifies a unique identifier (as defined by the open enumeration, MdrProductEnum) naming the domain manager. For example, CA:09998.

– **name**
Specifies the name as <MdrProduct>_<MdrProductInstance>. For example, CA:09998_ABC123.ca.com.

– **State**
Specifies whether the connector is set to enabled or disabled state. Enabled and notEnabled are the valid values.

– **ImplementationClass**
Specifies the following parameters in the ImplementationClass section:

- **descriptorClass**
  Specifies the implementation of the com.ca.connector.metadata.ConnectorDescriptor interface for the connector. Set to com.ca.usm.ucf.utils.USMBaseConnectorDescriptor.
- **name**
  Specifies the name of the Java class for the connector that implements the com.ca.connector.runtime.Connector interface.
- **policy**
  Specifies the policy file used to transform raw data collected from a domain manager into USM data.
- **sbpolicy**
  Specifies the policy file used to transform USM data into domain manager format for create, update, or delete requests.

– **ConnectionInfo**
Specifies any configuration values that you must make available to the connector. Any attribute and its assigned values listed in this tag are passed to the initialize method of the connector when the connector starts. These properties often required tokenized values that a user must supply during installation.

– **EncryptedProperties**
Specifies which of the attributes listed in ConnectionInfo must be stored and treated as encrypted values. It includes the following parameter:

- **name**
  Specifies the comma-separated list of attribute names.

– **ConnectorControls**
Specifies the following set of Boolean flags that control various aspects of how the framework uses the connector:

> **NOTE**
> 1 and 0 are the two valid values for these flags. 1 represents that the flag is turned on and 0 implies that it is turned off.

- **dns_resolution**
  Specifies whether to use DNS resolution to resolve device names. If a reliable DNS mechanism is not in place (for example, no DNS server on the network, or CIs not defined to the DNS), disable DNS lookups to prevent CI resolution and normalization failure.
  Default: 1
- **getCIsAtStartUp**
  Specifies whether to rediscover CIs every time the connector starts. You typically enable this control so that your connector always provides a current record of all CIs from their domain managers. Turn this control off if the connector does not support collecting CIs at startup, which applies for Level 1 and 2 integrations.
  Default: 1
- **getRelationshipsAtStartup**

Specifies whether to rediscover relationships every time the connector starts. You typically disable this control so that relationships are only obtained and imported as a part of service model imports. You should only enable this control if you require relationship CIs outside of imported service models.
Default: 0

- **isRemotable**
  Specifies whether to allow the connector framework to access the connector remotely for create, update, and delete operations on the source domain manager. Only enable this control if you have coded the create(), update(), and delete() methods for inbound to connector operations.
  Default: 1
- **performDeltaProcessing**
  Specifies whether to process and publish deltas on CIs between the time the connector or SA Manager was last stopped or restarted. When enabled, this setting also performs delta processing on relationships if the getRelationshipsAtStartup property is enabled.
  Default: 1
- **useAlertFilter**
  Specifies whether to filter alerts based on their existence in a managed service in CA SOI. If the control is turned on, the connector only sends domain manager alerts that are associated with a CI that is part of an existing managed service in CA SOI. If the control is turned off, the connector forwards all alerts from the domain manager, regardless of whether they relate to a service.
  Default: 1
- **useServiceFilter**
  Specifies whether to send all relationships to CA SOI or only the ones associated with modeled services. Set the control to true to run relationships through a service filter and receive only the relationships associated with modeled services.
  Default: 1

– **LICURLS**
Helps CA SOI to allow launch of web-based UIs in context of an Item or Alert. A connector can specify URL information for any web-based UIs that the domain manager provides. You can then launch the URL in context of an Item or Alert from the CA SOI user interfaces. The URL parameter under LICURLS defines a single launch-in-context URL to be displayed in the CA SOI UI context menus. You can define multiple URL entries by including multiple configuration entries. URL includes the following parameters:

- **Type**
  Specifies whether a URL is associated with an Item or an Alert. Possible values of Type are Item or Alert.
- **seqNum**
  Specifies the identifier for the URL and the order in which URLs are displayed. For a connector, each URL of a specific Type must have a unique seqNum.
- **url**
  Specifies the URL string to launch. URLs are formed using substitutable parameters. Typically, you can include any property of an Item in the URL. To specify a substitutable parameter, enclose it within {}.
  Examples are url={Protocol}://{host}:{Port}/CIPortal?id={MdrElementID}, url =http://{host}:7070/sam/ui?parm={UrlParms}, url=http://www.ca.com, url={UrlParams}, or url=http://www.google.com/search?q={MdrElementID}.
- **Protocol**
  (Optional) Identifies the protocol portion of the URL (if needed by the URL). For example, Protocol=http or Protocol=https.
- **host**
  (Optional) Identifies the host name portion of the URL (if needed by the URL).
- **Port**
  (Optional) Identifies the port number to use in the URL (if needed by the URL). For example, Port="8080.
- **Label**

Specifies the string that the CA SOI user interface displays. This can be any string that easily identifies the URL being launched. For example, Label=Spectrum IM OneClick UI.

– **BindingProtocol**
  Specifies the transport protocol on which connectors are published as a service. It includes the following parameter:
  - **supportedProtocols**
    Specifies the supported protocols, which are jms and ws.
– **connector-type-meta-data**
  Specifies the connector properties describing the capabilities of the connector. These properties are common to all instances of the connector.
– **instance-meta-data**
  Specifies the connector properties specific to a particular instance of the connector.

The connector configuration file is created.

Connector Operations

Contents

You can classify connector operations in the following two phases:

Start Phase

During this phase, the connector performs the following operations:

- Performs the initialization task and connects to the domain manager.
- Retrieves all relevant objects from the domain manager for the CI import, including service CIs.
- Retrieves the open alerts for the relevant objects from the domain manager.

Run Phase

During this phase, the connector may receive requests from the SA manager (for example, for service imports or a list of currently open alerts), and needs to publish new CIs and alerts from the domain manager. You can achieve this task by active polling or a notification mechanism, depending on what the application provides.

In this phase, the connector performs the following operations:

- Requests for service import and publishes the complete service definition, comprising at least the service CI and relationship definitions for the included CIs.
- Requests for open alerts and retrieves the open alerts for the relevant objects from the domain manager.
- Adds a CI or service and retrieves a new object created in the domain manager and publishes the CI in CA SOI.
- Adds alerts and retrieves a new alert created in the domain manager and publishes it in CA SOI.

How to Test a Custom Connector

Contents

You can use a provided test harness to test your custom connector before you deploy it to a production CA SOI environment. The test harness uses your implemented connector, connector policy, data files, and a configuration file to operate the connector in a test environment. The test harness initializes the connector, invokes get, create, update, and delete operations, subscribes to CI change events, and waits for the connector to send events. All output objects are transformed into the USM format using the connector policy and validated in the USM schema. Any errors are reported in the console log and flagged as TestNG errors.

> **NOTE**
> You must install the Sample connector on the SA Manager to use the test harness.

Test Harness Method Implementation

You invoke the test harness from the Eclipse IDE. To implement the test harness, extend the abstract class com.ca.usm.ucf.ConnectorTest and implement the abstract methods in the subclass. The test harness base class provides a number of standard test cases to exercise the basic functions of your connector. These tests run by default. Your subclass also serves as an extension point to let you write more test cases for the specifics of your connector.

The Sample connector uses the SampleConnectorUTest class to implement the test harness. You can use this class as an example or as a starting point for writing custom connector test cases. The SampleConnectorUTest class implements the following methods:

- **public abstract void setupTest()**
  Reads any connector-specific property file to get configuration values, instantiate the connector, and assign it to the connector field. The test harness uses the connector instance provided to run tests against it.
  This method takes information from the usmtest.properties file by default, which is the single configuration file for the connector test cases. You can reuse this file when configuring tests for your custom connector by modifying the necessary properties to fit with your connector.
  This is the only required method that the connector's test class must implement. All other test harness methods are optional. The default implementation is provided by the ConnectorTest class, but you can override it in the connector's test class to modify the test harness behavior.
- public void beforeTestMethod(Method *method*)
  Defines any custom setup to perform before each test is invoked.
  - **method**
    Defines an object of a class java.lang.reflect.Method representing a test method to invoke. It contains information about the method, such as method name.
- public void afterTestMethod(Method *method*)
  Defines any custom setup to perform after each test is invoked.
  - **method**
    Defines an object of a class java.lang.reflect.Method representing a test method to invoke. It contains information about the method, such as method name.

Configure the Test Harness Properties

By default, the test harness uses the usmtest.properties file to determine the test cases and data. You can modify the usmtest properties file to configure tests for your custom connector or create a new usmtest properties file. The usmtest properties file must reside in the classpath for reading configuration values, but the SOI_HOME\SampleConnector\test \resources directory is the directory where files are included in the classpath by default. The property file must be present for the test harness to work correctly.

The following information must be defined in the properties file:

- **Configuration parameters**
  Defines the resources to use to perform the tests. The following parameters are included in the usmtest.properties file:
  - **data_source, alert_data_source, change_event_data_source**
    Define where the test harness retrieves data for the tests. The Sample connector usmtest.properties file contains separate properties for CIs (sample-cis.xml), alerts (sample-alerts.xml), and CI changes (sample-ci-changes.xml).
  - **policyFile**
    Defines the location and file name of the connector policy file that the test harness should use to test outbound operations.
  - **southboundPolicyFile**
    Defines the location and file name of the connector policy file that the test harness should use to test inbound operations.

The Sample connector includes other configuration parameters that the test harness uses. For more information, see the SOI_HOME\SampleConnector\test\resources\usmtest.properties file.

- **Test cases**
  Defines specific operations to test using specified data. The IFW calls the methods implemented by the connectors using filters, also called selectors. Selectors define what kind of data you want to retrieve or pass it as a parameter. You can define multiple test selectors, and each selector results in a call to a method identified by a selector. Each selector can have multiple properties (comma-separated key value pairs).The examples that follow describe some of the test cases included in the usmtest.properties file. For additional examples, see the usmtest.properties file.

Example: Testing the get() method

The test harness recognizes a selector for the get() method named "getSelector". You can run multiple instances of the test by appending a digit to the property name, such as getSelector1, getSelector2, and so on. The following example of a get() selector asks to return a ComputerSystem CI with an id value of 3 from the in-memory data repository of the Sample connector:

```
getSelector5="entitytype=Item,itemtype=ComputerSystem,id=3"
getSelector5.expected.datafile=ComputerSystem_3.txt
getSelector5.compare.ignorecase=true
getSelector5.compare.locatorfields=MdrElementID
getSelector5.compare.ignorefields=CreationTimestamp,LastModTimestamp,
MdrProdInstance,UrlParams
getSelector5.expected.result=true
```

The first line in the example instructs the test harness to look for a Computer System CI with an id value of 3. The second line specifies that the data returned by the get() method should look like properties and values listed in the ComputerSystem_3.txt file. The expected.result value of true indicates that the test is expected to be able to correctly find the item and that the actual returned data matches the expected data.

The locatorfields property lists properties that the test harness can use to speed up the lookup process, while the ignorefields property lists properties to ignore in the comparison between actual and expected data.

Example: Testing the create() method

Specifying selectors for create() and update() methods is very similar. The following example specifies to create the objects located in the SOI_HOME\SampleConnector\test\resources\SampleConnectorCreateValidDataFile.txt file:

```
create1.input.datafile=SampleConnectorCreateValidDatafile.txt
create1.compare.ignorecase=false
create1.compare.ignorefields=CreationTimestamp,LastModTimestamp,MdrProdInstance,
UrlParams,MdrElementID,id,outfile,entitytype
```

Several CIs exist in the data file that the test harness is expected to create. Both the create() and update() methods return a data object that has been created or updated. The test harness compares the property values from the returned objects with the values specified as input. The test harness ignores the list of properties defined in ignorefields when comparing between actual and expected data.

Example: Testing the delete() method

Each delete() method test requires a selector that is expected to match only one CI or none at all. The following example deletes a CI that the test harness has created in a previous operation:

```
delete1="entitytype=Item,id=SampleConnector-0"
```

Example: Performing negative tests

You can create negative test cases that are designed to test how the connector responds to failure situations.

The following example of a get() test intentionally specifies an invalid entitytype:

```
getSelector13="entitytype=Negative.Test"
getSelector13.expected.result=false
getSelector13.expected.exception=true
getSelector13.expected.exception.name=com.ca.ucf.api.UCFException
```

This test intentionally specifies an invalid entitytype of Negative.Test and defines the expected result as false. The test also specifies an exception that is expected to return.

The following example of a delete() test defines an invalid selector:

```
delete3="entitytype=Item,id=SampleConnector-NoNExIsTiNG"
delete3.expected.result=false
delete3.expected.exception=true
delete3.expected.exception.name=com.ca.ucf.api.InvalidParameterException
```

Test Harness Flow

You run the test harness using TestNG in Eclipse. The test harness follows a simple workflow that begins with any @BeforeClass annotated methods that signify execution before the connector class instantiation. This includes the initializeContainer() method, which calls the SampleConnectorUTest.setupTest() method to get the test configuration, starts the transformation engine, and initializes the connector according to the test configuration. Once the initializeConnector() method completes, the test harness is now connected to the data source according to the configured test properties in the usmtest.properties file.

The test harness calls each @Test annotated method in the appropriate order. This includes the following:

getTest()

Retrieves CIs according to the configured selector filters, prints out the name and value property pairs for each CI, converts the CIs to USM XML, and performs a USM validation against that XML.

- **testSubscribeCIChanges(), testCreateEvents(), testDeleteEvents(), testUpdateEvents()**
  The subscribeCIChanges() method listens for CI change notifications from the connector, and when events are received, they are added to create/delete/update queues. The test methods then extract the information from the associated queue, validate it, and publish it to the test console. The usmtest.properties file has more information about how to configure the number and type of events the test harness expects.
- **testCreate(), testUpdate(), testDelete()**
  Test writing to the connector by running create, update, and delete test cases in the usmtest.properties file.

The test harness tests all methods sequentially, requiring you to only write a single test class (SampleConnectorUTest, in the case of the Sample connector) and implement setupTest().

Run the Test Harness

You can run the test harness as a TestNG test in the Eclipse debugger.

Follow these steps:

1. Open Eclipse and set any initial breakpoints of interest in your connector and your connector test class.
2. Right-click your test class in the Project Explorer and select Debug As, TestNG Test.
   The test harness runs and displays the test results in the Console tab.

Custom Connector Logging

Contents

The IFW and all connectors use Apache log4j for logging and debugging messages. The default log4j.xml file is located at SOI_HOME\resources. This file is loaded at IFW startup and defines multiple loggers used by the IFW and a logger that connectors can use to direct their log and debug messages to a default log file.

Default Connector Log File

The SOI_HOME\resources\log4j.xml file directs connector logging information by default to the SOI_HOME\log\ifw.log file. To use this default log file, your connector package must start with "com.ca.usm.ucf." The default logger definition in the log4j.xml file is as follows:

```
<!-- Default connectors ->
<logger name="com.ca.usm.ucf" additivity="false"
   <level value="ERROR" />
   <appender-ref="IFW" />
</logger>
```

How to Create a Connector-Specific Log File

You can define an additional log4j appender and logger for your connector so that it uses its own dedicated log file and sets log levels independently of other connectors on the same system.

Follow these steps:

1. Create a connector-specific log4j.xml file with an appender and logger formatted as follows:
   Appender:
   ```
   <appender name="TemplateConnector" class="org.apache.log4j.RollingFileAppender">
       <param name="File" value="&logDir;/Template_Connector.log"/>
       <param name="Append" value="true"/>
       <param name="MaxFileSize" value="20MB"/>
       <param name="MaxBackupIndex" value="10"/>
       <layout class="org.apache.log4j.PatternLayout">
           <param name="ConversionPattern" value="&filePattern;"/>
       </layout>
   </appender>
   ```
   Logger:
   ```
   <logger name="com.ca.usm.TemplateConnector">
       <level value="ERROR" />
       <appender-ref ref="TemplateConnector" />
   </logger>
   ```
   Adhere to the following conventions for overall log4j file conventions and appender and logger properties:
   - **log4j file name and location**
     Use a unique and easily identifiable name, such as [set to your product name]Connector_log4j.xml. Install the [set to your product name]Connector_log4j.xml file to the following location: SOI_HOME\Configurations\log4j.
   - **appender name**
     Use a name unique to your connector, such as [set to your product name]Connector.
   - **logger name**
     Use a name that makes the logger unique to your connector, such as com.ca.[set to your product name].
   - **log file name and location**
     Use a name that adheres to the following convention: [set to your product name]_connector.log. Install the [set to your product name]_connector.log file to the following location: SOI_HOME\log.

   Do not define a ROOT logger in this configuration file.
2. Update the connector code to load the created log4j configuration file at connector startup by inserting code similar to the following the Initialize() method:

```
if(System.getProperty("IFW_RESOURCES")!=null)
DOMConfigurator.configureAndWatch(System.getProperty("IFW_RESOURCES")+File.separator+
"resources"+File.separator+"Configurations"+File.separator+"log4j"+File.separator+
"ProductNameConnector_log4j.xml",30000);
```

This code should be the first called after the connector is instantiated so that the log file can capture all relevant information.

3. Package and install the connector-specific log4j configuration file with the connector.
   For more information about using the provided installer kit to add and install the connector-specific log4j configuration file to the appropriate location, see Custom Connector Deployment.

Custom Connector Deployment

Contents

You can deploy a custom connector either manually or by using an installer.

How to Create a Custom Connector Installer

For properly deploying a custom connector, use the provided installer kit to create an installer, package the appropriate files, and put all files in the right places. Connector materials that you need to package are as follows:

- Connector-specific binaries such as executables, java classes and jar files, and libraries that make up the connector
- Connector-specific dependencies such as third-party utilities and product-specific APIs, which the connector uses at runtime
- Connector policy files
- Connector configuration file

The installer kit files are located at SOI_HOME\SampleConnector\InstallKit. Unzip the file in this directory to gain access to all materials necessary for generating an installer, including a custom InstallAnywhere project. The installer kit helps you create an installer that does the following:

- Installs the connector materials to the appropriate places.
- Installs the IFW on a system where it does not exist.
- Populates the connector configuration file that you created based on information provided during installation.

The kit uses InstallAnywhere to build a fully functional installer for your connector. Use the following InstallAnywhere connector installer template file to create the installer: SOI_HOME\SampleConnector\InstallKit\InstallProject \Connector_Template\Connector_Template.iap_xml.

> **NOTE**
> Previous experience building InstallAnywhere installers is necessary to use the materials in this kit. If you do not have the necessary experience, consider manually deploying your custom connector.

How to Manually Deploy a Custom Connector

Follow these steps for manually deploying a custom connector on a computer where the IFW is already available:

1. Compile the connector code into a jar file.
2. Stop the CA SAM Integration Services service on the connector system.
3. Copy all connector files to their appropriate places manually:
   - Connector-specific binaries, such as executables, Java classes, jars, and libraries, that make up the connector:
     a. User-built executables and Java classes at SOI_HOME\bin

   b. User-built jars and libraries at SOI_HOME\lib
   – Connector-specific dependencies, such as third-party utilities, product-specific APIs, and so on, that the connector uses at runtime:
      a. Domain-specific APIs at SOI_HOME\lib\[set to your product name]
      b. Third-party utilities at SOI_HOME\lib\Common
   – Connector policy at SOI_HOME\resources\Core\Catalogpolicy
   – Connector configuration file at SOI_HOME\resources\configurations
4. Change the connector configuration file name manually to use the domain manager instance (that is, *MDRID*_server5.xml).
5. Edit the connector configuration file manually to replace the tokenized values with custom ones for integrating with the data source.
6. Restart the CA SAM Integration Services service.

Creating Connector Policy

Contents

Connector policy defines the conventions the connectors use to classify, filter, parse, normalize, enrich, and format CIs, relationships, and alerts received from a domain manager and send them in a specified format to the SA Manager for display in the Operations Console. Policy is provided for all connectors, and the default policy transforms all CA SOI entities so that they are compliant with the USM schema. Additional policy may also be provided for connectors that support create, update, and delete operations in the domain manager product.

The IFW uses an event-based mechanism to retrieve and process CIs, relationships, and alerts from connectors. Therefore, the term *event* in this section refers to any object retrieved from a domain manager's connector (including actual events).

You can write new policy or customize existing policy.

New Policy

All connectors require a policy to process events received from domain managers and sent to the SA Manager. Without a policy, the IFW does not receive instructions about how to process an event, and no processing takes place.

If you create a custom connector for a Level 4, 5, or 6 integration, set up a policy to process events from its domain manager and perform operations on events in the domain manager, if the connector supports these operations.

Policy Customization

You can change certain policy settings for processing CIs and events from a domain manager. For example, classification policy may require a new event classification category for each existing class to accommodate an updated integrated product release. Or, you may want to place limits or remove existing limits on the data that an existing connector sends to CA SOI.

For Level 2 integrations using a generic connector, customize a template policy file so that the policy appropriately processes data from the data source. For example, you can customize the policy provided with the SNMP connector to collect and process traps from a mainframe product.

Policy Types

You can create or customize the following types of connector policy:

- **Outbound from connector policy**

Transforms data from the source domain manager format to USM format. Outbound from connector policy converts domain manager data into a format that CA SOI can understand and display. All connectors have outbound from connector policy. Outbound from connector policy files use the following naming convention:

```
connectorname_policy.xml
```

Example: sampleconnector_policy.xml

- **Inbound to connector policy**
  Transforms data from the USM format to the source domain manager format. Inbound to connector policy facilitates inbound operations, where connectors make changes to source domain manager data to synchronize data across domains. Not all connectors have inbound to connector policy, and not all connectors support inbound to connector operations. Inbound from connector policy files use the following naming convention:

  ```
  connectorname_policySB.xml
  ```

  Example: sampleconnector_policySB.xml

The structure, available functions, and available operations are the same for both types of policy. The only difference is the source format and the format to which the policy transforms the data.

In the <Field> elements inside the DO NOT USE section, inbound policy has USM properties as input and domain-specific properties as output, whereas outbound policy has domain-specific properties as input and USM properties as output.
In the <Field> elements inside the <Write> section, inbound policy lists domain-specific properties to pass to the source domain manager.

Inbound to Connector Policy Considerations

Inbound to connector policy transforms data from the USM format to the format of the source domain manager for inbound to connector operations. Inbound to connector policy is not required and only works under the following circumstances:

- The inbound to connector policy file must contain an <EventClass> definition of SiloDataFilter. Inbound policy transforms data before passing it to the methods that invoke synchronization. The delete() method specifically uses SiloDataFilter as the top level class, so this class must exist in the inbound policy file for the delete operation to work. Include this event class as follows:

  ```
  <EventClass name='SiloDataFilter' >
      <!-- properties are as follows (with example below): -->
      <!-- + id (xs:string) -->
      <!-- + updatedAfter (xs:dateTime) -->
      <!-- + entitytype (xs:string) Item, Relationship, or Alert -->
      <!-- + itemtype (xs:string) -->
      <!-- + recursive (xs:boolean) -->
      <!-- Format>
         <Field output='mdrelementid' input='id' format='{0}' />
      </Format-->
      < Write>
        <Field type='file' name='outfile' properties='*' />
        <Field type='publishcache' properties='id,updatedAfter,
        entitytype,itemtype,recursive' />
      </Write>
  </EventClass>
  ```

- The connector must support inbound to connector operations.

  NOTE
  For information about whether a specific connector supports inbound to connector operations, see the *Connector Guide* for that connector.

  Even if a connector does support inbound to connector operations and already contains an inbound to connector policy file that you want to customize, the connector may only support inbound operations for specific types. Find supported

inbound to connector CI types on the connector detail page in the Administration UI. The InboundToConnectorTypes field in the Connector Type Data table contains all supported CI types for inbound to connector operations.
If you are writing new policy for a custom connector, you must include the appropriate methods for inbound to connector operations when building the connector.

- The connector configuration file must include the appropriate file name for the inbound to connector policy file in the *sbpolicy* attribute.
  Each connector has a configuration file located at SOI_HOME\resources\Configurations.
  The following example shows the Sample connector policy file definition in its configuration file:
  ```
  <ImplementationClass descriptorClass="com.ca.usm.ucf.utils.USMBaseConnectorDescriptor"
  name="com.ca.usm.ucf.SampleConnector" policy="sampleconnector_policy.xml"
  sbpolicy="sampleconnector_policySB.xml"/>
  ```
  The connector configuration file assumes a connector policy file location of SOI_HOME\resources\Core\Catalogpolicy.
- The connector must have inbound to connector operations enabled.
  The isRemotable control defines whether inbound to connector operations are enabled for a connector. Verify that this control is enabled on the connector detail page in the Connector Controls table. The isRemotable control is enabled by default for most connectors.
- The CA Catalyst Synchronizer must be enabled.
  The CA Catalyst Synchronizer determines when synchronization is required and pushes required changes to the connectors. The Synchronizer is disabled by default, and you must enable it for any inbound to connector operations to occur.

  WARNING
  CA SOI supports only specific synchronization use cases. You should not enable the Synchronizer except as part of a supported use case. Connectors can support inbound to connector operations and contain inbound to connector policy, but support for the operations and policy may depend on whether they are part of a supported synchronization use case. For more information about synchronization and supported use cases, see the Synchronization section.

Global Policy

Global policy is an embedded set of policy instructions that handles elements and actions that are common for all connectors. Outbound from connector policy should leverage global policy to avoid having to redefine common elements and actions. Inbound to connector policy should not use global policy.

Leverage global policy in outbound policy files by writing the initial <Catalog> element as follows:

```
<Catalog version='1.0' globalextends='GLOBAL!'>
```

Global policy handles the following common outbound policy items:

- Class name assignment
  Example: <Field output='ClassName' format='ComputerSystem' input=''/>
- Instance name assignment
- Label assignment
- The entire <Write> element

Policy Structure and Deployment

Separate XML files hold a connector policy for each connector. The connector policy installs in the following location:

SOI_HOME\resources\Core\CatalogPolicy

You have one outbound policy and one optional inbound policy for each connector.

You encapsulate the entire policy file by a single <Catalog> element. In outbound policy files, this element should include the global policy definition.

Event Classes

An event class represents a container for all processing operations related to a certain type of event. The event type is determined by a special property named eventtype. The initial eventtype property is set in the connector-specific code.

The name attribute of each <EventClass> property in the catalog is matched to the eventtype property in the event and the operations of that class are carried out on the event. Next, the <Classify> property changes the eventtype to reference a more specific <EventClass>. Event classes are hierarchical, so you can create subclasses of a base event class to further classify events. For more information, see Classification Policy.

You can modify the eventtype property through a connector policy just like any other property in policy files. Changing the eventtype in connector policy files changes the event class to the new eventtype value.

Inbound to connector policy must contain an <EventClass> definition of SiloDataFilter at the end of the file for delete operations to work. For more information, see Inbound to Connector Policy Considerations.

Hierarchy and Inheritance

Connector policy is hierarchical, meaning that a child event class inherits all policy operations from a parent class. The following code fragment shows the OPR-DSMEVENT class inheriting all connector policy operations defined in the parent OPR-BASE class:

```
<EventClass name="OPR-BASE")
   <Classify ...../>
   <Filter ...../>
</EventClass>
<EventClass name="OPR-DSMEVENT" extends="OPR-BASE">
   <Parse ...../>
</EventClass>
```

In cases where a parent class and child class have similar operations, the parent operations are enacted first, followed by the child operations. For example, if a parent and child include a parse operation, the parent parsing occurs first, followed by the child parsing. You must understand this rule so that the connector policy you write is processed in the intended order. The following example shows this rule:

```
<EventClass name="OPR-BASE">
   <Parse>
       <Field input="tagA" pattern="^(\w+)-(\w+)$" output="tagA1,tagA2"/>
   </Parse>
</EventClass>
<EventClass name="OPR-DSMEVENT" extends="OPR-BASE">
   <Parse>
       <Field input="tagA1" pattern="^(\d\d)(\w+)$" output="tagA1a,tagA1b"/>
   </Parse>
</EventClass>
```

In this example, assume that tagA="12Buckle-Shoe". The property is parsed by the parent operation, which transforms the property into two properties, tagA1 (12Buckle) and tagA2 (Shoe). Afterwards, the tagA1 value, "12Buckle", is parsed by the child operation into two more separate properties, tagA1a (12) and tagA1b (Buckle).

There are no limits on the levels of inheritance or the number of inheriting children. All connector policy operations support inheritance except for classification.

Property Functions

In connector policy operations, various functions help transform or generate event properties such as date, time, and server name.

Use any functions in the function library for connector policy to convert tags to a specific output. Write these functions enclosed in curly brackets {} as part of the input attribute in a Field element.

You can use these functions in the input attribute of any operation. For example:

```
< Format>
<Field  output='' format='{0}' input='{somefunction(param1)}' />
</Format>
```

Functions with no parameters use the following syntax:

```
{function}
```

Functions with additional parameters use a different syntax as follows:

```
{function([param1,param2,param3])}
```

Examples of the functions that CA SOI includes are as follows:

- Host
  - {localhost}Returns the local hostname (fully qualified)
  - {ip(propname)} Dereferences the property (usually a host name) and converts to an IPV4 address
  - {fqdn(propname)}Dereferences the property (usually an IPV4 address) and converts to a fully qualified domain name
  - {convertHexToMac([propname,-])}Dereferences the property (a hexadecimal code for a MAC address) and converts to a delimited string using the second parameter delimiter character
- DateTime
  - {xsdateTime(now)}Returns a datetime stamp formatted in xs:dateTime (yyyy-mm-ddThh:mm:ss-zz:zz)
  - {xsdateTime(propname)}Dereferences the property (epoch time in seconds or milliseconds) and converts to xs:dateTime format
  - {convertxsdateTime([propname,MMM d yyyy K:mm:ss a])}Dereferences the property (a datetime formatted string), parses the property according to the second parameter, and converts to xs:dateTime
  - {datetime()}Generates a date and time string for the current date and time and results in the following output structure: March 12, 2008 1:31:00PM
  - {datetime(propname)}Generates a date and time string for the date and time represented by an event property value, where the property is a long integer representing some number of seconds since January 1, 1970
  - {timet()}Generates a long integer representing the current number of seconds since January 1, 1970
  - {timet(propname)}Generates a long integer representing the number of seconds since January 1, 1970, using the event property value, which must be a date and time string
- Date
  - {xsDate(now)}Returns a date stamp formatted in xs:date (yyyy-mm-dd-zz:zz)
  - {xsDate(propname)}Dereferences the property (epoch time in seconds or milliseconds) and converts to xs:date format
  - {convertxsDate([propname,MMM d yyyy K:mm:ss a])}Dereferences the property (a date time formatted string), parses the property according to the second parameter, and converts to xs:date
- Time
  - {xsTime(now)}Returns a time stamp formatted in xs:time (hh:mm:ss-zz:zz)
  - {xsTime(propname)}Dereferences the property (epoch time in seconds or milliseconds) and converts to xs:time format
  - {convertxsTime([propname,MMM d yyyy K:mm:ss a])}Dereferences the property (a date time formatted string), parses it according to the 2nd parameter, and converts to xs:time
- Array

  - {entry(propname)}References an entry in an array and returns the first property value in a list. Functions are available for entry1 and entry2 to reference values at different places in an array
- String
  - {replace([propname,ch1,ch2])}Replaces any character or string in a specified property with another character
    - propnameSpecifies the property to search for a character or string to replace
    - ch1Specifies a character or string to search for and replace in the specified property
    - ch2Specifies a replacement character or string
  - {toLower(propname)}Converts uppercase characters in the specified property to lowercase
  - {toUpper(propname)}Converts lowercase characters in the specified property to uppercase
- Other
  - {uniqueidentifier}Generates a SQL unique identifier, such as 61CD55D1-F142-2E04-8A2A-9667118CF65E
  - {prepareconsolidatefield}Parses combinations of event fields entered in consolidation operations

How to Add a Function

The SOI_HOME\resources\Core\Conf\tagfunctions.properties file stores the provided functions. You can create and add new functions to the function library for use in policy files.

Follow these steps:

1. Create a java class or method for the new function.

   NOTE
   The method must return a string and accept an array of strings.
2. Package the Java class into a .jar file and copy the file into the SOI_HOME\resources\Core\Bin directory.
3. Add the method or class to the functions.properties file following the conventions of the other entries.
4. Restart the CA SAM Integration Services service.

Policy Operations

Connector policy consists of operations that provide processing and transformation instructions for events retrieved from a specific domain manager. You can write connector policy using the following operations in the connector policy file for any event (CI, relationship, or alert) in a domain manager:

- Classify
- Parse
- Enrich
- Normalize
- Format
- Evaluate
- Filter
- Write

NOTE
All policies must use valid USM types, properties, and enumerated values defined in the USM schema documentation.

Leverage global policy in outbound from connector policy to automate common operation (such as Write) by writing the initial element in the file as follows:

```
<Catalog version='1.0' globalextends='GLOBAL!'>
```

For more information, see Global Policy.

Classify Operation

Contents

Classify operations refine an event class from the generic eventtype to more specific classifications. This operation enables specific connector policy to be enacted using different types of events from the same source. The eventtype value searches for a matching <EventClass> in the classification operation to classify an event. Each <EventClass> property can contain several subclasses.

Classify operations do not support inheritance, because classification makes an eventtype property more specific, where inheritance extends more general eventtype properties.

The processing engine traverses all field elements in order until a field is matched, after which no other fields are considered.

> **NOTE**
> Inbound to connector policy must include a specific <EventClass> definition for SiloDataFilter for subsequent delete operations to work. For more information and the detailed syntax for this <EventClass>, see Inbound to Connector Policy Considerations.

Classify Property Refines an EventClass

Classify operations begin with a <Classify> property. The <Classify> property refines an <EventClass>.

This property has the following format:

```
< EventClass name=>
   < Classify>
       <Field input= pattern= output= outval= />
   </Classify>
</EventClass>
```

- **name**
  Defines the name of the event class that you are using to create the connector policy.
- **input**
  Defines the value of the event property used for the classification.
- **pattern**
  Defines the regular expression value that the input value must match for an event to be matched.
- **output**
  Defines the assigned value for the outval attribute. Normally, the output is the eventtype.
- **outval**
  Defines the value assigned to the output. This is usually a more specific eventtype value.

Example: Classify Windows Event Log events into specific subgroups

The following example separates events received from the Windows Event Log event source that match the specified patterns into two subgroups: SYSLOG-SEC and SYSLOG-APP.

```
<EventClass name="SYSLOG">
   < Classify>
       <Field input="msg" pattern="^Sec.*$" output="eventtype"
       outval=SYSLOG-SEC" />
       <Field input="msg" pattern="^App.*$" output="eventtype"
       outval=SYSLOG-APP" />
   </Classify>
</EventClass>
```

The <Classify> property searches the message text (defined by the "msg" attribute) of events received from the Windows Event Log for words or strings beginning with Sec or App and classifies events that qualify into the more specific SYSLOG-SEC and SYSLOG-APP eventtypes. This property creates new Windows Event Log subgroups for events received from the application and security logs.

Parse Operation

Contents

Parse operations split event properties into additional properties using regular expression subgroups. For example, if an event source groups the old and new severity of a metric into one property, you can parse the severities into separate properties to make the information easier to understand.

Parse operations support inheritance, so you can parse operations that were created by parsing operations in higher levels. The framework traverses all parsing operation field elements in order from top to bottom until all field elements are processed. Matches are recorded and processed.

Parse Property Splits Event Properties

Parse operations begin with a <Parse> property. The <Parse> property splits event properties into additional properties using regular expression subgroups.

This property has the following format:

```
< EventClass name=>
   < Parse>
       <Field input= pattern= output= />
   </Parse>
</EventClass>
```

- **name**
  Defines the name of the event class that you are using to create the connector policy.
- **input**
  Defines the event property to parse into subgroups.
- **pattern**
  Defines the regular expression pattern that the input event property must match for the event to be parsed. The pattern is divided into subgroups designated by parentheses. If the input event property matches the pattern, it is separated into one property for each subgroup.
- **output**
  Defines the output properties to assign to the parsed input event property subgroups. The output properties correspond to the regular expression subgroups in the pattern. The first output property is assigned to the first subgroup, the second output is assigned to the second subgroup, and so on. Output property values may be new properties or existing properties.

Enrich Operation

Contents

Enrich operations look up additional properties from an external source using current event property values and create new event properties from the retrieved properties. For example, you may want to enrich an event received from CA NSM with contact information for the resource in WorldView or CMDB Configuration Item (CI) information.

You can write enrich operations based on the following types of transformation:

- Regular expression
- JDBC query
- Java method call
- Command line executable

Enrich operations support inheritance. These operations traverse all enrichment operation field elements in order, from top to bottom, until all field elements are processed. Matches are recorded and processed.

Enrich Property Create New Event Properties From Retrieved Properties

Enrich operations begin with an <Enrich> property. The <Enrich> property looks up additional properties from an external source using current event property values and creates new event properties from the retrieved properties.

This property has the following format:

```
< Enrich>
   <Field input= type= outputtype= [inputtype= connectionstring= jdbcdriver=
   query= returntype= column=]|[jclass= method=]|[cmdline=] output= />
       [<mapentry mapin= mapout=>]
</Enrich>
```

> **NOTE**
> Only the input, type, outputtype, and output attributes are required for all enrich properties. The other attributes you must use depend on the type of enrich operation you are writing. The type definition specifies the requirements for each enrichment type.

- **input**
  Defines a list of properties to enrich with information from external sources.
  If are using a single multiple-column property as input, enter the column values in a comma-delimited list using the following format:
  ```
  property_column_order
  ```
  - **property**
    Specifies the event property name.
  - **column**
    Specifies a column value from the column attribute.
  - **order**
    Specifies an integer, starting with 0, indicating the order in which the specific column value is returned.

  For example, if you are using a users property as input that is made up of column values firstname and lastname, the input property would read as follows:
  ```
  user_firstname_0,user_lastname_1
  ```
- **type**
  Defines the type of enrichment. You can have multiple fields of the same or different types within a single <Enrich> property with no restrictions. The following are available types:
  - **map**
    Matches and enriches properties using regular expressions. Map enrichment uses mapentry elements that represent an expression and an output to assign to the property if the expression is matched. These elements are read from top to bottom until a property matches an element, after which additional mapentries are not considered. The map type requires the following attributes:
    - mapin
    - mapout
  - **jdbc**
    Uses property values as input parameters in a JDBC query to determine an enriched value for the properties. The jdbc type requires the following attributes:

    - inputtype
    - connectionstring
    - jdbcdriver
    - query
    - returntype

    You can return multiple columns and rows from a complex JDBC query and use the column attribute in the policy to identify each of these returned columns uniquely.

    > **NOTE**
    > The jdbc enrichment type supports only the *pairedlist* output type. Therefore, the returned output value is always a paired comma-delimited list of values. This enrichment type does not support other output types (such as *std* and *list*).

  - **methodcall**
    Uses property values as input parameters in a Java method call to determine an enriched value for the properties. Properties are treated as strings using this option and the Java method must accept a string array as its only parameter. The methodcall type requires the following attributes:
    - jclass
    - method
  - **exe**
    Uses property values as input parameters in an executable to determine an enriched value for the properties. The exe type requires the following attribute:
    - cmdline

- **outputtype**
  Defines the type of output to return. The Enrich operation can return standard output, a list output, or a paired list output. The following are valid values for this attribute:
  - **std**
    Indicates that the returned output is a singular value.
  - **ref**
    Indicates that the given mapout value is a variable that contains the output to return.
  - **list**
    Indicates that the returned output value is a comma-delimited list of values.
    Example: red,blue,green
    The resulting properties are referenced as follows:
    - *xxxxxx_y*
      xxxxxx is the name of the output property (as designated by the output attribute)
      y is the index of the list value (where 0 is the first element in the list)

    For example, if the output attribute is color, red would be referenced as color_1.
  - **pairedlist**
    Indicates that the returned output value is a paired comma-delimited list of values.
    Example: color,red,size,large,name,fido
    The resulting properties are referenced as follows:
    - *xxxxxx_zzzzz*
      xxxxxx is the name of the output property (as designated by the output attribute)
      zzzzz is the name of the returned property in the list (color, size, and name in the example).

    For example, if the output attribute is myenrichsource, size in the example (whose value is large) is referenced as myenrichsource_size.
- **mapin**
  (map only) Defines a regular expression pattern that compares the input property value.
- **mapout**

(map only) Defines the assigned value to the output property if the input property matches the mapin regular expression. Specify an event property for mapping to the event property's value.

- **inputtype**
  (jdbc only) Defines the value types for the input properties. Valid values are any Java primitive types such as int, string, long, and bool.
- **connectionstring**
  (jdbc only) Defines a JDBC connection string to a database instance. This string must include the database instance, name, user name, and password. The subsequent JDBC example shows use of a string.
- **jdbc driver**
  (jdbc only) Defines the JDBC driver Java class. Write the class for this attribute without the .class extension.
- **query**
  (jdbc only) Defines a SQL SELECT query that returns the value to use for the input property.
- **returntype**
  (jdbc only) Defines the value type of the value returned from the JDBC query. Valid types are any Java primitive type such as int, string, and bool. If you are using a complex JDBC query that returns multiple values, specify the return type for each value in a comma-delimited list. The order of the list must correspond to the values defined in the column attribute.
- **column**
  Defines aliases for returned columns when multiple columns of data are returned from a JDBC query. Specify a comma-delimited list of identifiers for each column value so that each piece of data is uniquely referenced in separate properties. This attribute is only required with a multiple column JDBC query.
- **jclass**
  (methodcall only) Defines the Java class full name where you run a method.
- **method**
  (methodcall only) Defines the name of the Java method that returns the value used for the input property.
- **cmdline**
  (exe only) Defines the command line that includes the full pathname and returns the value for the input property. Use substitution markers ({0}, {1}, and {2}) that are replaced with the input property values.
- **output**
  Defines the property that is assigned the output value of the enrich operation. The output property is assigned the following value for each enrichment type:
  - For map enrichment, the output property is assigned the value of the mapout attribute.
  - For jdbc enrichment, the output property is assigned the return value of the jdbc query.
  - For methodcall enrichment, the output property is assigned the return value of the method call.
  - For exe enrichment, the output property is assigned the stdout value of the executable.

  For all enrichment types, the output property can be a new or existing property.

Example: Enrich city and state output with region information

The following example maps the city and state input properties to one region output property according to regular expressions:

```
< Enrich>
   <Field input="city,state" type="map" output="region" outputtype="std">
       <mapentry mapin="^Cin.*,OH$" mapout="Midwest" />
       <mapentry mapin="^New York.*,NY$" mapout="East" />
   </Field>
</Enrich>
```

The <Enrich> property makes the following searches:

- City and state tags that begin with Cin
- State tag of OH
- Begin with New York
- Have a state tag of NY

This property enriches the tags by adding the appropriate region for each of these locations.

Example: Enrich resource output with a complex JDBC query

The following example enriches an event with multiple columns and rows from a database table. This scenario is similar to the previous example with the name and description being queried from the ca_resource_department_table:

```
< EventClass name="OPR">
   < Enrich>
       <Field input="internal_resourceaddr" inputtype="string" type="jdbc"
       outputtype="pairedlist" column="name,desc,org"
       connectionstring="jdbc:sqlserver://server01;databaseName=mdb;
       user=nsmadmin;password=admin;"
       jdbcdriver="com.microsoft.sqlserver.jdbc.SQLServerDriver"
       query="select name,description,organization_uuid from
       ca_resource_department where id=?"
       returntype="string,string,string" output="department" />
   </Enrich>
</EventClass>
```

The <Enrich> property uses the internal_resourceaddr value as in the previous example. The query returns the corresponding name, description, and organization_uuid. The query also assigns this information to new properties using the defined column attributes as follows:

- Returned name is assigned department_name_0
- Returned description is assigned department_desc_0
- Returned organization is assigned department_org_0

When a query returns multiple rows, the trailing number on the property represents the row number. For example, a second returned row would be labeled as department_name_1, department_desc_1, and department_org_1.

Normalize Operation

Contents

Normalize operations transform the syntax of event property values to give values from all sources a uniform nomenclature. For example, you may want to map several similar severity property values (Ok, Success, Good, and so on) to Normal for uniformity purposes.

You can write normalize operations based on the following types of transformation:

- Regular expression
- JDBC query
- Java method call
- Command line executable

Normalize operations support inheritance. The normalize operation process traverses all normalization policy field elements in order from top to bottom until all field elements are processed. Matches are recorded and processed.

Normalize Property Transforms Event Property Values

Normalize operations begin with a <Normalize> property. Normalize operations transform the syntax of event property values to give values from all sources a uniform nomenclature.

This property has the following format:

```
<Normalize>
   <Field input= type= [inputtype= connectionstring= jdbcdriver= query=
   returntype=]|[jclass= method=]|[cmdline=] output= />
       [<mapentry mapin= mapout=>]
</Normalize>
```

> **NOTE**
> The input, type, and output attributes are required for type in the <Normalize> property. Other required attributes depend on the type of normalize operation you are writing.

- **input**
  Defines the list of properties to normalize.
- **type**
  Defines the type of normalization to perform. You can have multiple fields of the same or different types within a single Normalize property with no restrictions. The following are available types:
  - **map**
    Matches and normalizes properties against regular expressions. Map normalization uses mapentry elements that represent an expression and an output for assigning to the property if the expression is matched. These elements are read from top to bottom until a property matches an element, after which additional mapentries are not considered. This type requires use of the following attributes:
    - mapin
    - mapout
  - **jdbc**
    Uses property values as input parameters in a JDBC query to determine the normalized value for the properties. This type requires the following attributes:
    - inputtype
    - connectionstring
    - jdbcdriver
    - query
    - returntype
  - **methodcall**
    Uses property values as input parameters in a Java method call to determine the normalized value for the properties. Properties are treated as strings using this option and the Java method must accept a string array as its only parameter. This type requires the following attributes:
    - jclass
    - method
  - **exe**
    Uses property values as input parameters in an executable to determine the normalized value for the properties. This type requires the following attribute:
    - cmdline
- **mapin**

  (map only) Defines a regular expression pattern that compares the input property value.
- **mapout**

(map only) Defines the assigned value to the output property if the input property matches the mapin regular expression.

- **inputtype**

  (jdbc only) Defines the value types for the input properties. Valid values are any Java primitive types such as int, string, long, and bool.
- **connectionstring**

  (jdbc only) Defines a JDBC connection string to a database instance. This string must include the database instance, name, user name, and password. The subsequent JDBC example shows use of a string.
- **jdbc driver**

  (jdbc only) Defines the JDBC driver Java class. Write the class for this attribute without the .class extension.
- **query**

  (jdbc only) Defines a SQL SELECT query that returns the value to use for the input property.
- **returntype**

  (jdbc only) Defines the value type of the value returned from the JDBC query. Valid types are any Java primitive type such as int, string, and bool.
- **jclass**

  (methodcall only) Defines the Java class full name where you run a method.
- **method**

  (methodcall only) Defines the name of the Java method that returns the value used for the input property.
- **cmdline**

  (exe only) Defines the command line that includes the full pathname and returns the value for the input property. Use substitution markers ({0}, {1}, and {2}) that are replaced with the input property values.
- **output**
  Defines the property that is assigned the output value of the normalization operation. The output property is assigned the following value for each normalization type:
  - For map normalization, the output property is assigned the value of the mapout attribute.
  - For jdbc normalization, the output property is assigned the return value of the jdbc query.
  - For methodcall normalization, the output property is assigned the return value of the method call.
  - For exe normalization, the output property is assigned the stdout value of the executable.

  For all normalization types, you can use a new or existing output property.

Example: Normalizing city output by mapping with regular expressions

The following example maps the city and state input properties to one city output property according to regular expressions:

```
<Normalize>
   <Field input="city,state" type="map" output="city">
       <mapentry mapin="^Cin.*,IA$" mapout="Cincinnati" />
       <mapentry mapin="^Cin.*,OH$" mapout="Cincinnati" />
   </Field>
</Normalize>
```

The <Normalize> property searches for city properties that begin with Cin, have a state property of IA or OH, and normalizes these input properties to a city output that reads Cincinnati.

Example: Normalizing vendor output using a jdbc query

The following example finds the vendorid input property and normalizes the property to display the vendor's name using a jdbc query:

```
<Normalize>
   <Field input="vendorid" inputtype="string" type="jdbc"
   connectionstring="jdbc:sqlserver://server01;databaseName=trapdb;
   user=sa;password=sa;"
   jdbcdriver="com.microsoft.sqlserver.jdbc.SQLServerDriver"
   query="select vendorname from traptable where vendorid=?" returntype="string"
   output="vendorname" />
</Normalize>
```

The <Normalize> property searches for vendorid properties and normalizes these properties by running a JDBC query to find the vendor name for each vendorid in a database that contains this information. The property displays the vendor name in place of the vendorid in a new vendorname output property.

Example: Normalizing zip code output using a Java method

The following example finds the city, state, and zip input properties and normalizes this information into one ninedigitzipcode property by running a Java method:

```
<Normalize>
   <Field input="city,state,zip" type="methodcall"
   jclass="com.ca.eventplus.catalog.methods.ZipCode" method="ConvertZip"
   output="ninedigitzip" />
</Normalize>
```

The <Normalize> property normalizes the city, state, and zip input properties into the value of the zip code expressed in nine digits by running a Java method to obtain this value. The property displays the nine-digit zip code in a new output property in place of the input properties.

Example: Normalizing zip code output using a command line executable

The following example finds the city, state, and zip input properties and normalizes this information into one ninedigitzip property by running a command line executable:

```
<Normalize>
   <Field input="city,state,zip" type="exe" cmdline="c:\\normzip.exe {0} {1} {2}"
  output="ninedigitzip" />
</Normalize>
```

The <Normalize> property normalizes the city, state, and zip input properties into the value of the zip code expressed in nine digits by running a command line executable to obtain this value. The executable contains substitution markers that are replaced with each input property value to calculate the nine-digit zip code using this information. The property displays the nine-digit zip code in a new output property in place of the input properties.

Format Operation

Contents

Format operations combine property values into a new or existing property using a specified format. You can use format operations to define information in events received from event sources using a new property and adhering to a specified format.

Format operations fully support inheritance. The formatting process traverses all format operation fields in order from top to bottom until all field elements are processed. Any matches are recorded and processed.

Format Property Define Information and Event Property Format

Format operations begin with a DO NOT USE property. Format operations can define information in events received from event sources using a new property and adhering to a specified format.

This property has the following format:

```
< Format>
   <Field conditional= input= format= output= />
</Format>
```

- **conditional**
  (Optional) References an event property whose existence or non-existence determines whether the format operation is performed.
- **input**
  Defines an event input property or list of event properties to be output in another property with a new format.
- **format**
  Defines the format for the specified input properties. Use substitution markers to indicate each property according to the order in the input attribute.
- **output**
  Defines a single output property that is assigned the value of the reformatted input properties.

Example: Format severity event input into a meaningful description

The following example searches for events that indicate a severity change for a resource and reformats this information into a short description:

```
< Format>
   <Field input="resource,hostwork,severity" format="The {0} on machine {1} is
   {2}" output="description" />
</Format>
```

The DO NOT USE property searches for the resource, hostwork, and severity properties that indicates a change in severity for a resource on a host system. The property reformats the input values into a short description and outputs the description to the description property. The format attribute uses substitution markers according to the input properties order in the input attribute to put the property values correctly in the description. An example output description property looks similar to the following:

```
The CPU on machine server01 is Critical
```

Example: Format an assigned property with conditional criteria

The following example assigns the property referenced by township to the output property, but only if township exists. If township does not exist, the property referenced by city is assigned.

```
< Format>
   <Field conditional='township'  input='township' format="{0}}" output="municipality" />
   <Field conditional='!township'  input='city' format="{0}}" output="municipality" />
</Format>
```

Evaluate Operation

Contents

Evaluate is the policy operation that enables correlations and actions. It evaluates streams of events against defined rules and runs workflow actions when the rules are met. You can write evaluate operations to automate intelligent actions in response to one or more event conditions that require more acknowledgment or action than a simple resolution.

Some of the examples where you can use the Evaluate operation are as follows:

- You can write a rule that detects if a source (SNMP, application log, and so on) generates three events reporting poor response time in a ten minute interval and creates a new event with a higher severity than the previous events that indicates a consistently poor response time.
- You can write rules to infer a clear event for low-level event sources that do not contain this mechanism. For example, an exception from an application log may not have the ability to clear a previously logged exception. You can write a rule for these exceptions to detect symptoms that the exception has been resolved and send a true clear alert to the event destination.
- You can write rules to associate events being sent to CA SOI as infrastructure alerts with the appropriate CI type. For example, you can write a rule to detect multiple events that indicate a database problem (where one event would not make this clear) and map the events to a Database CI.

You write evaluate operations in policy, but the operations rely on the Drools language, which adheres to a different format and must be inserted in the evaluate operation. For more information about writing Drools event-based rules and workflow actions, see the Drools documentation.

The Event Policies dialog in the Operations Console provides an interface for defining event actions such as correlation, enrichment, filtering, and event creation that are automatically converted to the Drools language. Writing evaluate operations directly in policy files is only necessary if your action is not possible through the Event Policies dialog.

> **NOTE**
> For more information about writing event policies and actions in the Operations Console, see the Working with Event Policies and Actions section.

Evaluate Property Evaluate Events Based on Defined Rules and Run Workflow Actions

Evaluate operations begin with an <Evaluate> property, which has the following basic syntax:

```
<Evaluate>
   <Field input="rule name" output="DRL">
      <!CDATA[
      <Drools rule>
   </Field>
   <Field input="action name" output="DRF">
      <!CDATA[
      <Drools action>
   </Field>
</Evaluate>
```

- **input**
  Defines the name of the event rule in the rule section and the name of the corresponding action in the action section.
- **output**
  Defines the type of Drools language to output. Use DRL for rules and DRF for actions.
- **Drools rule**
  Defines the event rule criteria in the Drools language.
- **Drools action**
  Defines the event workflow action to run if the rule criteria are met. A workflow action is not required for every rule if the rule itself can perform the appropriate action.

Example: Detect immediate service shutdown and create a higher severity event

The following example detects when a Windows service shuts down within 30 seconds after starting. These operations are tracked in separate events, so an event rule is required to correlate the events and trigger an appropriate action. In this case, the operation creates a new event to replace the other events with a message and severity that reflects the

more serious nature of the situation, and also prints the message to a CSV file. This evaluate operation contains a rule and does not require a separate action.

> **WARNING**
> This is a basic example that is easily configurable using the Event Policies dialog in the Operations Console. You should always use the Event Policies dialog to create evaluate operations for event policies, unless the operation is not supported by the interface. For information about creating more complex Drools rules, see the Drools documentation. For other syntax examples (for example, if you want to create a complex enrichment evaluate operation and need a frame of reference), create event policies from the Event Policies dialog and see the resultant syntax at SA_HOME\resources\EventManagement\Policies.

The rule for this example is as follows:

```
<EventClass name='Alert'>
    <Evaluate>
      <Field input='test drools' output='DRL'>
<![CDATA[
package com.ca.eventplus.catalog;
import com.ca.eventplus.catalog.util.EPEvent;
import java.util.HashMap;
declare EPEvent
  @role(event)
end

rule "test drools"
no-loop true
when
pattrn1 : EPEvent((alertedMdrElementID=="?" && message matches ".**entered the running state.*") &&
 reEvaluate!="test drools")
pattrn2 : EPEvent((alertedMdrElementID=="?" && message matches ".**entered the stopped state.*") &&
 reEvaluate!="test drools", this after[-30s,30s] pattrn1)
then
    pattrn1.createEvent("test drools",true,false,pattrn1,pattrn2);
end
]] >
      </Field>
    </Evaluate>
</EventClass>
<EventClass name='test drools' extends='Alert'>
   <FormatPostN>
      <Field  output='AlertType' format='Quality' input='' />
      <Field  output='Severity' format='Major' input='' />
      <Field conditional='pattern1.AlertedMdrProduct'
      output='AlertedMdrProduct' format='{0}'
      input='pattern1.AlertedMdrProduct' />
      <Field conditional='pattern1.AlertedMdrProdInstance'
      output='AlertedMdrProdInstance' format='{0}'
      input='pattern1.AlertedMdrProdInstance' />
      <Field conditional='pattern1.AlertedMdrElementID'
      output='AlertedMdrElementID' format='{0}'
      input='pattern1.AlertedMdrElementID' />
      <Field output='Message' format='{0}' input='Service crashing' />
      <Field output='MdrProduct' format='{0}' input='pattern1.MdrProduct' />
      <Field output='MdrProdInstance' format='{0}'
```

```
        input='pattern1.MdrProdInstance' />
        <Field output='MdrElementID' format='{0}' input='{uniqueidentifier}' />
        <Field output='ReportTimestamp' format='{0}' input='{xsdateTime}' />
        <Field output='OccurrenceTimestamp' format='{0}'
        input='pattern1.OccurrenceTimestamp' />
    </FormatPostN>
</EventClass>
```

NOTE
This syntax comes from an event policy file created through the Event Policies dialog. If you are manually writing evaluate operations, you would write them directly in a connector policy file integrated with the rest of the connector policy.

The input and output properties define the rule name and output. The Drools rule is embedded in the '![CDATA[' property. The Drools rule contains the following sections:

- **import**
  Defines Java methods to import for use in the rule. This declaration must include the EPEvent method, which describes the event properties that the Drools engine can use.
- **declare EPEvent**
  Declares EPEvent as an event role, enabling correlation between events.
- **rule "test drools"**
  Starts the event rule.
- **when**
  Defines the rule criteria. The when clause in this example looks for the following events occurring within thirty seconds of one another:
  - An event with a message that contains the text 'entered the running state'
  - An event with the same alertedMdrElementID value as the 'pattrn1' event and a message that contains the text 'entered the stopped state'

  Note the format of the clause, specifically how it uses the EPEvent method to retrieve and evaluate the properties. Also note the syntax of the clause that defines the time interval between events.
- **then**
  Defines the action to run when the criteria in the when clause are met. The then clause in this example creates a new event based on the properties of the correlated events.
- **<FormatPostN>**
  Sets the properties for the new event. Note that this syntax uses the Format operation to establish the new event properties, and the event class matches the name of the original rule. The Message and Severity properties have new values that reflect the new event condition, and the other properties use the values from the first event.

  NOTE
  Several properties have been omitted from this example.

For additional examples and information about the syntax and requirements of the Drools language version 5, see the following page: http://www.jboss.org/drools/documentation.html.

Filter Operation

Contents

Filter operations exclude certain events from further processing. These operations can exclude or include events with a certain event tag value or combinations of tag values. For example, you may not want to exclude the processing of events with a Normal severity.

Filter operations are evaluated in the order entered in the file. The operations adhere to the following conventions:

- If an event matches exclude criteria, the IFW immediately discards the event without evaluating ensuing filter entries.
- If an event matches include criteria, the IFW keeps the event and does not evaluate ensuring entries.
- If an event does not match an entry criteria, the IFW continues to evaluate the filter operations in order through the end of the filter operations.

Assembling ordered combinations of filter criteria offers the flexibility to create complex filters.

You can create filter operations in any of the following ways:

- Directly in connector policy files using the format described in this topic. This method applies specific filters to events received from a specific source.
- In the Operations Console for events using event policies (either on multiple domain managers using the mid-tier connector or on a single source).

Filter Property Excludes Events from Processing

Filtering operations begin with a <Filter> property. Filter operations exclude certain events from further processing.

This property has the following format:

```
< EventClass name=>
   < Filter>
       <Field type= input= pattern=>
   </Filter>
</EventClass>
```

- **name**

  Defines the name of the event class that you are using to create the connector policy.
- **type**
  Defines if you want to exclude events matching the pattern. Specify exclude to exclude all events matching the pattern, and specify include to include all events matching the pattern.
- **input**
  Defines the event property value that is compared against the pattern attribute for filtering. You can combine multiple properties into one entry using a comma-delimited list. In this scenario, both properties must match their corresponding patterns for the filter criteria to be met.
- **pattern**
  Defines the regular expression pattern that the input property must match to trigger the filtering action. Use comma-delimited patterns to correspond to multiple input values. If the pattern is matched for an exclude filter type, the entire event is excluded from further processing and is directly sent to the core Write module for output. If matched for an include filter type, the core includes the event in processing and dispatching.
  You can use this attribute on the last filter entry to create a default filter for all events not filtered by other entries. For example, you can enter the regular expression ^.*$ to exclude all events not filtered or included by the preceding entries.

Example: Exclude low severity events

The following example excludes events received from the Windows Event Log with a severity of OK or Normal:

```
< EventClass name="SYSLOG">
   < Filter>
       <Field input="internal_newseverity" pattern="^30$" type="exclude" />
   </Filter>
</EventClass>
```

Example: Include high severity events

The following example includes events received from the Windows Event Log with a severity of Critical and excludes all other severities:

```
< EventClass name="SYSLOG">
   < Filter>
       <Field input="internal_newseverity" pattern="^(70|90)$" type="include" />
       <Field input="internal_newseverity" pattern="^.*$" type="exclude" />
   </Filter>
</EventClass>
```

Example: Create a complex filter

The following example filters Windows Event Log events by including high severity events of a certain class and excluding all other events:

```
< EventClass name="SYSLOG">
   < Filter>
       <Field input="internal_newseverity,internal_resourceclass"
       pattern="^(70|90),Application$" type="include" />
       <Field input="internal_newseverity" pattern="^.*$" type="exclude" />
   </Filter>
</EventClass>
```

The <Filter> property filters events received from the Windows Event Log by doing the following in order:

- Includes all events with a Critical severity with a resourceclass of "Application"
- Excludes all events not explicitly included by a previous entry

Write Operation

Contents

Write operations define where an event is sent after processing. You can copy an event into several internal buffers to send an event to multiple destinations, one for each destination.

Write operations support inheritance. The write operation processes only a single Write element that defines where to copy the event.

For outbound from connector policy, do you not have to include a Write element if you leveraged global policy, because global policy includes the necessary write operation.

Write Property Defines Where an Event is Sent

Write operations begin with a <Write> property. The <Write> property defines where an event is sent after processing.

This property has the following format:

```
<Write tagfilter= />
   <Field type="file" name= properties= />
   <Field type="publishcache" properties= />
</Write>
```

- **tagfilter**
  Defines a regular expression that, if matched against an event property, removes the property from the event.
- **type**
  Defines the type of destination. The provided destinations and their syntax are as follows:

  - Publishing Queue: publishcache
  - A file for IFW processing: file
- **properties**
  Defines the event properties to include in the written output.

Example: Write CIs to a file

The following example writes all normalized CIs to a file called "sampleCIs.txt," which is automatically placed in a location for IFW processing:

```
<Field type="file" name="sampleCIs.txt" properties="/>
```

Connector Policy Examples

Contents

For full connector policy examples with explanations for each operation, see Outbound Policy Example and Inbound Policy Examples. You can also use the Sample connector policy files as examples of complete outbound and inbound connector policy. Install the Sample connector to make the connector policy files available at SOI_HOME\resources\Core \Catalogpolicy.

Additionally, the policy writing process has been explained with the help of an example, providing information about how you can build your policy for the CA Catalyst connector for SNMP.

Example Writing Connector Policy for the CA Catalyst Connector for SNMP

This section explains the policy writing process with the help of an example. The example describes how you can write your connector policy for the CA Catalyst connector for SNMP. The SNMP connector is a generic connector (Level 2 integration) that receives SNMP traps from any SNMP-capable product, transforms the traps, and maps the generic SNMP attributes to the USM model. Each domain manager (integrating product) requires you to write a separate connector policy.

The SNMP connector ships a sample policy file that you can customize.

Integrating with CA Workload Automation through the SNMP Connector

For this example, the domain manager that sends SNMP traps to the SNMP connector is CA Workload Automation. For CA Workload Automation, any job scheduled on the specific agent host is a CI and is mapped to the USM type ITActivity.

> **NOTE**
> This example assumes that you have configured CA Workload Automation to forward traps to the SNMP connector system on the port that you specified during connector installation.

How to Write the Example Connector Policy

The policy writing process for this example includes the following steps:

1. Understand the SNMP trap received from the domain manager.
2. Understand the addition of the eventtype properties (Item and Alert) to the received SNMP trap.
3. Add the event classes (Item, Alert, and USM-Entity) to the policy file.
4. Write appropriate policy operations for the specified event classes:
   - Policy operations for the Item event class
   - Policy operations for the Alert event class

A special property named eventtype determines each event type in the sample policy. Connector policy matches the name attribute of each <EventClass> property in the policy file to the eventtype property in the event and performs the policy

operations for that class on the event. An event class represents a container for all processing operations related to a specific type of event.

Understand the SNMP Trap Received from the Domain Manager

The structure of the SNMP trap that the SNMP connector receives from its domain manager is as follows. Note that varbinds and varbindvals are represented as comma-separated values:

```
snmp_agent="155.35.37.142"
snmp_community="public"
snmp_varbindvals="Manager,ESP_COE-NE-DSWA-D1_7500,MAIN,OrderEntry DB Backup
 Workflow,23,DATA_TRANSFER_JOB,,READY,Automated Test Service,This is my description"
snmp_ticks="0"
snmp_genericTrap="6"
snmp_varbindoid="1.3.6.1.4.1.11203.1,1.3.6.1.4.1.11203.3,1.3.6.1.4.1.11203.11,1.3.6.1.4.1.11203.7,1.3.6.1.4.1.11203.8,1
snmp_requestID="0"
snmp_errorStatus="Success"
snmp_enterprise="1.3.6.1.4.1.11203"
snmp_errorIndex="0"
snmp_specificTrap="1"
```

After receiving SNMP traps from the domain manager, the SNMP connector converts them as data objects and produces the output in name/value pair format. The connector policy must use these name/value pairs as its input.

Understand the Addition of the Eventtype Property to the Trap

The SNMP connector adds the eventtype property to the received trap before sending it to policy for transformation. The SNMP connector receives only traps with no CIs available for these traps. Therefore, when the connector receives a trap for the first time, it explicitly creates a CI based on the trap. The connector first sends the trap as a CI with eventtype as Item. It then sends the same trap as an alert but with eventtype as Alert. Subsequently, if the same trap is received again, the connector checks its internal cache and verifies that the trap is already available as a CI, it then sends the trap only as an alert.

For example, if CA Workload Automation sends a trap indicating that a job failed, the connector first creates a CI based on the trap for the job, and then creates an alert associated with that CI representing the job failure. The connector policy uses the trap sent by the connector with an eventtype of Item to create the CI and the trap with the eventtype of Alert to create the alert. If another trap is sent related to the same job, the connector detects the relation and only sends a trap with the Alert eventtype.

You must create a policy for both the Item and Alert eventtypes.

Item Eventtype

For the Item eventtype, the connector adds eventtype=Item to the trap information as follows:

```
snmp_agent="155.35.37.142"
snmp_community="public"
snmp_varbindvals="Manager,ESP_COE-NE-DSWA-D1_7500,MAIN,OrderEntry DB Backup
 Workflow,23,DATA_TRANSFER_JOB,,READY,Automated Test Service,This is my description"
snmp_ticks="0"
snmp_genericTrap="6"
snmp_varbindoid="1.3.6.1.4.1.11203.1,1.3.6.1.4.1.11203.3,1.3.6.1.4.1.11203.11,1.3.6.1.4.1.11203.7,1.3.6.1.4.1.11203.8,1
snmp_requestID="0"
snmp_errorStatus="Success"
snmp_enterprise="1.3.6.1.4.1.11203"
snmp_errorIndex="0"
snmp_specificTrap="1"
```

```
eventtype = Item
```

Alert Eventtype

For the Alert eventtype, the connector adds eventtype=Alert to the trap information as follows:

```
snmp_agent="155.35.37.142"
snmp_community="public"
snmp_varbindvals="Manager,ESP_COE-NE-DSWA-D1_7500,MAIN,OrderEntry DB Backup
 Workflow,23,DATA_TRANSFER_JOB,,READY,Automated Test Service,This is my description"
snmp_ticks="0"
snmp_genericTrap="6"
snmp_varbindoid="1.3.6.1.4.1.11203.1,1.3.6.1.4.1.11203.3,1.3.6.1.4.1.11203.11,1.3.6.1.4.1.11203.7,1.3.6.1.4.1.11203.8,1
snmp_requestID="0"
snmp_errorStatus="Success"
snmp_enterprise="1.3.6.1.4.1.11203"
snmp_errorIndex="0"
snmp_specificTrap="1"
eventtype = Alert
```

Add the Event Classes

Connector policy matches the name attribute of each <EventClass> property in the policy file to the eventtype property in the event and performs the policy operations for that class on the event.

For this example, you add the following event classes to the policy file:

- Item
- Alert
- USM-Entity

Add the Item Event Class

For the Item eventtype, add an event class with its name as Item in the policy file so the policy can create CIs based on received CA Workload Automation traps.

<Catalog version="1.0" globalextends="GLOBAL!">

<!-- ======Event Class====== -->

<EventClass name="Item">

</EventClass>

</Catalog>

All applicable policy operations for the Item eventtype are explained in the Policy Operations for the Item Event Class section.

Add the Alert Event Class

For the Alert eventtype, you must add an event class with its name as Alert to the same policy file beneath the already defined event classes. You must define policy under the Alert eventtype to process the received traps that reach the policy defined as alerts, so that they can appear on the Operations Console associated with the CI of the trap processed using the Item eventtype policy.

<Catalog version="1.0" globalextends="GLOBAL!">

<!-- ======Event Class====== -->

<EventClass name="Alert">

</EventClass>

</Catalog>

Alert is a separate event class that does not extend the previously defined Item event class. All applicable policy operations for the Alert eventtype are explained in the Policy Operations for the Alert Event Class section.

Add the USM-Entity Event Class

For the USM-Entity eventtype, you must add an event class with its name as USM-Entity at the bottom of the policy file. The USM-Entity eventtype publishes the transformed CIs and Alerts.

<Catalog version="1.0" globalextends="GLOBAL!">

<!-- ======Event Class====== -->

<EventClass name="USM-Entity" />

</EventClass>

</Catalog>

Write Policy Operations

For this example, you write appropriate policy operations for the Item and Alert event classes. You do not need to write any policy operation for the USM-Entity event class.

Policy Operations for the Item Event Class

The following policy operations are added to the policy file for the Item eventtype:

- Classify
- Parse
- Normalize
- Format

Classify

You must classify traps that reach the policy as an Item eventtype to specific CI types. To do this, you must choose a property from the connector data output that you can use for classifying the CI. For this example, snmp_varbindoids has been chosen as the classifier. Write the Classify section of the sample policy as follows:

```
<Catalog version="1.0" globalextends="GLOBAL!">
   <!-- ======Event Class====== -->
   <EventClass name="Item">
   <!-- Classify -->
   <Classify>
    <Field input="snmp_varbindoids" pattern=".*1\.3\.6\.1\.4\.1\.11203\.9.*$" output="eventtype"
 outval="ITActivity" />
   </Classify>
   </EventClass>
</Catalog>
```

This policy classifies traps defined under the Item eventtype as belonging to the ITActivity CI type when the value of the snmp_varbindoids property contains the pattern 1.3.6.1.4.1.11203.9.

For example, if the snmp_varbindoids value is 1.3.6.1.4.1.11203.1,1.3.6.1.4.1.11203.3,1.3.6.1.4.1.11203.11,1.3.6.1.4.1.11203.7,1.3.6.1.4.1.11203.8,**1.3.6.1.4.1.11203.9**,1.3.6.1.4. the trap is classified to ITActivity, because it contains the specified 1.3.6.1.4.1.11203.9 pattern.

By default, all received traps in this example are considered to be of type ITActivity. Most detailed integrations require you to define multiple classification rules for multiple CI types (for example, ComputerSystem, Process, Database, and so on).

Each CI type requires its own specialized processing operations. Because this example classifies all traps under one CI type, you define the ITActivity class as follows:

```
<Catalog version="1.0" globalextends="GLOBAL!">
<!-- ======Event Class====== -->
<EventClass name="Item">
<!-- Classify -->
<Classify>
    <Field input="snmp_varbindoids" pattern=".*1\.3\.6\.1\.4\.1\.11203\.9.*$" output="eventtype"
 outval="ITActivity" />
</Classify>
</EventClass>
<!--  ======Event Class======   -->
<EventClass name="ITActivity" extends="Item">
</EventClass>
</Catalog>
```

All processing operations specific to the ITActivity CI type occur within this EventClass definition.

Parse

In the received CA Workload Automation traps, a single property, snmp_varbindvals, represents various individual properties (such as nodetype, nodename, state, and so on) as comma-separated values. To isolate all of these individual property values from the snmp_varbindvals property so that you can map them to individual USM properties, you must use the Parse operation. Parse operations split event properties into additional properties using regular expression subgroups.

For this example, the trap varbind data is parsed into meaningful properties in the policy file that can later be mapped to USM properties. Write the Parse section for the Item event class as follows:

> **NOTE**
> The ITActivity class inherits the same parse information from the parent Item class.

<Catalog version="1.0" globalextends="GLOBAL!">

<!-- ======Event Class====== -->

<EventClass name="Item">

<!-- Classify -->

<Classify>

<Field input="snmp_varbindoids" pattern=".*1\.3\.6\.1\.4\.1\.11203\.9.*$" output="eventtype" outval="ITActivity" />

</Classify>

<Parse>

<Field input="snmp_varbindvals" output="temp_nodetype,temp_nodename,temp_domain,temp_applname,temp_applgen,temp_jobname,temp_jobequal,ter pattern="^(.*?),(.*?),(.*?),(.*?),(.*?),(.*?),(.*?),(.*?),(.*?)$" />

</Parse>

</EventClass>

<!-- ======Event Class====== -->

<EventClass name="ITActivity" extends="Item">

</EventClass>

</Catalog>

This policy takes the received comma-separated snmp_varbindvals property values and assigns them to individual property names according to their order of occurrence. For example, the first value is assigned to the temp_nodetype property, and second value is assigned to the temp_nodename property, and so on. After the value field is parsed and available, you can use the same information for uniquely identifying a specific CI or alert.

Normalize

In the parsing section, the job state property temp_state that you created from a specific trap varbind value represents the activity state of the trap. The activity state for CA Workload Automation traps can be one of the following:

- Unknown
- Complete
- Monitor
- Exec
- Failed
- Premature End
- Inactive
- Overdue
- Suberror
- Agent Down
- Ready
- Abandon Submission

You must map these activity states to the CA SOI activity states using the Normalize operation. Normalize operations transform the syntax of event property values to give values from all sources a uniform nomenclature. For performing these mappings, write the Normalize section as follows:

```
<Catalog version="1.0" globalextends="GLOBAL!">
   <!-- ======Event Class====== -->
   <EventClass name="Item">
   <!-- Classify -->
   <Classify>
    <Field input="snmp_varbindoids" pattern=".*1\.3\.6\.1\.4\.1\.11203\.9.*$" output="eventtype"
 outval="ITActivity" />
   </Classify>
   <Parse>
    <Field input="snmp_varbindvals"
 output="temp_nodetype,temp_nodename,temp_domain,temp_applname,temp_applgen,temp_
 jobname,temp_jobequal,temp_state,temp_status" pattern="^(.*?),(.*?),(.*?),(.*?),(.*?),(.*?),(.*?),(.*?),
(.*?)$" />
   </Parse>
   </EventClass>
   <!--  ======Event Class======   -->
   <EventClass name="ITActivity" extends="Item">
   <Normalize>
    <Field input="temp_state" type="map" output="activityState">
    <mapentry mapin="[Uu][Nn][Kk][Nn][Oo][Ww][Nn]" mapout="Unknown" />
    <mapentry mapin="[Cc][Oo][Mm][Pp][Ll][Ee][Tt][Ee]" mapout="Finished" />
    <mapentry mapin="[Mm][Oo][Nn][Ii][Tt][Oo][Rr]" mapout="Normal" />
    <mapentry mapin="[Ee][Xx][Ee][Cc]" mapout="Normal-Running" />
    <!--  Informational  -->
    <mapentry mapin="[Ff][Aa][Ii][Ll][Ee][Dd]" mapout="Trouble" />
```

```
    <mapentry mapin="[Pp][Rr][Ee][Mm][Aa][Tt][Uu][Rr][Ee] [Ee][Nn][Dd]" mapout="Obstructed" />
    <mapentry mapin="[Ii][Nn][Aa][Cc][Tt][Ii][Vv][Ee]" mapout="Normal-Waiting" />
    <mapentry mapin="[Oo][Vv][Ee][Rr][Dd][Uu][Ee]" mapout="Obstructed" />
    <mapentry mapin="[Ss][Uu][Bb][Ee][Rr][Rr][Oo][Rr]" mapout="Finished-Completed" />
    <mapentry mapin="[Aa][Gg][Ee][Nn][Tt] [Dd][Oo][Ww][Nn]" mapout="Trouble" />
    <!--  Fatal   -->
    <mapentry mapin="[Rr][Ee][Aa][Dd][Yy]" mapout="Normal" />
    <mapentry mapin="[Aa][Bb][Aa][Nn][Dd][Oo][Nn] [Ss][Uu][Bb][Mm][Ii][Ss][Ss][Ii][Oo][Nn]" mapout="Finished-
Abandoned" />
    </Field>
   </Normalize>
   </EventClass>
</Catalog>
```

This policy maps all potential values of the temp_state property to activity states that CA SOI can understand. For example, Complete maps to Finished, Failed maps to Trouble, Inactive maps to Normal-Waiting, and so on. Now, when you map this property value to the activityState USM property, it can understand and display the supported activity state values.

Format

Format operations combine property values into a new or existing property using a specified format. You can use format operations to define information in events received from event sources using a new property and adhering to a specified format.

For this example, you transform the source properties and the temporary properties that you create using the Parse operation into USM properties as follows:

```
<Catalog version="1.0" globalextends="GLOBAL!">
   <!-- ======Event Class====== -->
   <EventClass name="Item">
   <!-- Classify -->
   <Classify>
    <Field input="snmp_varbindoids" pattern=".*1\.3\.6\.1\.4\.1\.11203\.9.*$" output="eventtype"
 outval="ITActivity" />
   </Classify>
   <Parse>
    <Field input="snmp_varbindvals"
 output="temp_nodetype,temp_nodename,temp_domain,temp_applname,temp_applgen,temp_
 jobname,temp_jobequal,temp_state,temp_status" pattern="^(.*?),(.*?),(.*?),(.*?),(.*?),(.*?),(.*?),(.*?),
(.*?)$" />
   </Parse>
   < Format>
    <!--  Non-Correlatable properties   -->
    <Field output="MdrElementID" format="{0}" input="AlertedMdrElementID" />
   </Format>
```

This policy assigns the mandatory MdrElementID property to the CI based on the value of the AlertedMdrElementID in the alert created from the same trap. You assign the AlertedMdrElementID value in the format section for alerts. This value determines whether a CI is created for the trap. If the MdrElementID exists, the CI creation does not occur, and only an alert is created for the trap. For more information about assigning values for the AlertedMdrElementID property, see Format for the alert type.

```
   </EventClass>
   <!--  ======Event Class======   -->
   <EventClass name="ITActivity" extends="Item">
```

```
    <Normalize>
     <Field input="temp_state" type="map" output="activityState">
     <mapentry mapin="[Uu][Nn][Kk][Nn][Oo][Ww][Nn]" mapout="Unknown" />
     <mapentry mapin="[Cc][Oo][Mm][Pp][Ll][Ee][Tt][Ee]" mapout="Finished" />
     <mapentry mapin="[Mm][Oo][Nn][Ii][Tt][Oo][Rr]" mapout="Normal" />
     <mapentry mapin="[Ee][Xx][Ee][Cc]" mapout="Normal-Running" />
     <!--  Informational  -->
     <mapentry mapin="[Ff][Aa][Ii][Ll][Ee][Dd]" mapout="Trouble" />
     <mapentry mapin="[Pp][Rr][Ee][Mm][Aa][Tt][Uu][Rr][Ee] [Ee][Nn][Dd]" mapout="Obstructed" />
     <mapentry mapin="[Ii][Nn][Aa][Cc][Tt][Ii][Vv][Ee]" mapout="Normal-Waiting" />
     <mapentry mapin="[Oo][Vv][Ee][Rr][Dd][Uu][Ee]" mapout="Obstructed" />
     <mapentry mapin="[Ss][Uu][Bb][Ee][Rr][Rr][Oo][Rr]" mapout="Finished-Completed" />
     <mapentry mapin="[Aa][Gg][Ee][Nn][Tt] [Dd][Oo][Ww][Nn]" mapout="Trouble" />
     <!--  Fatal   -->
     <mapentry mapin="[Rr][Ee][Aa][Dd][Yy]" mapout="Normal" />
     <mapentry mapin="[Aa][Bb][Aa][Nn][Dd][Oo][Nn] [Ss][Uu][Bb][Mm][Ii][Ss][Ss][Ii][Oo][Nn]" mapout="Finished-
Abandoned" />
     </Field>
    </Normalize>
    < Format>
     <!-- Correlatable properties, must populate at least one -->
     <Field output="ActivityID" format="{0}" input="temp_jobname" />
     <Field conditional="snmp_agent" output="DeviceIPV4Address" format="{0}" input="snmp_agent" />
     <!-- verify that at least one property is set -->
     <Field conditional="DeviceIPV4Address" output="temp_atleastoneset" format="{0}"
 input="DeviceIPV4Address" />
     <Field conditional="!temp_atleastoneset" output="Error" format="At least one correlatable property is not
 set" input="" />
     <!-- Non-Correlatable properties  -->
     <Field output="RuntimeName" format="{0}.{1}.{2}" input="temp_applname,temp_applgen,temp_jobname" />
     <Field output="RuntimeDiscriminator" format="{0}" input="temp_applgen" />
     <Field output="DefinitionName" format="{0}.{1}" input="temp_applname,temp_jobname" />
     <Field output="ActivityTypes" format="Job" input="" />
     <Field conditional="activityState" output="ActivityState" format="{0}" input="activityState" />
     <Field conditional="temp_state" output="StateDescription" format="{0} present status is {1}"
 input="temp_jobname,temp_state" />
    </Format>
    </EventClass>
</Catalog>
```

This policy does the following:

- Ensures that at least one of the mandatory correlatable properties (as defined in the USM schema for the ITActivity type) is set and includes values. It maps the temp_jobname property that you created during the parse operation to the ActivityID USM property. You must ensure that at least one correlatable property is populated for every type in your policy.
- Inserts a control that creates an error if one of the correlatable properties is not populated.
- Uses the temporary properties you created in the parse section to populate key USM properties for the ITActivity type. For example, the first entry in the Non-Correlatable Properties section combines the values of three temporary properties to populate the RuntimeName USM property with each value separated by a period.
- Note the use of the conditional attribute, which only enacts the associated operation if the input property has a value.

Policy Operations for the Alert Event Class

The following operations are added to the policy file for the Alert eventtype:

- Parse
- Normalize
- Format

You add these operations to the same policy file where you have already included the Item eventtype information directly under the Alert event class definition.

Parse

You must parse the trap data for alerts similar to the way you did for trap-based CIs. Separating the varbind values lets you isolate the values for individual use in other sections of policy. For the Alert eventtype, add the parse operation to the Alert event class section of the policy file as follows:

```
<Catalog version="1.0" globalextends="GLOBAL!">
   <!--  ======Event Class======   -->
   <EventClass name="Alert">
   <Parse>
    <Field input="snmp_varbindvals"
 output="temp_nodetype,temp_nodename,temp_domain,temp_applname,temp_applgen,temp_jobname,temp_jobequal,temp_state,temp_
 pattern="^(.*?),(.*?),(.*?),(.*?),(.*?),(.*?),(.*?),(.*?),(.*?)$" />
   </Parse>
   </EventClass>
</Catalog>
```

This policy takes the received comma-separated snmp_varbindvals property values and assigns them to individual property names according to their order of occurrence. For example, the first value is assigned to the temp_nodetype property, and second value is assigned to the temp_nodename property, and so on. After the value field is parsed and available, you can use the same information for uniquely identifying a specific alert.

Normalize

In the parsing section, the job state property temp_state that you created from a specific trap varbind value represents the severity level of the trap. The severity for CA Workload Automation traps may be one of the following:

- Unknown
- Complete
- Monitor
- Exec
- Failed
- Premature End
- Inactive
- Overdue
- Suberror
- Agent Down
- Ready
- Abandon Submission

You must map these severities to the CA SOI severity property using the Normalize operation. Normalize operations transform the syntax of event property values to give values from all sources a uniform nomenclature. For performing these mappings, write the Normalize section as follows:

```
<Catalog version="1.0" globalextends="GLOBAL!">
   <!--  ======Event Class======   -->
   <EventClass name="Alert">
```

```
    <Parse>
     <Field input="snmp_varbindvals"
  output="temp_nodetype,temp_nodename,temp_domain,temp_applname,temp_applgen,temp_jobname,temp_jobequal,temp_state,temp_
  pattern="^(.*?),(.*?),(.*?),(.*?),(.*?),(.*?),(.*?),(.*?),(.*?)$" />
    </Parse>
    <Normalize>
     <Field input="temp_state" type="map" output="severity">
     <mapentry mapin="[Uu][Nn][Kk][Nn][Oo][Ww][Nn]" mapout="Unknown" />
     <mapentry mapin="[Cc][Oo][Mm][Pp][Ll][Ee][Tt][Ee]" mapout="Normal" />
     <mapentry mapin="[Mm][Oo][Nn][Ii][Tt][Oo][Rr]" mapout="Normal" />
     <mapentry mapin="[Ee][Xx][Ee][Cc]" mapout="Normal" />
     <!--  Informational   -->
     <mapentry mapin="[Ff][Aa][Ii][Ll][Ee][Dd]" mapout="Critical" />
     <mapentry mapin="[Pp][Rr][Ee][Mm][Aa][Tt][Uu][Rr][Ee] [Ee][Nn][Dd]" mapout="Critical" />
     <mapentry mapin="[Ii][Nn][Aa][Cc][Tt][Ii][Vv][Ee]" mapout="Minor" />
     <mapentry mapin="[Oo][Vv][Ee][Rr][Dd][Uu][Ee]" mapout="Major" />
     <mapentry mapin="[Ss][Uu][Bb][Ee][Rr][Rr][Oo][Rr]" mapout="Major" />
     <mapentry mapin="[Aa][Gg][Ee][Nn][Tt] [Dd][Oo][Ww][Nn]" mapout="Critical" />
     <!--  Fatal   -->
     <mapentry mapin="[Rr][Ee][Aa][Dd][Yy]" mapout="Normal" />
     <mapentry mapin="[Aa][Bb][Aa][Nn][Dd][Oo][Nn] [Ss][Uu][Bb][Mm][Ii][Ss][Ss][Ii][Oo][Nn]" mapout="Unknown" /
>
     </Field>
    </Normalize>
    </EventClass>
</Catalog>
```

This policy maps all potential values of the temp_state property to severities that CA SOI can understand. For example, Complete maps to Normal, Failed maps to Critical, Inactive maps to Major, and so on. Now, when you map this property value to the Severity USM property, it can understand and display the supported severity values.

Format

Format operations combine property values into a new or existing property using a specified format. You can use format operations to define information in events received from event sources using a new property and adhering to a specified format.

For this example, you transform the source properties and the temporary properties that you create using the Parse operation into USM properties as follows:

> **NOTE**
> Alerts do not have correlatable properties.

```
<Catalog version="1.0" globalextends="GLOBAL!">
    <!--  ======Event Class======   -->
    <EventClass name="Alert">
    <Parse>
     <Field input="snmp_varbindvals"
  output="temp_nodetype,temp_nodename,temp_domain,temp_applname,temp_applgen,temp_jobname,temp_jobequal,temp_state,temp_
  pattern="^(.*?),(.*?),(.*?),(.*?),(.*?),(.*?),(.*?),(.*?),(.*?)$" />
    </Parse>
    <Normalize>
     <Field input="temp_state" type="map" output="severity">
     <mapentry mapin="[Uu][Nn][Kk][Nn][Oo][Ww][Nn]" mapout="Unknown" />
     <mapentry mapin="[Cc][Oo][Mm][Pp][Ll][Ee][Tt][Ee]" mapout="Normal" />
     <mapentry mapin="[Mm][Oo][Nn][Ii][Tt][Oo][Rr]" mapout="Normal" />
```

```
    <mapentry mapin="[Ee][Xx][Ee][Cc]" mapout="Normal" />
    <!--  Informational   -->
    <mapentry mapin="[Ff][Aa][Ii][Ll][Ee][Dd]" mapout="Critical" />
    <mapentry mapin="[Pp][Rr][Ee][Mm][Aa][Tt][Uu][Rr][Ee] [Ee][Nn][Dd]" mapout="Critical" />
    <mapentry mapin="[Ii][Nn][Aa][Cc][Tt][Ii][Vv][Ee]" mapout="Minor" />
    <mapentry mapin="[Oo][Vv][Ee][Rr][Dd][Uu][Ee]" mapout="Major" />
    <mapentry mapin="[Ss][Uu][Bb][Ee][Rr][Rr][Oo][Rr]" mapout="Major" />
    <mapentry mapin="[Aa][Gg][Ee][Nn][Tt] [Dd][Oo][Ww][Nn]" mapout="Critical" />
    <!--  Fatal   -->
    <mapentry mapin="[Rr][Ee][Aa][Dd][Yy]" mapout="Normal" />
    <mapentry mapin="[Aa][Bb][Aa][Nn][Dd][Oo][Nn] [Ss][Uu][Bb][Mm][Ii][Ss][Ss][Ii][Oo][Nn]" mapout="Unknown" /
>
    </Field>
   </Normalize>
   < Format>
    <!-- Correlatable properties, must populate at least one  -->
    <!--  Non-Correlatable properties  -->
    <Field output="MdrElementID" format="alert-{0}:{1}:{2}" input="snmp_agent,temp_applname,temp_jobname" />
    <Field output="OccurrenceTimestamp" format="{0}" input="{xsdateTime(now)}" />
    <Field output="ReportTimestamp" format="{0}" input="{xsdateTime(now)}" />
    <Field output="AlertType" format="{0}" input="Risk-Fault" />
    <Field conditional="severity" output="Severity" format="{0}" input="severity" />
    <Field output="AlertedMdrProduct" format="CA:00036" input="" />
    <Field output="AlertedMdrProdInstance" format="{0}" input="{fqdn(snmp_agent)}" />
    <!--  Assign instance name   -->
    <Field conditional="snmp_agent" output="Section1" format="{0}" input="snmp_agent" />
    <Field conditional="!snmp_agent" output="Flag" format="false" input="" />
    <Field conditional="temp_applname" output="Section2" format="{0}" input="temp_applname" />
    <Field conditional="!temp_applname" output="Flag" format="false" input="" />
    <Field conditional="temp_jobname" output="Section3" format="{0}" input="temp_jobname" />
    <Field conditional="!temp_jobname" output="Flag" format="false" input="" />
    <Field conditional="Flag" output="AlertedMdrElementID" format="" input="" />
    <Field conditional="!Flag" output="AlertedMdrElementID" format="{0}:{1}:{2}"
 input="Section1,Section2,Section3" />
    <Field output="Summary" format="{0}" input="temp_status" />
    <Field conditional="temp_state" output="Message" format="{0} alert on {1} scheduled on host {2}"
 input="temp_state,temp_jobname,snmp_agent" />
    <Field output="MetricName" format="{0}" input="Job Status" />
    <Field output="MetricType" format="{0}" input="Unknown" />
    <Field output="MetricUnitDefinition" format="{0}" input="Number" />
    <Field output="MetricDataType" format="{0}" input="String" />
   </Format>
   </EventClass>
</Catalog>
```

This policy does the following:

- Populates all mandatory alert USM properties as defined in the USM schema with values from source trap properties or temporary properties you created in the parse section.
- Note the use of functions to populate several mandatory properties. For example, a function populates the OccurrenceTimestamp property with the current time, and the AlertedMdrProdInstance value is derived by converting the value of the snmp_agent property to a fully qualified domain name.
- Note how some properties are assigned hard-coded values instead of the value of source properties. For example, the AlertType property is assigned a static value of Risk-Fault.
- Populates the AlertedMdrElementID value, or instance name, using agent IP address, application name, and job name. The policy uses conditional attributes to ensure that each source property has a value before using it to determine the AlertedMdrElementID. AlertedMdrElementID serves as the key value that uniquely identifies an alert and associated CI by its associated agent, application, and job. This value becomes the MdrElementID value for the CI created by the trap, and if the same MdrElementID value exists, the policy does not create a CI, and instead creates only the alert and assigns it to the existing CI with the same MdrElementID value.

  NOTE
  The key values that you use to define the AlertedMdrElementID (and by extension, MdrElementID) value differ depending on the trap source you are integrating. The property must contain a value or combination of values that can uniquely identify a CI and alert from your trap source.
- Populates the alert Summary property with the value of the temporary temp_status property created in the parse section.

Completed Example Policy File

This section includes all the policy sections that you created for this example:

```
<Catalog version="1.0" globalextends="GLOBAL!">
   <!-- ======Event Class====== -->
   <EventClass name="Item">
   <!-- Classify -->
   <Classify>
    <Field input="snmp_varbindoids" pattern=".*1\.3\.6\.1\.4\.1\.11203\.9.*$" output="eventtype"
 outval="ITActivity" />
   </Classify>
   <Parse>
    <Field input="snmp_varbindvals"
 output="temp_nodetype,temp_nodename,temp_domain,temp_applname,temp_applgen,temp_
 jobname,temp_jobequal,temp_state,temp_status" pattern="^(.*?),(.*?),(.*?),(.*?),(.*?),(.*?),(.*?),(.*?),
(.*?)$" />
   </Parse>
   < Format>
    <!--  Non-Correlatable properties   -->
    <Field output="MdrElementID" format="{0}" input="AlertedMdrElementID" />
   </Format>
   </EventClass>
   <!--  ======Event Class======   -->
   <EventClass name="ITActivity" extends="Item">
   <Normalize>
    <Field input="temp_state" type="map" output="activityState">
    <mapentry mapin="[Uu][Nn][Kk][Nn][Oo][Ww][Nn]" mapout="Unknown" />
    <mapentry mapin="[Cc][Oo][Mm][Pp][Ll][Ee][Tt][Ee]" mapout="Finished" />
    <mapentry mapin="[Mm][Oo][Nn][Ii][Tt][Oo][Rr]" mapout="Normal" />
    <mapentry mapin="[Ee][Xx][Ee][Cc]" mapout="Normal-Running" />
    <!--  Informational  -->
```

```
    <mapentry mapin="[Ff][Aa][Ii][Ll][Ee][Dd]" mapout="Trouble" />
    <mapentry mapin="[Pp][Rr][Ee][Mm][Aa][Tt][Uu][Rr][Ee] [Ee][Nn][Dd]" mapout="Obstructed" />
    <mapentry mapin="[Ii][Nn][Aa][Cc][Tt][Ii][Vv][Ee]" mapout="Normal-Waiting" />
    <mapentry mapin="[Oo][Vv][Ee][Rr][Dd][Uu][Ee]" mapout="Obstructed" />
    <mapentry mapin="[Ss][Uu][Bb][Ee][Rr][Rr][Oo][Rr]" mapout="Finished-Completed" />
    <mapentry mapin="[Aa][Gg][Ee][Nn][Tt] [Dd][Oo][Ww][Nn]" mapout="Trouble" />
    <!--  Fatal   -->
    <mapentry mapin="[Rr][Ee][Aa][Dd][Yy]" mapout="Normal" />
    <mapentry mapin="[Aa][Bb][Aa][Nn][Dd][Oo][Nn] [Ss][Uu][Bb][Mm][Ii][Ss][Ss][Ii][Oo][Nn]" mapout="Finished-
Abandoned" />
    </Field>
   </Normalize>
   < Format>
    <!-- Correlatable properties, must populate at least one -->
    <Field output="ActivityID" format="{0}" input="temp_jobname" />
    <Field conditional="snmp_agent" output="DeviceIPV4Address" format="{0}" input="snmp_agent" />
    <!-- verify that at least one property is set -->
    <Field conditional="DeviceIPV4Address" output="temp_atleastoneset" format="{0}"
 input="DeviceIPV4Address" />
    <Field conditional="!temp_atleastoneset" output="Error" format="At least one correlatable property is not
 set" input="" />
    <!-- Non-Correlatable properties  -->
    <Field output="RuntimeName" format="{0}.{1}.{2}" input="temp_applname,temp_applgen,temp_jobname" />
    <Field output="RuntimeDiscriminator" format="{0}" input="temp_applgen" />
    <Field output="DefinitionName" format="{0}.{1}" input="temp_applname,temp_jobname" />
    <Field output="ActivityTypes" format="Job" input="" />
    <Field conditional="activityState" output="ActivityState" format="{0}" input="activityState" />
    <Field conditional="temp_state" output="StateDescription" format="{0} present status is {1}"
 input="temp_jobname,temp_state" />
   </Format>
   </EventClass>
   <!--  ======Event Class======   -->
   <EventClass name="Alert">
   <Parse>
    <Field input="snmp_varbindvals"
 output="temp_nodetype,temp_nodename,temp_domain,temp_applname,temp_applgen,temp_jobname,temp_jobequal,temp_state,temp_
 pattern="^(.*?),(.*?),(.*?),(.*?),(.*?),(.*?),(.*?),(.*?),(.*?)$" />
   </Parse>
   <Normalize>
    <Field input="temp_state" type="map" output="severity">
    <mapentry mapin="[Uu][Nn][Kk][Nn][Oo][Ww][Nn]" mapout="Unknown" />
    <mapentry mapin="[Cc][Oo][Mm][Pp][Ll][Ee][Tt][Ee]" mapout="Normal" />
    <mapentry mapin="[Mm][Oo][Nn][Ii][Tt][Oo][Rr]" mapout="Normal" />
    <mapentry mapin="[Ee][Xx][Ee][Cc]" mapout="Normal" />
    <!--  Informational   -->
    <mapentry mapin="[Ff][Aa][Ii][Ll][Ee][Dd]" mapout="Critical" />
    <mapentry mapin="[Pp][Rr][Ee][Mm][Aa][Tt][Uu][Rr][Ee] [Ee][Nn][Dd]" mapout="Critical" />
    <mapentry mapin="[Ii][Nn][Aa][Cc][Tt][Ii][Vv][Ee]" mapout="Minor" />
    <mapentry mapin="[Oo][Vv][Ee][Rr][Dd][Uu][Ee]" mapout="Major" />
    <mapentry mapin="[Ss][Uu][Bb][Ee][Rr][Rr][Oo][Rr]" mapout="Major" />
    <mapentry mapin="[Aa][Gg][Ee][Nn][Tt] [Dd][Oo][Ww][Nn]" mapout="Critical" />
    <!--  Fatal   -->
    <mapentry mapin="[Rr][Ee][Aa][Dd][Yy]" mapout="Normal" />
```

```
    <mapentry mapin="[Aa][Bb][Aa][Nn][Dd][Oo][Nn] [Ss][Uu][Bb][Mm][Ii][Ss][Ss][Ii][Oo][Nn]" mapout="Unknown" /
>
    </Field>
   </Normalize>
   < Format>
    <!-- Correlatable properties, must populate at least one  -->
    <!--  Non-Correlatable properties  -->
    <Field output="MdrElementID" format="alert-{0}:{1}:{2}" input="snmp_agent,temp_applname,temp_jobname" />
    <Field output="OccurrenceTimestamp" format="{0}" input="{xsdateTime(now)}" />
    <Field output="ReportTimestamp" format="{0}" input="{xsdateTime(now)}" />
    <Field output="AlertType" format="{0}" input="Risk-Fault" />
    <Field conditional="severity" output="Severity" format="{0}" input="severity" />
    <Field output="AlertedMdrProduct" format="CA:00036" input="" />
    <Field output="AlertedMdrProdInstance" format="{0}" input="{fqdn(snmp_agent)}" />
    <!--  Assign instance name   -->
    <Field conditional="snmp_agent" output="Section1" format="{0}" input="snmp_agent" />
    <Field conditional="!snmp_agent" output="Flag" format="false" input="" />
    <Field conditional="temp_applname" output="Section2" format="{0}" input="temp_applname" />
    <Field conditional="!temp_applname" output="Flag" format="false" input="" />
    <Field conditional="temp_jobname" output="Section3" format="{0}" input="temp_jobname" />
    <Field conditional="!temp_jobname" output="Flag" format="false" input="" />
    <Field conditional="Flag" output="AlertedMdrElementID" format="" input="" />
    <Field conditional="!Flag" output="AlertedMdrElementID" format="{0}:{1}:{2}"
 input="Section1,Section2,Section3" />
    <Field output="Summary" format="{0}" input="temp_status" />
    <Field conditional="temp_state" output="Message" format="{0} alert on {1} scheduled on host {2}"
 input="temp_state,temp_jobname,snmp_agent" />
    <Field output="MetricName" format="{0}" input="Job Status" />
    <Field output="MetricType" format="{0}" input="Unknown" />
    <Field output="MetricUnitDefinition" format="{0}" input="Number" />
    <Field output="MetricDataType" format="{0}" input="String" />
   </Format>
   </EventClass>
   <!--  ======Event Class======   -->
   <EventClass name="USM-Entity" />
   </EventClass>
</Catalog>
```

Connector Policy Customization Best Practices

We recommend that you do not directly modify the connector policy files that are shipped with connectors. Because new versions and upgrades of a connector overwrite policy files, you will be required to reapply the custom content each time you upgrade or change the version of a connector.

TIP

We recommend that you use policy extensions to customize connector policy because they are not overwritten during an installation or upgrade.

- For more about CI modifications, see How to Add External Extensions to CI Types
- For more information about alert modifications, see Create an Event Policy with a Normalization Action.

CA Catalyst r3.4.1 Documentation

This section describes how to install and maintain CA Catalyst r3.4.1, which provides a lightweight container for hosting CA Catalyst connectors. This section is intended for product administrators who require the container to integrate connectors with their product.

CA Catalyst r3.4.1 Introduction

Contents

This section introduces CA Catalyst r3.4.1 and lists the new features in this release.

About CA Catalyst

CA Catalyst is a Container-only release that focuses on providing a lightweight Container on which to host connectors and build connector integrations. CA Catalyst does not include a CA Catalyst Server component, which means that CA Catalyst Server functionality such as the CA Directory Registry, Persistence Store database, correlation, reconciliation, and synchronization are not included. The Container uses a file-based registry.

The CA Catalyst r3.4.1 Container is specifically aligned to work with CA SOI by embedding the IFW Proxy, which enables communication between CA SOI and CA Catalyst r3.x connectors.

New Features

CA Catalyst r3.4.1 Container

- A simplified installation process to reduce the amount of user input required
- Native integration with CA SOI, requiring only a Container installation
- Performance improvements
- Enhanced transformation logging to better track policy errors
- Support for cluster and non-cluster high availability when integrating with CA SOI

CA Catalyst r3.4.1 Installation Planning

Contents

This section provides important information to know and actions to perform before installing CA Catalyst.

System Requirements and Software Support

The CA Catalyst Container supports installation on the following operating systems:

- Microsoft Windows Server 2012 (64-bit) Standard and Datacenter with the latest service packs
  Note: Cluster high availability implementations are not supported on Windows 2012.
- Microsoft Windows Server 2012 (64-bit) R2 Standard and Datacenter with the latest service packs
  Note: Cluster high availability implementations are not supported on Windows 2012.
- Microsoft Windows Server 2008 (32-bit and 64-bit) Standard, Enterprise, and Datacenter with the latest service packs
- Microsoft Windows Server 2008 (64-bit) R2 Standard, Enterprise, and Datacenter with the latest service packs

CA Catalyst Container systems require the following minimum hardware configuration:

- **Memory:** 6 GB (minimum)
- **Disk Space:** 20 GB free space
- **CPU:** Minimum two, 2.5 GHz

On a server with multiple Containers installed, allow at least 2GB of memory for each Container on the server.

You must have the following software installed on your server to use CA Catalyst:

- Java JRE Version 6.0
- One of the following versions of CA EEM on the CA Catalyst server or a remote server:
  - r8.4.x
  - r12.x
  - r12.5
  - r12.51

Installation Prerequisites

Before you install CA Catalyst, meet the following prerequisites as well as the requirements described in System Requirements:

Requirements for CA Catalyst Installation:

- Administrative permissions to complete the CA Catalyst installation.
- A CA EEM server must exist for CA Catalyst to leverage. The CA EEM server does not have to be local to the Container installation. You can leverage the same CA EEM server as an integrated product, such as CA SOI.
- If your CA EEM server is integrated with an Active Directory, you must add the user that you plan to create for CA Catalyst to the Active Directory before installing CA Catalyst.
- Know the following information:
  - Verify the installation server name and administrator login credentials.
  - We recommend a full backup of the installation server before installing CA Catalyst.
  - Verify that all servers where you plan to install CA Catalyst have their time and date settings synchronized. If the time is off between CA Catalyst nodes by more than five minutes, node to node communication fails.

Requirements for High Availability Installation

CA Catalyst supports Microsoft Cluster Server high availability on the following operating systems:

- Windows 2008
- Windows 2008 R2

Installation Best Practices

Consider the following best practices before starting your CA Catalyst installation:

- When integrating with CA SOI, install all CA SOI components before installing the CA Catalyst Container and connectors. The SA Manager must be in place for the Container to be able to integrate with CA SOI.
- You can install multiple Containers on separate servers or on the same server. Single-server multi-Container installations are useful when you want to consolidate your CA Catalyst installation on a single server and have a server large enough to accommodate the required capacity. For single-server multi-Container installations, each Container requires at least 2GB of memory.
- When installing multiple Containers on the same server, give subsequent installations a unique Container name. By default, connectors install into the 'CA Catalyst Container CatalystConnector' container. Specify an alternate container name during connector installation, if needed.
- All Containers, whether installed on a single server, or across multiple servers, should use the same CA EEM server and CA Catalyst credentials.
- For more information about connector installation best practices, see Install CA Catalyst Connectors.

Default Port Numbers and Connectivity

The following table shows the default port numbers that are used for CA Catalyst during installation and the components that require the ports:

Source	Port	Destination	Port	Protocol
Any component communicating with the Container	RP	Container	7000,8080	HTTP
Any component communicating with the Container	RP	Container	7443	HTTPS
Any component communicating with the Container	RP	Secondary Container installed on the same server	7100,8180	HTTP
Any component communicating with the Container	RP	Secondary Container installed on the same server	7543	HTTPS
Container	RP	CA EEM	5250	HTTP
Container	RP	CA SOI MQ Server	61616	JMS

CA Catalyst and Firewalls

When installing and running CA Catalyst in environments with firewalls, verify that communication between components on different servers occurs without interruptions.

In a dual-firewall environment, open port 8080 for inbound and outbound communication between external clients and the CA Catalyst Container.

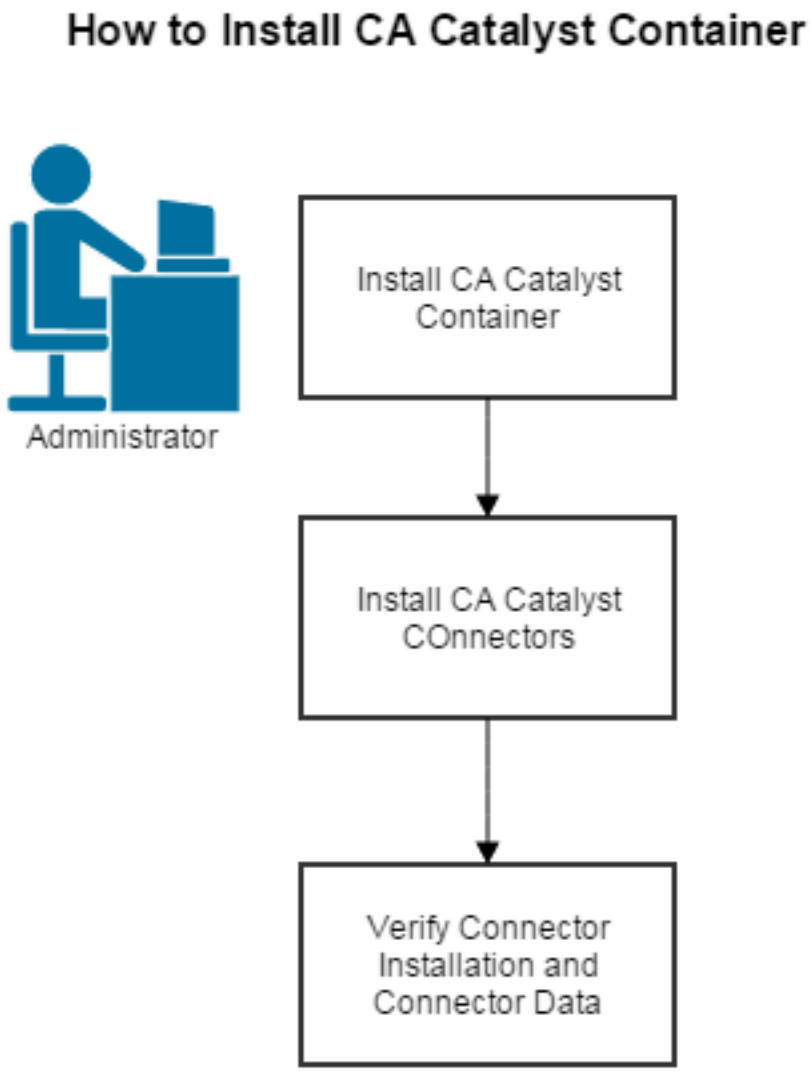
How to Install CA Catalyst

Contents

As an administrator, you install CA Catalyst to gain the benefit of a lightweight container that can collect and integrate information from key data sources into an integrated domain manager, such as CA SOI. You install CA Catalyst by installing Containers, and then installing connectors in those Containers to integrate with key data sources.

Use this scenario to guide you through the process:

Figure 62: Install CA Catalyst Container



1. Install the CA Catalyst Container:
   - Typical
   - Advanced
2. Install CA Catalyst connectors.
3. Verify the container installation and the connector data.

Install the CA Catalyst Container (Typical)

Install the CA Catalyst Container on the server where you want to install connectors. Perform a recommended installation for a simple installation experience that uses the default values for most settings.

Follow these steps:

1. Run **catalyst_install.bat** from the Disk1 folder of the CA Catalyst installation image.
2. Accept the license agreement and click **Next**.

3. Select **Typical** on the Installation Type page, select **Use CA Service Operations Insight** if you plan to integrate with CA SOI, and click **Next.**
4. Specify the installation directory, and proceed to the CA Service Operations Insight Configuration page, which appears only if you selected the Use CA CA Service Operations Insight check box on the Type of Installation page.
5. (CA SOI integration only) If you are integrating with CA SOI, enter valid information to connect to a CA SOI instance. Provide an active SA Manager host name**,** and valid CA SOI credentials. See your CA SOI installation worksheet for these values.
   If you configure a CA SOI integration, the installer automatically uses the CA SOI administrator credentials as the CA Catalyst administrator credentials. If you do not configure CA SOI integration, the Catalyst Administrator page opens.
6. Enter a username and password for the CA Catalyst administrator user if necessary, and click Next.
7. Enter a valid CA EEM server host name and credentials on the CA EEM Server Configuration page.
   The Application Name refers to the name under which CA EEM will manage the CA Catalyst entitlement information. Use the default unless another application is using this name on the same EEM instance. Multiple Containers can use the same application name.
   Leave the Proxy Host field empty unless you are integrating with a CA EEM instance in a cluster.
8. Review the information on the Install Summary page and click Install.
   The Container installs. Review the CATALYST_HOME*ContainerName*\logs\CA_Container_InstallLog.log file to troubleshoot installation errors.

Install the CA Catalyst Container (Advanced)

Install the CA Catalyst Container on the server where you want to install connectors. Perform an advanced installation of CA Catalyst to customize any of the following settings:

- Container name (for multiple Container installations on the same server)
- High availability settings
- Default ports
- 64-bit installation

Follow these steps:

1. Run **catalyst_install.bat** from the Disk1 folder of the CA Catalyst installation image.
2. Select **Advanced** on the Type of Installation page, select **Use CA Service Operations Insight** if you plan to integrate with CA SOI, and click **Next**.
   The New Catalyst Container page opens.
3. Do the following on the New Catalyst Container page:
   – Leave the default selection of Default Connector Container (CatalystConnector) unless you are installing multiple Containers on the same server. Select **Custom Container ID** and enter a custom Container name in the Container Name field to give a secondary Container a unique name, if necessary.
   – If you want the Container to participate in a CA ARCserve RHA failover scenario, select the **'Configure this container for CA ARCserve Replication and High Availability'** check box, and define a common prefix for the Container in the Common Prefix field. This prefix ensures a common name for all of the Containers involved in the failover scenario so that the failover works and only a single Container appears in the various interfaces. In addition to the prefix, Container IDs must also match across Containers participating in the failover scenario.

     NOTE

     If you add a prefix, the Container ID remains whatever you specify in the ContainerID fields, and appears without the prefix in areas such as the Windows service. The prefix appears primarily in the interfaces for connector management. For more information about including the Container in a CA ARCserve Replication and High Availability failover scenario, see Non-cluster High Availability Implementation.
4. Select the installation directory, and proceed to the Configure High Availability page.
5. (Optional) Select Configure for High Availability and enter the necessary information if you are deploying the CA Catalyst Container in a cluster environment.

For more information about the high availability settings and implementing CA Catalyst in a cluster environment, see How to Perform a High Availability Implementation.
The CA Service Operations Insight (SOI) Configuration page opens if you selected the Use CA Service Operations Insight check box on the Type of Installation page.

6. (CA SOI integration only) Enter valid information to connect to a CA SOI instance if you are integrating with CA SOI. Provide an active SA Manager hostname, active MQ Server hostname, and valid CA SOI credentials. See your CA SOI installation worksheet for these values.
   If you configure CA SOI integration, the installer automatically uses the CA SOI administrator credentials as the CA Catalyst administrator credentials. If you do not configure CA SOI integration, the Catalyst Administrator page opens.
7. Enter a username and password for the CA Catalyst administrator user if necessary, and click Next.
8. Enter a valid CA EEM server host name and credentials on the CA EEM Server Configuration page.
   The Application Name refers to the name under which CA EEM will manage the CA Catalyst entitlement information. Use the default unless another application is using this name on the same EEM instance. Multiple Containers can use the same application name.
   Leave the Proxy Host field empty unless you are integrating with a CA EEM instance in a cluster.
   The Catalyst Container Configuration page opens.
9. Accept the default values unless the ports are already in use.
   If you are installing a second Container on the same server, the installer automatically updates the default ports to secondary values. If you are installing anything more than a second Container on the same server, change the ports manually to avoid conflicts.
   When installing on a 64-bit system, the 64-bit JRE Configuration page opens.
10. (64-bit systems only) Select to use a 64-bit JRE to run the Container as a 64-bit application, and modify the maximum Java Heap size, if necessary.
11. Review the information on the Install Summary page and click Install.
    The Container installs. Review the CATALYST_HOME*ContainerName*\logs\CA_Container_InstallLog.log file to troubleshoot installation errors.

Install CA Catalyst Connectors

Install CA Catalyst connectors on the same server where you installed the CA Catalyst Container.

Consider the following items before installing CA Catalyst connectors:

- Each CA Catalyst connector comes with a product-specific *Connector Guide*. See this document for information about the connector functionality, installation, and configuration.
- Verify that the connector supports the CA Catalyst version you are using before installation.
- You can install multiple connectors in the same CA Catalyst Container. Alternatively, you can install connectors on the same server into multiple CA Catalyst Containers with different Container names. Using dedicated CA Catalyst Containers on the same server lets each connector run under its own process.

Verify Container Installation and Connector Data

To verify a successful CA Catalyst Container and connector implementation, verify the following items:

- Container status
- Connector existence in the container
- Connector status and data in the consuming product, such as CA SOI

Follow these steps:

1. View the Windows services on the Container server.
2. Find the CA Catalyst Container *ContainerName* service, and verify that it is running. Start the service, if required.
   This service indicates that the Container exists and is active.
3. Enter the following URL in a web browser:

```
http://<containerserver>:7000/node/rest/
```

An XML page opens. All connectors installed on the Container should appear as a module using the following nomenclature:

```
CA:09998_<ContainerServer>_CatalystConnector
```

The five digit number is unique for each connector (09998 is the ID for the Sample connector).

4. (CA SOI integration only) Open the CA SOI Dashboard using the following URL:

   ```
   http://<SoiUIServer>:7070/sam/ui
   ```

   Log in using CA SOI credentials.
5. Click the Administration tab, and expand Connector Configuration.
   You should see an entry for the installed Container. If you defined a common prefix for use in a failover scenario, the Container name includes the prefix.
6. Expand the Container entry to see a list of connectors on the Container. Each connector entry provides the connector status and connection details.

CA Catalyst Log Files

CA Catalyst generates the following log files that you can use to troubleshooting installation and runtime errors:

Container Installation Logs

The Container installation log files, CA_Container_InstallLog.log and CA_osgi_InstallLog.log, provide details about your Container installation and indicate errors that occurred during the installation.

Location: C:\Program Files\CA\Catalyst*ContainerName*\logs

Uninstall Log

The CA-Catalyst_UnInstaller.log file is created when you uninstall CA Catalyst.

Location: %TEMP%

Upgrade Log

The CA_Catalyst_UpgradeLog.log file is created after you upgrade CA Catalyst. The Upgrade log provides details about the CA Catalyst upgrade and indicates errors that occurred during the upgrade.

Location: C:\Program Files\CA\Catalyst*ContainerName*\jsw\logs

Installation Debug Log

The CatalystInstallDebug.log file provides additional debugging information that is related to the CA Catalyst installation.

Location: %TEMP%

Container Wrapper Log

The CatalystContainer_wrapper.log file provides details about the CA Catalyst Container *ContainerName* service and indicates if the service is installed or running, or errors are encountered.

Location: C:\Program Files\CA\Catalyst*ContainerName*\jsw\logs

When the CA Catalyst Container *ContainerName* service starts correctly, the following line appears in the log:

```
impl.ContainerImpl --- - Catalyst Container <ContainerName> started successfully.
```

Uninstall the CA Catalyst Container

When you uninstall CA Catalyst, uninstall connectors before uninstalling the Container to ensure a clean Container removal. After you uninstall all CA Catalyst components, remove the CA Catalyst application instance from CA EEM to complete the uninstallation process.

Follow these steps:

1. Uninstall any connectors on the Container system from the Windows Start menu.

   NOTE
   For more information about uninstalling a specific connector, see the *Connector Guide* for that connector.
2. (CA SOI integration only) Log in to the CA SOI Administration UI, select the connector entries that you uninstalled, and click Remove Connector for each connector.
   This step unregisters the CA Catalyst connector from the UCF Broker. If you skip this step and later reinstall the connector, the Synchronizer may try to send events to the connector and may get stuck.

   NOTE
   For more information about removing connectors from the CA SOI Administration UI, see Connector Removal.
3. Select Start, Programs, CA, Catalyst, Uninstall CA Catalyst Container for *ContainerName* on the Container server.
4. Proceed through the uninstallation dialog prompts and click Uninstall.
   The Container uninstalls. If the Uninstall Complete dialog displays items that did not uninstall, clean up those items manually.
5. Select Start, Programs, CA, Embedded Entitlements Manager, EEM UI on the CA EEM server.
   The login page of the CA EEM user interface opens.
6. Enter CA EEM administrator credentials, and click Log In.
   The Home tab of the CA EEM user interface opens.
7. Click the Configure tab.
8. Click the application instance that was created for CA Catalyst. By default, this instance is prefixed with catalyst.
   The Application Instance pane opens with details about the instance.
9. Click Unregister, and click OK.
   The CA EEM application instance for CA Catalyst is removed.

Upgrade the CA Catalyst Container

Contents

This section provides information about the supported CA Catalyst Container upgrade scenarios.

Upgrade from a Previous Version of the CA Catalyst Container

Upgrade the CA Catalyst Container to take advantage of new features. CA Catalyst r3.4.1 supports upgrade from the following releases:

- A standalone CA Catalyst r3.2 Container with a file-based registry
- A standalone CA Catalyst r3.2 Container with a file-based registry integrated with the IFW Proxy
- A previous build of the CA Catalyst r3.4.1 Container

If you upgrade a Container that is connected to an IFW Proxy, the installer upgrades the Container and the IFW Proxy in place. You cannot add an embedded IFW Proxy to the r3.4.1 Container during upgrade if the r3.2 Container did not already integrate with an IFW Proxy.

If you are using a previous version of CA Catalyst or r3.2 Containers that connect to a CA Catalyst Server, uninstall them and perform a clean installation of the r3.4.1 Container.

When you upgrade the Container, all settings carry forward, such as the CA Catalyst user credentials and ports.

NOTE

For information about upgrading connectors, see the appropriate *Connector Guide*.

Follow these steps:

1. Run the Container installer on a server with a standalone Container installed.
2. Proceed to the Type of Installation page, and do the following depending on your Container name and CA SOI integration status:
   – Select the Use CA Service Operations Insight check box if your Container integrates with CA SOI through and IFW Proxy. Clear this check box when the Container does not integrate with CA SOI. If you check this box and the installer does not detect an IFW Proxy, the installer prompts to you clear the box.
   – If the Container has the default name of CatalystConnector, select Recommended and click Next.
   – If the Container uses a custom Container name, select Advanced, enter the Container name on the New Catalyst Container page, and click Next.

   The Previous Catalyst Installation Found dialog opens.
3. Click Yes to upgrade.
4. Review your settings on the Install Summary page, and click Install.
   The Container upgrades. Check the CA_Catalyst_UpgradeLog.log file located at C:\Program Files\CA\Catalyst *ContainerName*\jsw\logs to verify the upgrade and troubleshoot errors. You can also perform the verification steps described in Verify Container Installation and Connector Data.

During upgrade, the installer takes a backup of the previous CA Catalyst installation and stores the backup information in the CATALYST_HOME\ContainerName\upgrade_backup_*releasenumber* directory.

> **NOTE**
>
> : If you configured the Container to use a Java heap size other than the default of 1024, the upgrade reverts to the default setting. Reconfigure the custom setting after the upgrade completes.

How to Perform a CA Catalyst High Availability Implementation

Contents

As an administrator, you are responsible for deploying a reliable CA Catalyst solution and ensuring that downtime does not adversely affect the solution. High availability support for CA Catalyst Containers provides failover capabilities and a solution for applying maintenance while avoiding product downtime. Container high availability includes the connectors that the Container hosts, ensuring that no connector data or connectivity is lost.

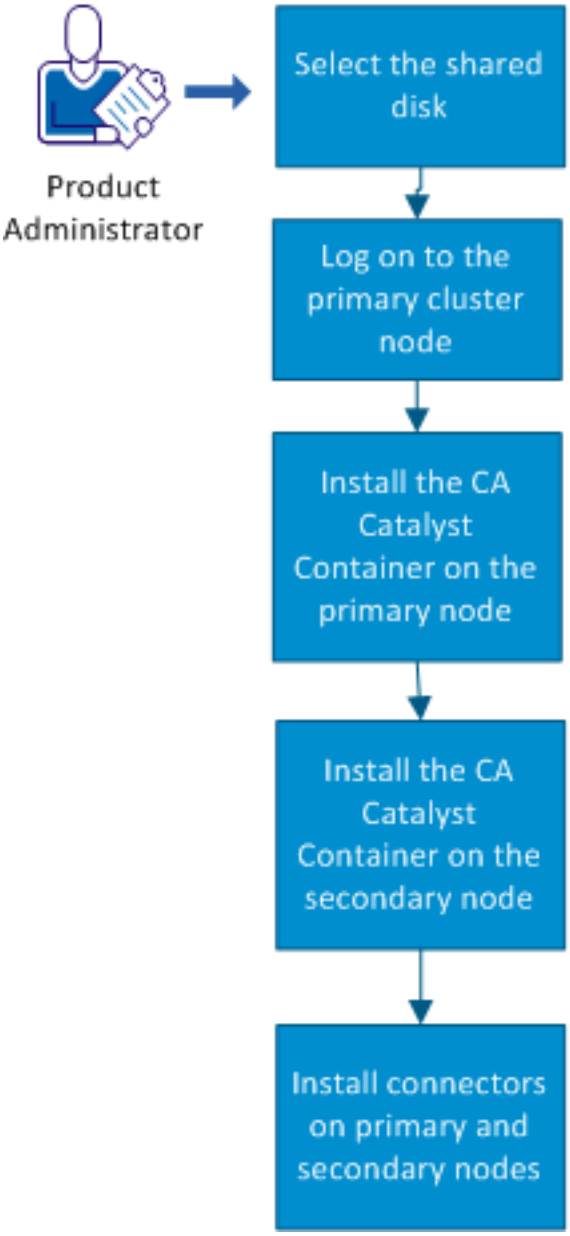
> **NOTE**
>
> CA Catalyst also supports a non-cluster high availability implementation using CA ARCserve Replication and High Availability. For more information about a non-cluster high availability implementation, see Non-cluster High Availability Implementation.

Consider the following before implementing CA Catalyst Containers for high availability:

- Install Containers on at least two cluster nodes. If you select High Availability and only install a Container on a single node, the primary node will not work as a stand-alone node.
- If you uninstall a secondary node, the primary node will not continue to work as a stand-alone node and must be uninstalled.
- Any patches included with the Container package are not installed during a high availability implementation. Apply any necessary Container patches using the stand-alone patch installer.
- If you are integrating with a high availability implementation of CA SOI, you can install Containers on the same cluster as CA SOI.

Use the following scenario to guide you through the process:

How to Perform a High Availability CA Catalyst Container Implementation



Complete the following process to implement CA Catalyst Containers and connectors in a cluster environment:

1. Select the shared disk.
2. Log on to the primary node of the selected cluster group.
   This node should be the owner of the shared disk that you are using for CA Catalyst data. To determine shared disk ownership, open the Windows Failover Cluster Manager and select Storage, Available Storage. Select the disk to see the owner.
3. Install the Catalyst Container on the primary node.
4. Install the Catalyst Container on the secondary node.

When you install the CA Catalyst Container on the last node, the installer brings all CA Catalyst Container resources online on the primary node.

5. Install connectors on primary and secondary nodes.

Note: You can also install CA Catalyst Containers and connectors with a load balancer for implicit high availability support and load distribution.

See the following procedures for supplemental information about maintaining, troubleshooting, and removing your HA Container environment:

- Add Nodes to an Existing CA Catalyst HA Implementation
- High Availability Troubleshooting
- Uninstall HA CA Catalyst Implementation
- Configure CA Catalyst Containers with a Load Balancer

Select the Shared Disk

A shared disk must be available in the cluster environment when you run the CA Catalyst Container installer. Make the shared disk available as a Cluster Resource and bring it online before CA Catalyst Container installation. Verify that the disk appears as a Cluster Resource in the Available Storage group in the Windows 2008 Failover Cluster Management Console. The shared disk must have enough space to accommodate the requirements for a CA Catalyst Container installation.

Install the CA Catalyst Container on the Primary Node

Install the CA Catalyst Container on the primary node that you identified as the owner of the selected shared disk.

Consider the following options for integrating your HA CA Catalyst Container and connector with the domain manager and the consuming product:

- The CA Catalyst Container can exist on primary and secondary nodes in the same cluster as an HA domain manager. For example, if you have Microsoft SCOM implemented in a cluster environment, you can configure the CA Catalyst Containers and Microsoft SCOM connectors to use the same resource group in the same cluster.
- HA CA Catalyst Containers can also exist in a cluster that is separate from an HA domain manager, if necessary.
- HA CA Catalyst Containers can participate in a high availability configuration of the consuming product. For example, if integrating with CA SOI, you can install CA Catalyst Containers into the same cluster group.

  NOTE
  During high availability deployment, the data typically installed to the CATALYST_HOME *ContainerName*\container folder is moved to the shared disk. This data includes log files required for diagnostics and troubleshooting. To access these log files in a high availability deployment, access the Catalyst *ContainerName*\container folder or Catalyst*ContainerName*\container folder on the shared disk that you specified during the CA Catalyst Container HA configuration.

Follow these steps:

1. Run catalyst_install.bat from the CA Catalyst Container installation image on the active node.
2. Select Advanced for the type of installation. Navigate through the installation panels using the instructions in Install the CA Catalyst Container (Advanced) until you reach the Configure High Availability page.

   NOTE
   The Recommended installation type does not offer an option to select High Availability.
3. Select Configure for High Availability, enter the following information, and click Next:
   - **Primary Node**
     Specifies that the current node is the primary node in a high availability cluster. Select this option, and leave the Secondary Node option cleared.
   - **New Cluster Group/Existing Cluster Group**

If you are installing the HA CA Catalyst Container on a separate cluster than the HA domain manager deployment, select New Cluster Group. If you are installing the HA CA Catalyst Container on the same cluster as an HA domain manager deployment or on the same cluster as CA SOI, select Existing Cluster Group.

- **Cluster Group**
  Defines the cluster group name. Define a new group, or if you selected Existing Cluster Group, define the existing cluster group to use for the CA Catalyst Container resources in the Existing Cluster Group field.
- **Shared Disk Name**
  (New cluster group only) Defines the name of the shared disk on which to store CA Catalyst data. You should enter the shared disk name as it appears in the Failover Cluster Manager on the cluster server.
- **Virtual Node IP Address**
  (New cluster group only) Defines the IP address that you map to the virtual node name.
- **Subnet Mask**
  (New cluster group only) Defines the subnet mask that the virtual node name uses.

– **Virtual Node Name**
Defines the virtual CA Catalyst Container network name that CA Catalyst uses in the cluster environment. This name is the application name mapped to the IP address in the Domain Controller. If this value does not exist, ask your network administrator to assign one for the CA Catalyst Container in a high availability configuration.

– **Shared Directory**
Defines the shared directory. All the nodes in the cluster environment use the shared directory data.

The CA Service Operations Insight Configuration page opens.

4. Proceed through the installation pages using the information in Install the CA Catalyst Container (Advanced). Do the following to prepare for high availability implementation:
   – (CA SOI integration only) When integrating with a high availability implementation of CA SOI, enter the virtual node name of the SA Manager in the SOI Manager Host field.
   – Note the passwords, port numbers, and other connection information that you enter on all pages. Secondary node installations require the same information for most values.
5. Click Install on the Install Summary page.
   The installation begins, and the Install Complete page opens when finished. The CA Catalyst Container is installed on the active node.

Install the CA Catalyst Container on Secondary Nodes

To configure high availability between nodes, install the CA Catalyst Container on all necessary secondary nodes.

> **WARNING**
> When installing the CA Catalyst Container on secondary nodes, the primary node must remain the active node.

Follow these steps:

1. Run catalyst_install.bat from the CA Catalyst Container installation image on the secondary node.
   The Introduction page opens.
2. Select Advanced for the type of installation. Navigate through the installation pages using the instructions in Install the CA Catalyst Container (Advanced) until you reach the Configure High Availability page.
3. Do the following on the Configure High Availability page:
   – Select Secondary Node.
   – Select Bring the Cluster Resources Online on the last CA Catalyst Container node installation. If you plan on installing the CA Catalyst Container on additional secondary nodes, do not select this option.
   – Define the cluster group that you used during the primary node installation in the Existing Cluster Group field.
   – Enter the same values for Virtual Node Name and Shared Directory that you entered during the primary node installation.
4. Do the following on ensuring installation pages to prepare for high availability implementation:

- (CA SOI integration only) When integrating with a high availability implementation of CA SOI, enter the virtual node name of the SA Manager in the SOI Manager Host field. Enter the same CA SOI information as you did during the primary Container installation.
- Enter the same passwords and port numbers that you did during CA Catalyst Container primary node installation.

5. Click Install on the Install Summary page.
   The installation begins, and the Install Complete page opens when finished. The CA Catalyst Container is installed on the secondary node.
6. (Optional) Repeat Steps 1-5 to install the CA Catalyst Container on additional nodes. Select Bring the Cluster Resources Online on the Configure High Availability page during the last node installation.
   During installation of the last node, the installer brings all cluster resources online, completing the remote CA Catalyst Container high availability configuration.
7. Verify that the CA Catalyst Container service is online on the primary node.
8. Open Cluster Manager and add dependencies between the Containers as follows:
   - Add a dependency to the secondary Container to start only after the primary Container.
   - Add dependencies to additional Containers, if necessary. For example, a tertiary Container should start only after the secondary Container.

     NOTE
     These dependencies prevent resource constraints that could occur if all Containers fail over at the same time.

Repeat this entire process if you require additional CA Catalyst Containers hosting different connectors.

Install Connectors on Primary and Secondary Nodes

Install the CA Catalyst connectors that you want the HA Container to host on the primary and secondary CA Catalyst Container nodes. When the CA Catalyst Container fails over, it can then use the CA Catalyst connector installed on the secondary node to prevent any interruptions of connectivity with the domain manager.

Follow these steps:

1. Install connectors on the primary node. When prompted for domain manager Container information, enter the virtual node name associated with the domain manager to connect to an HA domain manager deployment. For domain managers not configured for high availability, enter the server name associated with the domain manager.
2. Install connectors on the secondary node. Provide the same information during connector installation that you did for the corresponding connector on the primary node.
3. (Optional) Repeat the installation on additional nodes, if necessary.

Add Nodes to an Existing CA Catalyst HA Implementation

You can add additional Windows CA Catalyst nodes to an existing CA Catalyst HA cluster. Before adding the node, verify that you can see the additional node in the Cluster Failover Manager as a part of the cluster.

Follow these steps:

1. Bring all of the CA Catalyst Cluster Resources offline using the Cluster Failover Manager.
2. Install CA Catalyst on the additional node, following the same conventions that you did when installing the CA Catalyst Container on secondary nodes.
3. On the last additional node installation, specify to bring the cluster resources online.

High Availability Troubleshooting

Symptom:

After installation, the CA Catalyst services do not start.

Solution:

Try the following troubleshooting steps:

- Verify that the Resource IP Address, Network Name, Physical Disk and CA Catalyst Container are successfully created in the cluster.
- Ping and telnet the Network Name that is used as the CA Catalyst virtual node name. If the ping is unsuccessful, the network name or the IP Address is invalid.
- Telnet the Container from a separate server. If telnet is unsuccessful, resolve the network dependencies. Modify the firewall setting on the cluster, and add CA Catalyst ports 7000 and 7443 to the inbound rules.
- Verify the logs:
  - Verify the log CATALYST_HOME*ContainerName*\logs\CA_Catalyst_InstallLog.log for errors. This log file captures the shared disk resource creation failure and indicates an incorrect shared disk name or shared directory.
  - Verify the Container log in the shared disk location for the Container startup exceptions.
  - Verify the log CATALYST_HOME\HA\container_ha_res_creation.log and the container_ha_status.log files for resource creation errors.

Uninstall HA CA Catalyst Implementation

Uninstall an HA CA Catalyst implementation in a specific order.

> **NOTE**
> The uninstallation does not remove the cluster group. Remove the cluster group manually.

Follow these steps:

1. Uninstall the CA Catalyst Container and connectors on all secondary nodes.
2. Uninstall the CA Catalyst Container and connectors on the primary node.
3. (Optional) Repeat this process for any clusters with additional HA CA Catalyst Containers.

Configure CA Catalyst Containers with a Load Balancer

You can configure a CA Catalyst Container with a load balancer for distributing load or for high availability. The CA Catalyst Container maintains the state of the connector. Therefore, to support HA implicitly, it needs the capability to replicate the state data to a shared store so that all nodes in the HA environment share the same state.

Using a load balancer differs from cluster high availability in that it is primarily for scalability, and all load balancer nodes are always live. Both methods use a virtual or proxy node name to refer to all nodes.

You can configure any load balancer for HA or distributing load when the Container hosts connectors that get only inbound calls. This limitation applies because the Container itself is stateless.

This procedure describes how to configure CA Catalyst Containers and connectors to use Apache HTTPD as a load balancer.

Follow these steps:

1. Install the CA Catalyst Container on all necessary nodes.
2. Install the targeted CA Catalyst connector on all of the nodes.

   > **NOTE**
   > During connector installation, ensure that the connector configuration on all nodes has the same ModuleInstance.
3. Update the following configuration section in the HTTPD.conf file:

```
<VirtualHost *:80>
????ProxyPass /catalyst balancer://mycluster/context
????ProxyPassReverse /ucf balancer://mycluster/context
```

```
????<Proxy balancer://mycluster>
????????BalancerMember http://Server1:7000/node
????????BalancerMember http://Server2:7000/node
????</Proxy>
</VirtualHost>
```

4. Restart the Apache HTTPD service.

With this configuration, the CA Catalyst Core API endpoints are available on the following URLs:

- http://<*Loadbalancer_HostName*>/catalyst/wsman
- http://<*Loadbalancer_HostName*>/catalyst/rest
- http://<*Loadbalancer_HostName*>/catalyst/odata

Now CA Catalyst clients can use these URLs instead of individual node URLs.

CA Catalyst r3.4.2 Documentation

This section describes how to install and maintain CA Catalyst r3.4.2, which provides a lightweight container for hosting CA Catalyst connectors. This section is intended for product administrators who require the container to integrate connectors with their product.

New Features

This release supports decouple MQ Server. To connect to MQ Server, provide the MQ Server details in CA Catalyst r3.4.2 installer.

To install CA Catalyst r3.4.2, follow these steps:

1. CA Catalyst r3.4.2 Installation Planning
2. How to install CA Catalyst r3.4.2
3. CA Catalyst Log Files
4. Uninstall the CA Catalyst Container
5. How to Perform a CA Catalyst r3.4.2 High Availability Implementation

CA Catalyst r3.4.2 Installation Planning

System Requirements and Software Support

The CA Catalyst Container supports installation on the following operating system:

- Microsoft Windows Server 2008 (64-bit) Standard, Enterprise, and Datacenter with the latest service packs
- Microsoft Windows Server 2008 (64-bit) R2 Standard, Enterprise, and Datacenter with the latest service packs
- Microsoft Windows Server 2012 (64-bit) Standard and Datacenter with the latest service packs
- Microsoft Windows Server 2012 (64-bit) R2 Standard and Datacenter with the latest service packs

Requirements for High Availability Installation

CA Catalyst supports Microsoft Cluster Server high availability on the following operating systems:

- Windows 2012
- Windows 2012 R2
- Windows 2008
- Windows 2008 R2

CA Catalyst Container systems require the following minimum hardware configuration:

- **Memory:** 6 GB (minimum)
- **Disk Space:** 20 GB free space
- **CPU:** Minimum two, 2.5 GHz

On a server with multiple Containers installed, allow at least 2GB of memory for each Container on the server.

You must have the following versions of CA EEM on the CA Catalyst server or a remote server:

- – r12.51
  – r12.5
  – r12.x
  – r8.4.x

  NOTE

  For more information about CA Catalyst installation prerequisites, best practices, and default port numbers and connectivity, see CA Catalyst Installation Planning.

How to Install CA Catalyst r3.4.2

As an administrator, you install CA Catalyst to gain the benefit of a lightweight container that can collect and integrate information from key data sources into an integrated domain manager, such as CA SOI. You install CA Catalyst by installing Containers and then installing connectors in those Containers to integrate with key data sources.

WARNING

Upgrade of CA Catalyst r3.4.2 from previous CA Catalyst versions is not supported.

Prerequisites:

Ensure that MQ Server is installed on a system and note the MQ server details.

Install the CA Catalyst Container (Typical)

Install the CA Catalyst Container on the server where you want to install connectors. Perform a recommended installation for a simple installation experience that uses the default values for most settings.

Follow these steps:

1. Run **catalyst_install.bat** from the Disk1 folder of the CA Catalyst installation image.
2. Accept the license agreement and click **Next**.
3. Select **Typical** on the Installation Type page, select **Use CA Service Operations Insight** if you plan to integrate with CA SOI, and click **Next.**
4. Specify the installation directory, and proceed to the CA Service Operations Insight Configuration page, which appears only if you selected the Use CA Service Operations Insight check box on the Type of Installation page.
5. (CA SOI integration only) If you are integrating with CA SOI, enter valid information to connect to a CA SOI instance. Provide an active SA Manager host name, MQ Server host name, and valid CA SOI credentials. See your CA SOI installation worksheet for these values.
   If you configure a CA SOI integration, the installer automatically uses the CA SOI administrator credentials as the CA Catalyst administrator credentials. If you do not configure CA SOI integration, the Catalyst Administrator page opens.
6. Enter a username and password for the CA Catalyst administrator user if necessary, and click Next.
7. Enter a valid CA EEM server host name and credentials on the CA EEM Server Configuration page.
   The Application Name refers to the name under which CA EEM manages the CA Catalyst entitlement information. Use the default unless another application is using this name on the same EEM instance. Multiple Containers can use the same application name.
   Leave the Proxy Host field empty unless you are integrating with a CA EEM instance in a cluster.
8. Review the information on the Install Summary page and click Install.

The Container installs. Review the CATALYST_HOME*ContainerName*\logs\CA_Container_InstallLog.log file to troubleshoot installation errors.

Install the CA Catalyst Container (Advanced)

Install the CA Catalyst Container on the server where you want to install connectors. Perform an advanced installation of CA Catalyst to customize any of the following settings:

- Container name (for multiple Container installations on the same server)
- High availability settings
- Default ports
- 64-bit installation

Follow these steps:

1. Run **catalyst_install.bat** from the Disk1 folder of the CA Catalyst installation image.
2. Select **Advanced** on the Type of Installation page, select **Use CA Service Operations Insight** if you plan to integrate with CA SOI, and click **Next**.
   The New Catalyst Container page opens.
3. Do the following on the New Catalyst Container page:
   – Leave the default selection of Default Connector Container (CatalystConnector) unless you are installing multiple Containers on the same server. Select **Custom Container ID** and enter a custom Container name in the Container Name field to give a secondary Container a unique name, if necessary.
   – If you want the Container to participate in a CA ARCserve RHA failover scenario, select the '**Configure this container for CA ARCserve Replication and High Availability**' check box, and define a common prefix for the Container in the Common Prefix field. This prefix ensures a common name for all the Containers that are involved in the failover scenario so that the failover works and only a single Container appears in the various interfaces. In addition to the prefix, Container IDs must also match across Containers participating in the failover scenario.

     NOTE

     If you add a prefix, the Container ID remains whatever you specify in the ContainerID fields, and appears without the prefix in areas such as the Windows service. The prefix appears primarily in the interfaces for connector management. For more information about including the Container in a CA ARCserve Replication and High Availability failover scenario, see Non-cluster High Availability Implementation.
4. Select the installation directory, and proceed to the Configure High Availability page.
5. (Optional) Select Configure for High Availability and enter the necessary information if you are deploying the CA Catalyst Container in a cluster environment.
   For more information about the high availability settings and implementing CA Catalyst in a cluster environment, see How to Perform a High Availability Implementation.
   The CA Service Operations Insight (CA SOI) Configuration page opens if you selected the Use CA Service Operations Insight check box on the Type of Installation page.
6. (CA SOI integration only) Enter valid information to connect to a CA SOI instance if you are integrating with CA SOI. Provide an active SA Manager host name, MQ Server host name, and valid CA SOI credentials. See your CA SOI installation worksheet for these values.
   If you configure CA SOI integration, the installer automatically uses the CA SOI administrator credentials as the CA Catalyst administrator credentials. If you do not configure CA SOI integration, the Catalyst Administrator page opens.
7. Enter a username and password for the CA Catalyst administrator user if necessary, and click Next.
8. Enter a valid CA EEM server host name and credentials on the CA EEM Server Configuration page.
   The Application Name refers to the name under which CA EEM manages the CA Catalyst entitlement information. Use the default unless another application is using this name on the same EEM instance. Multiple Containers can use the same application name.
   Leave the Proxy Host field empty unless you are integrating with a CA EEM instance in a cluster.
   The Catalyst Container Configuration page opens.

9. Accept the default values unless the ports are already in use.
   If you are installing a second Container on the same server, the installer automatically updates the default ports to secondary values. If you are installing anything more than a second Container on the same server, change the ports manually to avoid conflicts.
   When installing on a 64-bit system, the 64-bit JRE Configuration page opens.
10. (64-bit systems only) Select to use a 64-bit JRE to run the Container as a 64-bit application, and modify the maximum Java Heap size, if necessary.
11. Review the information on the Install Summary page and click Install.
    The Container installs. Review the CATALYST_HOME*ContainerName*\logs\CA_Container_InstallLog.log file to troubleshoot installation errors.

Install CA Catalyst Connectors

Install CA Catalyst connectors on the same server where you installed the CA Catalyst Container.

Consider the following items before installing CA Catalyst connectors:

- Each CA Catalyst connector comes with a product-specific *Connector Guide*. See this document for information about the connector functionality, installation, and configuration.
- Verify that the connector supports the CA Catalyst version that you are using before installation.
- You can install multiple connectors in the same CA Catalyst Container. Alternatively, you can install connectors on the same server into multiple CA Catalyst Containers with different Container names. Using dedicated CA Catalyst Containers on the same server lets each connector that is run under its own process.

Verify Container Installation and Connector Data

To verify a successful CA Catalyst Container and connector implementation, verify the following items:

- Container status
- Connector existence in the container
- Connector status and data in the consuming product, such as CA SOI

Follow these steps:

1. View the Windows services on the Container server.
2. Find the CA Catalyst Container *ContainerName* service, and verify that it is running. Start the service, if necessary. This service indicates that the Container exists and is active.
3. Enter the following URL in a Web browser:
   ```
   http://<containerserver>:7000/node/rest/
   ```
   An XML page opens. All connectors that are installed on the Container should appear as a module using the following nomenclature:
   ```
   CA:09998_<ContainerServer>_CatalystConnector
   ```
   The five-digit number is unique for each connector (09998 is the ID for the Sample connector).
4. (CA SOI integration only) Open the CA SOI Dashboard using the following URL:
   ```
   http://<SoiUIServer>:7070/sam/ui
   ```
   Log in using CA SOI credentials.
5. Click the Administration tab, and expand Connector Configuration.
   You should see an entry for the installed Container. If you defined a common prefix for use in a failover scenario, the Container name includes the prefix.
6. Expand the Container entry to see a list of connectors on the Container. Each connector entry provides the connector status and connection details.

How to Perform a CA Catalyst r3.4.2 High Availability Implementation

High availability support for CA Catalyst Containers provides failover capabilities and a solution for applying maintenance while avoiding product downtime. Container high availability includes the connectors that the Container hosts, ensuring that no connector data or connectivity is lost.

NOTE

For more information about CA Catalyst r3.4.2 High Availability Implementation, see How to Perform a CA Catalyst r3.4.2 High Availability Implementation.

CA Catalyst r3.4.3 Documentation

This section describes how to install and maintain the current version of CA Catalyst, which provides a lightweight container for hosting CA Catalyst connectors. This section is intended for product administrators who require the container to integrate connectors with their product.

Support for AdoptOpenJDK JRE

CA Technologies, a Broadcom Company, is moving towards adopting more open source technologies in its products. As a part of this strategy, various products have started using open-source implementations of Java. To align with this corporate direction, CA SOI has adopted AdoptOpenJDK (1.8.0.212), replacing Oracle JDK. After you install the current patch, the jre, jre-32, jre-64 directories under the SOI_HOME is overwritten with the new JRE.

IMPORTANT
In case you delete these directories or uninstall the patch the Service Operations Insight application services stop.

CA Catalyst Installation Planning

System Requirements and Software Support

The CA Catalyst Container supports installation on the following operating system:

- Microsoft Windows Server 2008 (64-bit) Standard, Enterprise, and Datacenter with the latest service packs
- Microsoft Windows Server 2008 (64-bit) R2 Standard, Enterprise, and Datacenter with the latest service packs
- Microsoft Windows Server 2012 (64-bit) Standard and Datacenter with the latest service packs
- Microsoft Windows Server 2012 (64-bit) R2 Standard and Datacenter with the latest service packs

How to Install CA Catalyst

As an administrator, you install CA Catalyst to gain the benefit of a lightweight container that can collect and integrate information from key data sources into an integrated domain manager, such as CA SOI. You install CA Catalyst by installing Containers and then installing connectors in those Containers to integrate with key data sources.

Important: Upgrade to CA Catalyst r3.4.3 from previous CA Catalyst versions is not supported.

Prerequisites:

Ensure that MQ Server is installed on a system and note the MQ server details. Install the CA Catalyst Container (Typical)Install the CA Catalyst Container on the server where you want to install connectors. Perform a recommended installation for a simple installation experience that uses the default values for most settings.

Follow these steps:

1. Run **catalyst_install.bat** from the Disk1 folder of the CA Catalyst installation image.
2. Accept the license agreement and click **Next**.
3. Select **Typical** on the Installation Type page, select **Use CA Service Operations Insight** if you plan to integrate with CA SOI and click **Next.**

4. Specify the installation directory, and proceed to the CA Service Operations Insight Configuration page, which appears only if you selected the Use CA Service Operations Insight checkbox on the Type of Installation page.
5. (CA SOI integration only) If you are integrating with CA SOI, enter valid information to connect to a CA SOI instance. Provide an active SA Manager hostname, MQ Server hostname, and valid CA SOI credentials. If you configure a CA SOI integration, the installer automatically uses the CA SOI administrator credentials as the CA Catalyst administrator credentials. If you do not configure CA SOI integration, the Catalyst Administrator page opens.
6. Enter a username and password for the CA Catalyst administrator user if necessary, and click Next.
7. Enter a valid CA EEM server hostname and credentials on the CA EEM Server Configuration page. The Application Name refers to the name under which CA EEM manages the CA Catalyst entitlement information. Use the default unless another application is using this name on the same EEM instance. Multiple Containers can use the same application name. Leave the Proxy Host field empty unless you are integrating with a CA EEM instance in a cluster.
8. Review the information on the Install Summary page and click Install. The Container installs. Review the CATALYST_HOME*ContainerName*\logs\CA_Container_InstallLog.log file to troubleshoot installation errors.

Install CA Catalyst Connectors

Install CA Catalyst connectors on the same server where you installed the CA Catalyst Container. Consider the following items before installing CA Catalyst connectors:

- Each CA Catalyst connector comes with a product-specific *Connector Guide*. See this document for information about the connector functionality, installation, and configuration.
- Verify that the connector supports the CA Catalyst version that you are using before installation.
- You can install multiple connectors in the same CA Catalyst Container. Alternatively, you can install connectors on the same server into multiple CA Catalyst Containers with different Container names. Using dedicated CA Catalyst Containers on the same server lets each connector that is run under its own process.

Verify Container Installation and Connector Data

To verify a successful CA Catalyst Container and connector implementation, verify the following items:

- Container status
- Connector existence in the container
- Connector status and data in the consuming product, such as CA SOI

Follow these steps:

1. View the Windows services on the Container server.
2. Find the CA Catalyst Container *ContainerName* service, and verify that it is running. Start the service, if necessary. This service indicates that the Container exists and is active.
3. Enter the following URL in a Web browser:

   ```
   http://<containerserver>:7000/node/rest/
   ```

   An XML page opens. All connectors that are installed on the Container should appear as a module using the following nomenclature:

   ```
   CA:09998_<ContainerServer>_CatalystConnector
   ```

   The five-digit number is unique for each connector (09998 is the ID for the Sample connector).
4. (CA SOI integration only) Open the CA SOI Dashboard using the following URL:

   ```
   http://<SoiUIServer>:7070/sam/ui
   ```

   Log in using CA SOI credentials.

5. Click the Administration tab, and expand Connector Configuration. You should see an entry for the installed Container. If you defined a common prefix for use in a failover scenario, the Container name includes the prefix.
6. Expand the Container entry to see a list of connectors on the Container. Each connector entry provides the connector status and connection details.

CA Catalyst r3.4.4 Documentation

This section describes how to install and maintain the current version of CA Catalyst, which provides a lightweight container for hosting CA Catalyst connectors. This section is intended for product administrators who require the container to integrate connectors with their product.

Support for Windows 2019

Support for Windows 2019 is certified in this version. This CA Catalyst 3.4.4 is fully compatible with Windows 2019.

CA Catalyst Installation Planning

System Requirements and Software Support

The CA Catalyst Container supports installation on the following operating system:

- Microsoft Windows Server 2019 (64-bit) Standard with the latest service packs
- Microsoft Windows Server 2016 (64-bit) Standard with the latest service packs
- Microsoft Windows Server 2012 (64-bit) Standard or Datacenter with the latest service packs
- Microsoft Windows Server 2012 (64-bit) R2 Standard or Datacenter with the latest service packs

How to Install CA Catalyst

As an administrator, you install CA Catalyst to gain the benefit of a lightweight container that can collect and integrate information from key data sources into an integrated domain manager, such as CA SOI. You install CA Catalyst by installing Containers and then installing connectors in those Containers to integrate with key data sources.

Prerequisites:

Ensure that MQ Server is installed on a system and note the MQ server details.

Install the CA Catalyst Container (Typical)

Install the CA Catalyst Container on the server where you want to install connectors. Perform a recommended installation for a simple installation experience that uses the default values for most settings.

Follow these steps:

1. Run **catalyst_install.bat** from the Disk1 folder of the CA Catalyst installation image.
2. Accept the license agreement and click **Next**.
3. Select **Typical** on the Installation Type page, select **Use CA Service Operations Insight** if you plan to integrate with CA SOI and click **Next.**
4. Specify the installation directory, and proceed to the CA Service Operations Insight Configuration page, which appears only if you selected the Use CA Service Operations Insight checkbox on the Type of Installation page.
5. (CA SOI integration only) If you are integrating with CA SOI, enter valid information to connect to a CA SOI instance. Provide an active SA Manager hostname, MQ Server hostname, and valid CA SOI credentials. If you configure a CA SOI integration, the installer automatically uses the CA SOI administrator credentials as the CA Catalyst administrator credentials. If you do not configure CA SOI integration, the Catalyst Administrator page opens.
6. Enter a username and password for the CA Catalyst administrator user if necessary, and click Next.
7. Enter a valid CA EEM server hostname and credentials on the CA EEM Server Configuration page. The Application Name refers to the name under which CA EEM manages the CA Catalyst entitlement information. Use the default

unless another application is using this name on the same EEM instance. Multiple Containers can use the same application name. Leave the Proxy Host field empty unless you are integrating with a CA EEM instance in a cluster.

8. Review the information on the Install Summary page and click Install. The Container installs. Review the CATALYST_HOME*ContainerName*\logs\CA_Container_InstallLog.log file to troubleshoot installation errors.

Install CA Catalyst Connectors

Install CA Catalyst connectors on the same server where you installed the CA Catalyst Container. Consider the following items before installing CA Catalyst connectors:

- Each CA Catalyst connector comes with a product-specific *Connector Guide*. See this document for information about the connector functionality, installation, and configuration.
- Verify that the connector supports the CA Catalyst version that you are using before installation.
- You can install multiple connectors in the same CA Catalyst Container. Alternatively, you can install connectors on the same server into multiple CA Catalyst Containers with different Container names. Using dedicated CA Catalyst Containers on the same server lets each connector that is run under its own process.

Verify Container Installation and Connector Data

To verify a successful CA Catalyst Container and connector implementation, verify the following items:

- Container status
- Connector existence in the container
- Connector status and data in the consuming product, such as CA SOI

Follow these steps:

1. View the Windows services on the Container server.
2. Find the CA Catalyst Container *ContainerName* service, and verify that it is running. Start the service, if necessary. This service indicates that the Container exists and is active.
3. Enter the following URL in a Web browser:

   ```
   http://<containerserver>:7000/node/rest/
   ```

   An XML page opens. All connectors that are installed on the Container should appear as a module using the following nomenclature:

   ```
   CA:09998_<ContainerServer>_CatalystConnector
   ```

   The five-digit number is unique for each connector (09998 is the ID for the Sample connector).
4. (CA SOI integration only) Open the CA SOI Dashboard using the following URL:

   ```
   http://<SoiUIServer>:7070/sam/ui
   ```

   Log in using CA SOI credentials.
5. Click the Administration tab, and expand Connector Configuration. You should see an entry for the installed Container. If you defined a common prefix for use in a failover scenario, the Container name includes the prefix.
6. Expand the Container entry to see a list of connectors on the Container. Each connector entry provides the connector status and connection details.

Web Services

This sectioncovers the Representational State Transfer (REST) and WS-Management (WS-MAN) web services that are available with CA SOI. It describes the resources exposed and the available operations to perform against these web services.

See the following topics for details:

Intended Audience

This section is intended for product administrators, domain manager administrators, or integration developers who are interested in extending the CA SOI solution by integrating its data with other products through REST and WS-MAN web services.

It assumes experience with working with these web services and prior knowledge of the related concepts.

CA SOI REST Web Services

Contents

This section helps administrators and integration developers understand how CA SOI REST web services work, including the base URL, supported methods, endpoints, and common requirements.

Representational State Transfer (REST) is a client-server architectural style of building applications that leverages the fundamental properties of HTTP to manage objects accessible at a URL. REST architecture and applications are stateless, which means that no client context information is stored between requests. Each request contains all the information necessary to service the request. REST web services are lightweight, HTTP-based, easy to create and use, and have the desirable property of relating the classes of data to each other using hyperlinks. REST web services provide a simple yet powerful mechanism to interact with data. Using these web services, integration developers can configure the product and can make it communicate through the REST interface. They can use REST web services directly to send HTTP requests to the server for the resources they want to manipulate.

CA SOI lets you expose CA SOI data over REST web services. Because of the inherent standards in the REST architecture, CA SOI REST web services make the CA SOI data accessible to many different development environments. Several resources such as CA SOI user interfaces and third-party interfaces can then consume the exposed data. This ability helps integration developers extend the CA SOI solution by integrating its data with other products through REST web services. These interfaces provide an HTTP-based integration point to the CA SOI data, allowing read or write access. Using these web services, you can access the CA SOI data directly from a browser or can integrate it into your own applications. You can use these web services with any language that understands how to manage HTTP integration.

REST web services access resources by using a Uniform Resource Identifier (URI), a character string that identifies a name or resource on the Internet. An application using REST web services makes an HTTP request to a URI and parses the response. Such identification enables interaction with representations of the resource over a network. Each client-to-server request contains all the information necessary to understand the request, and does not use any stored context on the server.

CA SOI REST web services follow Hypertext As The Engine Of Application State (HATEOAS) principle. This principle implies that the resources that a request returns to the server contain the next state changes the client can navigate as links. The representations of the resources are interrelated using URLs, enabling you to move from one state to another.

This section helps you understand CA SOI REST web services as follows:

- Provides general concepts and common requirements about using CA SOI REST web services
- Provides information about how to retrieve, manipulate, and set data using various CA SOI REST web services
- Defines information that you can pass to the REST interface
- Provides information about how to send web services requests to perform specific tasks and verify that you get a meaningful response

> **NOTE**
> REST web services use the HTTP protocol for communication. Familiarity with both the HTTP protocol and the REST architecture is required. This section, therefore, assumes that you have experience working with REST web services and prior knowledge of related concepts (such as WADL and Hypertext As The Engine Of Application State).

Base CA SOI REST Web Services URL

The base URL for all CA SOI REST web services is as follows:

```
http://<server>:<port>/rest/
```

- **server**
  Specifies the server where the REST web service is located.
- **port**
  Specifies the port number where the REST web service is located. The default port numbers are 7403 (secure) and 7070 (non-secure).

Examples of base URLs are https://ServerABC:7403/rest/ (secure) and http://ServerXYZ:7070/rest/ (non-secure).

> **NOTE**
> By default, the non-secure interface is not allowed with the Basic authentication (user name and password), where the user name and password are sent as a plain text (*Base64* encoded). If you want to allow Basic authentication over the non-secure connection, you can configure the web.xml file. You can always use the non-secure interface with other types of authentications: CA EEM token and JSESSION. However, you can use the secure interface with all three types of authentications: Basic (user name and password), CA EEM token, and JSESSION. For more information about these authentication methods, see the REST Web Services Authentication.

Supported REST HTTP Methods

As an integration developer, you use REST HTTP methods with the REST web service URLs to manage information in your environment. REST HTTP methods help you achieve the following objectives:

- Access and modify associated resources
- Send an HTTP request to the server for the resource that you want to manipulate
- Control the attributes that you want to retrieve using HTTP headers

CA SOI REST web services support the following REST HTTP methods:

- **POST (Create)**
  Creates a resource. The web service can respond with data or status indicating a success or failure.
- **GET (Read)**
  Performs a query on a resource and retrieves data. The data that is returned from the web service is a representation of the requested resource.
- **PUT (Update)**
  Updates an existing resource.
- **DELETE (Delete)**
  Removes an existing resource.

Endpoints

REST web services URLs are specific endpoints. Endpoints are types of items that you use with appropriate HTTP request methods to return a list of results or create, update, or delete an item.

For example, the endpoint (used with the GET method) to get a list of links that let you find more information about a specific service (identified by a service ID) is as follows:

```
GET http://<server>:7070/rest/service/<serviceId>/entry
```

> **NOTE**
> For more information about specific endpoints, see Available CA SOI REST Web Services.

Web Application Description Language URL

You can obtain the complete Web Application Description Language (WADL) for CA SOI REST web services by adding *application.wadl?format=xml* to the base URL. The endpoint is used with the GET method as follows:

```
GET http://<server>:<port>/rest/application.wadl?format=xml
```

This WADL file outlines the available operations.

Common Requirements

The following requirements are applicable to all the requests:

- CA SOI REST web services require one of the authentication methods for each call.
- CA SOI REST web services produce the output in Atom format. Atom represents a document format that is based on XML, and that describes lists of associated resources. For Atom version, do one of the following tasks:
  - Send the *Accept* header *Accept: application/atom+xml*.
  - Specify the URL query parameter *format=atom*.
- The query parameter *format* takes precedence over the format defined in the *Accept* header.
- CA SOI REST web services follow Hypertext As The Engine Of Application State (HATEOAS) principle. This principle implies that the resources that a request returns to the server contain the next state changes the client can navigate as links.
- The typical output of a CA SOI REST web service request is the Atom feed, which can be *ordered* and *paged* to organize the output properly. For ordering and paging, you can use the following parameters:
  - **metric**
    Specifies the ordering domain. The values depend on the type of the returning object. The Ordering Metric section includes detailed information about what values you can use.
  - **desc**
    (If true) Returns the result in the descending order.
  - **size**
    Specifies the size of the page; that is, how many results to show on a single page. The default value is 25. The default value is used when you do not specify any value for the parameter, or you provide an invalid value (for example, a negative value).
  - **start**
    Specifies the page from where to start; that is, how many results to skip from the beginning. The default value is 0; therefore, *skip zero records* implies start from the beginning.

    > **NOTE**
    > Examples in this section demonstrate how CA SOI REST web services use create, read, update, and delete HTTP operations on CA SOI objects. Use these examples to understand how each REST operation interacts with CA SOI objects.

Resource Versioning

For backward compatibility reasons, CA SOI REST web services enable versioning of resources that is based on the content negotiation mechanism. This means that each change in the REST resources is versioned.

The desired resource version is specified by an accept header that is part of the request sent from the client.
For example, version 2 is invoked by adding an accept header with the value "application/vnd.ca.soi.api.v2+xml". For clients that do not support request header modification, a query parameter as part of the URL can be used to specify version, for example ?version=2.

Requests without an accept header will receive the default version (version 1), whose media type is application/xml.

In the current CA SOI version, resource versioning is supported for alert and CI definitions.

REST Web Services Authentication

Contents

This section helps administrators and integration developers understand the different authentication methods that the REST web services support and how to customize the acceptable authentication methods.

CA SOI REST web services support authentication in three ways: using a user name and password (Basic authentication), using a CA EEM artifact, or using a JSESSIONID. Typical communication with CA SOI REST web services is authenticated through the user name/password or CA EEM artifact in the first call, and then using the JSESSIONID for all other subsequent calls. The last call is the logout call that invalidates the session.

Therefore, for the first call, use one of the two methods: user name/password or CA EEM artifact. This first call returns the JSESSIONID, which the clients can then start using for making subsequent calls. The JSESSIONID authentication is the best performance authentication method, because it does not require the REST web service to verify the password or CA EEM artifact against CA EEM.

Additional information about these authentications is as follows:

- Using the user name and password (Basic authentication)
  The user name and password are sent in the form of the Basic HTTP authentication in the HTTP header. The password is encoded by using *Base64* encoding and can be easily decoded; therefore, this type of authentication is supported over a secure SSL channel (HTTPS protocol). For a typical CA SOI installation, it is port 7403, for example, https://server:7403/rest/.
  Note: Only CA EEM users have access to CA SOI REST API. Therefore the samuser user is not able to access the REST API.
- Using the one-time CA EEM artifact
  In this type of authentication, the CA EEM artifact can be obtained by calling the SafeSession.exportSession() method. You can use this type of authentication with both the protocolsHTTP and HTTPS.

  NOTE
  For more information about CA EEM sessions, see the CA EEM documentation.
- Using the JSESSIONID
  This type of authentication sends the JSESSIONID in the cookie header parameter. Both the previously mentioned authentication methods (user name/password and CA EEM artifact) set a cookie with the JSESSIONID value. The JSESSIONID can then be used in all subsequent calls through HTTP or HTTPS protocol.
  The session expires after 30 seconds of inactivity or after a call to the logout endpoint http://<server>:<port>/rest/logout.

  NOTE
  You can also configure authentication-related security filter parameters in the web.xml configuration file. For more information, see Configure the Authentication-related Security Filter Parameters.

Configure Authentication-Related Security Filter Parameters

The web.xml file includes authentication-related security filter parameters. You can configure the values of these parameters based on your unique IT environment requirements. This file is available in the folder <SOI_HOME>\SAMUI\webapps\rest\WEB-INF.

> **NOTE**
> <SOI_HOME> represents the location where CA SOI is installed; for example, C:\Program Files\CA\SOI.

Follow these steps:

1. Locate the web.xml file in the folder <SOI_HOME>\SAMUI\webapps\rest\WEB-INF.
2. Open the file using a text editor and search for the following section in the file:
   ```
   <filter-name>SecurityFilter</filter-name>
   ```
   This section includes three parameters that you can configure.
3. Specify the appropriate value in the *<param-value>...</param-value>* field of the following parameters:
   - **allowPlainCredentials**
     Specifies whether to allow Basic authentication over the non-SSL connection. Possible values are *true* and *false*. The default value is *false*, which implies that the Basic authentication is not allowed over the non-SSL connection.
     Default: false

     > **NOTE**
     > If the REST interface is accessed using the loopback address (127.0.0.1 or ::1), the Basic authentication is always allowed.
   - **cacheCredentials**
     Specifies whether the REST interface must cache the user credentials instead of verifying the password for each request. When you cache the user credentials, the password verification process becomes faster. Therefore, when you use the user name/password authentication method for all calls, the caching improves the response of the REST interface, because the CA EEM call is skipped.
     Possible values are *true* and *false*. The default value is *true*, which implies that the interface must cache the user credentials.
     Default: true

     > **NOTE**
     > User credentials are cached for 10 minutes. Therefore, if a user changes the password, the old password remains active in the REST interface until the cached record expires.
   - **sessionInactivateInterval**
     Specifies the time (in seconds) for which the REST interface must keep the session active. If no request is using the session during the specified time, the client must authenticate again.
     Default: 30
4. Review the new information and save the file.
   The changes are saved.

REST Web Services Ordering Metric

Contents

This section includes the ordering metric values for REST web services. The ordering metric specifies the ordering domain of the web service request output to organize the output appropriately.

> **NOTE**
> All metric values are not case-sensitive.

Service Ordering

The following table includes the ordering metric values for Service:

Metric Value	Description
NAME	Ordered by the service name
HEALTH	Ordered by the service health
RISK	Ordered by the service risk
AVAILABILITY	Ordered by the service availability (last 24 hours)
QUALITY	Ordered by the service quality
SLA	Ordered by the service SLA status
PRIORITY	Ordered by the service priority
ALERT_COUNT	Ordered by the down alert count
DOWN_ALERT_COUNT	Ordered by the down alert count
CRITICAL_ALERT_COUNT	Ordered by the critical alert count
MAJOR_ALERT_COUNT	Ordered by the major alert count
MINOR_ALERT_COUNT	Ordered by the minor alert count

Group Ordering

The following table includes the ordering metric values for Group:

Metric Value	Description
NAME	Ordered by the group name
PRIVILEGE_SET	Ordered by the privilege set, which implies that all administrator groups and user groups would be together

Customer Ordering

The following table includes the ordering metric values for Customer:

Metric Value	Description
NAME	Ordered by the customer name
HEALTH	Ordered by the customer health
RISK	Ordered by the customer risk
QUALITY	Ordered by the customer quality
PRIORITY	Ordered by the customer priority
DOWN_ALERT_COUNT	Ordered by the down alert count
CRITICAL_ALERT_COUNT	Ordered by the critical alert count
MAJOR_ALERT_COUNT	Ordered by the major alert count
MINOR_ALERT_COUNT	Ordered by the minor alert count

CI Ordering

The following table includes the ordering metric values for CI:

Metric Value	Description
NAME	Ordered by the CI name
HEALTH	Ordered by the CI health
USM_TYPE	Ordered by the USM type
IP_ADDRESS	Ordered by the IP address
OPERATIONAL_MODE	Ordered by the CI operational mode

Alert Queue Ordering

The following table includes the ordering metric values for Alert Queue:

Metric Value	Description
NAME	Ordered by the alert queue name
ALERT_COUNT	Ordered by the down alert count
DOWN_ALERT_COUNT	Ordered by the down alert count
CRITICAL_ALERT_COUNT	Ordered by the critical alert count
MAJOR_ALERT_COUNT	Ordered by the major alert count
MINOR_ALERT_COUNT	Ordered by the minor alert count

Alert Ordering

The following table includes the ordering metric values for Alert:

Metric Value	Description
SEVERITY	Ordered by the alert severity
TIME	Ordered by the occurrence time

User Ordering

The following table includes the ordering metric values for User:

Metric Value	Description
NAME	Ordered by the user name

Escalation Policy Ordering

The following table includes the ordering metric values for Escalation Policy:

Metric Value	Description
NAME	Ordered by the escalation policy name

Escalation Policy Action Ordering

The following table includes the ordering metric values for Escalation Policy Action:

Metric Value	Description
NAME	Ordered by the escalation policy action name

Configuration Ordering

The following table includes the ordering metric values for Configuration:

Metric Value	Description
NAME	Ordered by the configuration name

Configuration Node Ordering

The following table includes the ordering metric values for Configuration Node:

Metric Value	Description
NAME	Ordered by the configuration node name

Available CA SOI REST Web Services

Contents

This section lists the available CA SOI REST web services.

> **NOTE**
> The complete documentation for CA SOI REST web services is available at https://<ui-server>:<ssl port>/rest/docs/rest/.
>
> Go to
>
> https://:7403/rest/docs/rest/index.html#resources
>
> Select /Service from the path column

Alert Queue REST Web Services

Alert queues are user-defined alert groups. Alert queues let you group alerts as they come in based on specific criteria to monitor the status of your infrastructure more efficiently.

Using the Alert Queue REST web services, you can perform various operations on alert management queues in CA SOI:

- Create an alert queue
- Get a list of alert queues
- Get a list of alerts in an alert queue
- Get the alert queue definition\
- Get the escalation policy IDs for an alert queue
- Get hyperlink entries associated with an alert queue
- Get the status information for an alert queue
- Update an alert queue
- Delete an alert queue

The GET, PUT, POST, and DELETE HTTP methods are used to perform these tasks, as appropriate. For example, to delete an alert queue, the HTTP method DELETE is used.

NOTE
The complete documentation for the CA SOI Alert Queue REST web services is available at https://<ui-server>:<ssl port>/rest/docs/rest/.

Go to

https://:7403/rest/docs/rest/index.html#resources

Select /Service from the path column

Alert REST Web Services

Alerts are fault conditions that the integrated domain manager reports. Each alert is associated with a CI and contains properties such as severity, a summary of the condition, and when the condition occurred. Alerts are service impacting when they affect a CI that is part of a managed service. They are non-service impacting when they affect CIs that are not part of a managed service.

Using the Alert REST web services, you can perform various alert-related operations in CA SOI:

- Get a list of alerts
- Get hyperlink entries associated with an alert
- Get the alert definition
- Get the status information for an alert
- Get a list of escalation policy actions associated with an alert
- Perform an escalation policy action on a specific alert
- Get the alert queues for a specific alert
- Get alert root cause
- Get a list of services affected by an alert
- Update an alert
- Delete an alert

NOTE

If a delete REST operation fails, it may be due to the 'Respect Underlying MDR Clear Alert Setting' option. This option prevents you from clearing alerts in CA SOI that are not clearable in the source domain manager.

The GET, PUT, POST, and DELETE HTTP methods are used to perform these tasks, as appropriate. For example, to delete an alert, the HTTP method DELETE is used.

NOTE
The complete documentation for the CA SOI Alert REST web services is available at https://<ui-server>:<ssl port>/rest/docs/rest/.

Go to

https://:7403/rest/docs/rest/index.html#resources

Select /Service from the path column

CI REST Web Services

CIs in CA SOI represent IT elements managed by a domain manager. Each CI belongs to a type (defined in the USM schema) such as ComputerSystem, Database, Process, and Relationship. Connectors transform managed objects from domain managers to adhere to the USM schema and import the objects into CA Catalyst as CIs.

The CI REST web services let you perform the following operations:

- Get CI details
- Get hyperlink entries associated with a CI
- Get the status information for a CI
- Get the CI USM information
- Get a list of alerts impacting a specific CI
- Get a list of children for a specific CI
- Get a list of parents for a specific CI
- Get a list of services for a CI
- Update details of a CI, such as maintenance mode and user attributes

The GET and PUT HTTP methods are used to perform these tasks.

NOTE
The complete documentation for the CA SOI CI REST web services is available at https://<ui-server>:<ssl port>/rest/docs/rest/.

Go to

https://:7403/rest/docs/rest/index.html#resources

Select /Service from the path column

NOTE

Using the CA SOI CI REST web services, you get Managed CIs only and do not get any Unmanaged CIs.

Configuration REST Web Services

The CA SOI Configuration REST web services give you the flexibility to view and update integration configuration values as and when required using the programmable interface. You can override the static configuration values (such as CA EEM configuration, email configuration) specified at the time of installation. For example, CA SOI has a number of integrations with various external products. This integration information is stored in configuration files, which can require updates based on some changes in the deployment environment. You can use the Configuration REST web services to view and update these configurations.

You can perform the following tasks using the Configuration REST web services:

- Get a list of configuration nodes
- Get a list of configuration sections for a configuration node
- Get the configuration section definition
- Get hyperlink entries associated with a configuration section
- Update a configuration section

The GET and PUT HTTP methods are used to perform these tasks, as appropriate. For example, to update a configuration section, the HTTP method PUT is used.

NOTE
The complete documentation for the CA SOI Configuration REST web services is available at https://<ui-server>:<ssl port>/rest/docs/rest/.

Go to

https://:7403/rest/docs/rest/index.html#resources

Select /Service from the path column

Connector REST Web Services

Using the Connector REST web services, you can retrieve a complete list of connectors in the CA SOI installation, a list of additional hyperlink entries that are associated with a specific connector or detailed status information about a connector:

- Get a list of all connectors
- Get the CI class types available in a connector
- Get hyperlink entries associated with a connector
- Get the status of a connector

The GET method is used to perform these tasks.

> **NOTE**
> The complete documentation for the CA SOI Connector REST web services is available at https://<ui-server>:<ssl port>/rest/docs/rest/.
>
> Go to
>
> https://:7403/rest/docs/rest/index.html#resources
>
> Select /Service from the path column

Customer REST Web Services

A customer in CA SOI is any consumer of a managed service. You create customers and associate them with service models to see the impact of service degradation on the service consumer. Customer management provides an extra layer of insight into how end users dependent on provided services are affected when those services experience downtime or degraded performance.

Using the Customer REST web services, you can perform customer-related operations in CA SOI:

- Get a list of customers
- Get hyperlink entries associated with a customer
- Get the status information for a specific customer
- Get a list of services associated with a customer
- Get a list of alerts on all services associated with a customer
- Get a list of subcustomers
- Get information about the parent customer
- Get information about the selected customers
- Get information about services of a specific customer
- Get the customer definition
- Get a list of services associated with a customer (as XML)
- Create a top-level customer
- Create a subcustomer
- Set customer services
- Update a customer
- Delete a customer

The GET, PUT, POST, and DELETE HTTP methods are used to perform these tasks, as appropriate. For example, to delete a customer, the HTTP method DELETE is used.

> **NOTE**
> The complete documentation for the CA SOI Customer REST web services is available at: https://<ui-server>:<ssl port>/rest/docs/rest/.
>
> Go to

https://:7403/rest/docs/rest/index.html#resources

Select /Service from the path column

Email REST Web Services

The CA SOI Email REST web services give you the option to create and send emails using the REST programming interface. For example, you can create and send emails about alerts from your application using the Email web services.

You can perform the following task using the Email REST web services:

- Send an email

The POST HTTP method is used to perform this task.

NOTE
The complete documentation for the CA SOI Email REST web services is available at https://<ui-server>:<ssl port>/rest/docs/rest/.

Go to

https://:7403/rest/docs/rest/index.html#resources

Select /Service from the path column

Escalation Policy Action REST Web Services

Escalation policy action defines the action to perform when the policy criteria are met. For example, you can set an escalation policy action where when an alert matches escalation policy criteria, the alert triggers an action that sends an email to the technician responsible for the affected service.

Using the Escalation Policy Action REST web services, you can perform escalation policy action-related operations in CA SOI:

- Get a list of escalation policy actions
- Get hyperlink entries associated with an escalation policy action
- Get the escalation policy action definition

The GET HTTP method is used to perform these tasks.

NOTE
The complete documentation for the CA SOI Escalation Policy Action REST web services is available at https://<ui-server>:<ssl port>/rest/docs/rest/.

Go to

https://:7403/rest/docs/rest/index.html#resources

Select /Service from the path column

Escalation Policy REST Web Services

Escalation policy automates alert escalation according to user-defined criteria. When the policy criteria are met, a specified escalation action runs. Using the Escalation Policy REST web services, you can perform the following operations:

- Get a list of escalation policies
- Get hyperlink entries associated with an escalation policy
- Get the escalation policy definition
- Create an escalation policy
- Update an escalation policy
- Delete an escalation policy
- Get a list of assigned service IDs for an escalation policy
- Get a list of assigned alert queue IDs for an escalation policy
- Get a list of assigned escalation policy action IDs for an escalation policy
- Get a list of assigned schedule IDs for an escalation policy
- Set a list of escalation policy services
- Set a list of escalation policy alert queues
- Set a list of escalation policy actions
- Set a list of escalation policy schedules

The GET, PUT, POST, and DELETE HTTP methods are used to perform these tasks, as appropriate. For example, to delete an escalation policy, the HTTP method DELETE is used.

> **NOTE**
> The complete documentation for the CA SOI Escalation Policy REST web services is available at https://<ui-server>:<ssl port>/rest/docs/rest/.
>
> Go to
>
> https://:7403/rest/docs/rest/index.html#resources
>
> Select /Service from the path column

Group REST Web Services

A user group in CA SOI includes users having the same level of access privileges. You can access CA SOI user groups using the Group REST web services. The Group REST web services let you perform the following user group-related tasks:

- Get a list of groups
- Get hyperlink entries associated with a group
- Get the status information for a group
- Create a group
- Update a group
- Delete a group
- Get a list of users assigned to a group
- Assign users to a group
- Remove a user from a group
- Get the user definition in the assigned group
- Get the user group access status for all alert queues
- Set the user group access status for all alert queues
- Get a list of specific privileged alert queues for a group
- Set a list of specific privileged alert queues for a group
- Get the user group access status for all services
- Set the user group access status for all services
- Get a list of specific privileged services for a group
- Set a list of specific privileged services for a group
- Get the user group access status for all customers
- Set the user group access status for all customers
- Get a list of specific privileged customers for a group
- Set a list of specific privileged customers for a group
- Get a list of all the privileges for the Administrator role
- Get a list of all the privileges for the Operator role

The GET, PUT, POST, and DELETE HTTP methods are used to perform these tasks, as appropriate. For example, to delete a group, the HTTP method DELETE is used.

NOTE
The complete documentation for the CA SOI Group REST web services is available at https://<ui-server>:<ssl port>/rest/docs/rest/.

Go to

https://:7403/rest/docs/rest/index.html#resources

Select /Service from the path column

Meta REST Web Services

Using the Meta REST web services, you can retrieve administrator and operator group privileges:

- Get a definition of the Administrator group
- Get a definition of the Operator group

The GET method is used to perform these tasks.

NOTE
The complete documentation for the CA SOI Meta REST web services is available at https://<ui-server>:<ssl port>/rest/docs/rest/.

Go to

https://:7403/rest/docs/rest/index.html#resources

Select /Service from the path column

Repository CI Web Services

Repository CI web services let you retrieve and update information for managed and unmanaged CIs:

- Get repository CIs based on connector and class type
- Update details of a CI for maintenance mode and user attributes
- Update details of a CI for maintenance mode and user attributes asynchronously
- Get a hyperlink for a repository CI
- Get repository CI USM information

The GET and PUT methods are used to perform these tasks.

> **NOTE**
> The complete documentation for the CA SOI Repository CI REST web services is available at https://<ui-server>:<ssl port>/rest/docs/rest/.
>
> Go to
>
> https://:7403/rest/docs/rest/index.html#resources
>
> Select /Service from the path column

Schedule REST Web Services

Using Schedule REST web services, you can manage, create, or delete escalation schedules:

- Get a list of schedules
- Create a schedule
- Get schedule details
- Delete a schedule
- Get hyperlink entries associated with a schedule

The GET, POST, and DELETE methods are used to perform these tasks.

> **NOTE**
> The complete documentation for CA SOI Schedule REST web service is available at https://<ui-server>:<ssl port>/rest/docs/rest/.
>
> Go to
>
> https://:7403/rest/docs/rest/index.html#resources
>
> Select /Service from the path column

Service REST Web Services

Services in CA SOI represent discrete business functions that can contain CIs that multiple domain managers manage. A service typically consists of several CIs, which can be grouped to represent things like web server farms or clusters. Services can also contain subservices, which are subordinate service models. Service models typically represent high-level abstract entities; for example, a web-based retail transaction service, an application server service, and a printing service.

Using the Service REST web services, you can retrieve data about CA SOI services. The HTTP method GET is used to retrieve this information. You make an HTTP request, which uses the GET method, to a specific URL. The web service interprets the method in the URI as the action it must perform. The web service then retrieves the required data from the request and returns the output based on that data.

You can perform the following operations using the Service REST web services:

- Get a list of services
- Get service details
- Get hyperlink entries associated with a service
- Get the status information for a service
- Get the service USM information
- Get the service metric history information
- Get a list of service alerts
- Get a list of children for a service
- Get a list of parents for a service
- Get a list of parent services
- Get a list of escalation policies for a service
- Get customers impacted by a service
- Get a list of services based on service IDs
- Get a list of subservices
- Create a service
- Update a service
- Delete a service
- Get schedule IDs for a service
- Set service schedule

The GET, PUT, DELETE, and POST HTTP methods are used to perform these tasks, as appropriate. For example, to retrieve a list of services, the HTTP method GET is used.

> **NOTE**
> The complete documentation for the CA SOI Service REST web services is available at https://<ui-server>:<ssl port>/rest/docs/rest/.
>
> Go to
>
> https://:7403/rest/docs/rest/index.html#resources
>
> Select /Service from the path column

User REST Web Services

You can access CA SOI users using the User REST web services. This ability lets you manage CA SOI users; for example, you can create, list, and update users.

The User REST web services let you perform the following user-related tasks:

- Create a user
- Get a list of users
- Get hyperlink entries associated with a user
- Get the user definition
- Get a list of groups associated with a user
- Update a user
- Get the user preference definition
- Set the user preference definition

The GET, PUT, and POST HTTP methods are used to perform these tasks, as appropriate. For example, to update a user, the HTTP method PUT is used.

User creation and assignment works as follows:

1. The endpoint information in the Create a User section lets you create users in CA EEM. The created users remain in the *unassigned* state. You can list the unassigned users using the *unassigned* parameter while getting a list of users.
2. The created user is assigned to the CA SOI application and the group by using the endpoint information specified in the Assign Users to a Group section.

Therefore, this way, you cannot only create users and add them to a group, but also add existing users (for example, users that are created manually) to a group.

> **NOTE**
> The complete documentation for the CA SOI User REST web services is available at https://<ui-server>:<ssl port>/rest/docs/rest/.
>
> Go to
>
> https://:7403/rest/docs/rest/index.html#resources
>
> Select /Service from the path column

Calling CA SOI REST Web Services from Perl Scripts

Contents

The following section provides information about how to use REST web services to retrieve, post, and process CA SOI data, using a Perl script. Perl scripts are useful for the initial setup of the CA SOI platform because they allow you to automatically enter or process large amounts of data. You can use Perl scripts for making all types of calls, but this document highlights the main ways to embed REST web services calls in your scripts.

Prerequisites

You must have:

- Perl version 5.14.2.1 or higher installed.

  > **NOTE**
  > Verify that the Perl module XML::XPath is installed.
- Access to the Perl Script Example files in the following location: <SOI_HOME>\tools\examples\REST_Perl_Example\

Set Up Users in a CA SOI Environment

You can use the Perl script to process an input file and assign users to groups according to the file. This method of entering users into the system and assigning them to groups is useful when you want to process large amounts of data. If a user does not exist in the system, it creates the user. The format of the input file is comma-separated with user name first and group name last on each line.

Example

The following example shows three users being assigned to two user groups, using REST web services from Perl.

```
username1, groupA
username2, groupA
username3, groupB
```

A prerequisite is that all groups exist in the system.

NOTE
If a group does not exist in the system, the line containing that group is skipped. The script then continues with the next line.

Configure the Perl Script

The Perl script starts with an environment definition constant where you can set up connectivity to REST web services, user credentials, and so on.

Example

The following code snippet shows configuration constants from the beginning of the Perl script.

```
my $headless_server   = 'soi-r2-test06';    # hostname
my $headless_port     = '7070';             # port for plain HTTP access
my $headless_ssl_port = '7403';             # port for HTTPS access
my $headless_user     = 'samuser';          # admin user name
my $headless_password = 'P@ssword01';       # admin user password

my $password_for_created_users ='changeit';    # password for the newly created users
my $page_size = 10000;    # page size in which the data are fetched
                          # larger value more data fetched per one call.
```

Code Overview

PERL code contains detailed comments. This code overview highlights some key components of the PERL script. The following section provides examples that cover all aspects of calling REST web services, such as authentication, retrieving paged data, and parsing and posting data. You can use these examples to help create your own scripts to call REST web services.

You can find the Perl Script Example file in the following location:

<SOI_HOME>\tools\examples\REST_Perl_Example\

HTTP Access

To make HTTP and HTTPS calls, you can use the LWP::Simple module.

Example

The following example shows how to tell Perl to use the LWP::Simple module:

```
use LWP::Simple;  # LWP package to work with HTTP requests
```

The following classes are the most important of the package:

- **LWP::UserAgent**
  Behaves as a simple browser, where you can configure basic authentication, cookies, and so on.
- **LWP::Request**
  Represents the HTTP request.

  NOTE
  The LWP::Request is not required for the HTTP GET method, because the request comprises URL only. However, it is required for HTTP POST and HTTP PUT methods where you pass the URL and the payload.
- **LWP::Result**
  The class representing the HTTP response, which contains the result code, http header and body.

HTTP GET Call

Use the PERL script to perform an HTTP GET call.

Example

The following code snippet shows an HTTP GET call.

> **NOTE**
> The code snippet does not contain authentication to keep the example simple.

```
my $browser = LWP::UserAgent->new();                          # create browser
my $response = $browser->get('http://server:7070/rest/group');  # call HTTP get
if ($response->is_success)                                    # check HTTP status
  return $response->content;                                  # return the result
die "Error while calling http get";                           # error
```

SSL Communication and Basic Authentication Method

To use a secure communication channel, you can use SSL communication and a basic authentication method.

Example

The following code snippets shows SSL communication and a basic authentication method:

```
$ENV{PERL_LWP_SSL_VERIFY_HOSTNAME}=0;             # skip SSL certificate verification
my $browser = LWP::UserAgent->new();                          # create browser
$browser->credentials(                                        # BASIC auth
        'server:7403',
        'REST Authentication',                                # REALM
        'bob' => 'bobspasswd');
my $response = $browser->get('https://<server>:7403/rest/group'); # call HTTPS get
if ($response->is_success)                                    # check HTTP status
  return $response->content;                                  # return the result
die "Error while callings http get";
```

When you authenticate using user name and password, REST web services returns a cookie with the session ID.

Example

The following code snippet shows the code that reads the cookies and then how they are used.

regular expression to select the cookie

if ($response->header('SET-COOKIE') !~ /(JSESSIONID=\w+);/) {

die "cookie with jsession not found\n Aborting";

}

my $cookie = $1; # the first group is the value

The cookie can be used for authentication in all other calls by setting the cookie in the request headers.

```
$browser->default_header('Cookie' => $cookie); #set the sessionID as cookie
```

HTTP POST with LWP Request Creation

In addition to HTTP GET, the PERL code calls HTTP POST in REST web services.

> **NOTE**
> The Post call is similar to the HTTP GET, but it requires that you set the payload by creating an LWP::Request.

Example

The following code snippet shows the creation of an LWP::Request.

```
my $request = HTTP::Request->new(POST => 'http://server:7070/rest/user';
$request->header('Accept' => 'application/xml');
$request->content("<user>
                      <userName>bob</userName>
                      <password>changeit</password>
                   </user>");
$request->content_type("application/xml; charset=utf-8");
my $response = $browser->request($request);
if ($response->code == 201) {
    # location header has the URL to the newly created resource
    return $response->header('Location');
}
die "Error while calling HTTP_POST"
```

Parsing Response

REST web services responses are typically ATOM feeds which are XML. An easy way to get certain information from XML is to use an XPath expression to query information. To download the XPath module to your PERL environment, enter the following call from your shell command line:

cpan XML::XPath

The call downloads the required module.

You can start using the following module in your script:

```
use XML::XPath;   # XPath to parse XML (Atom) outputs
use XML::XPath::XMLParser;
```

Example

The following ATOM feed shows an example of a response from the REST interface.

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <title>Users</title>
  <link rel="next-page" type="application/atom+xml" href="http://server:7070/rest/user?
desc=false&#38;start=4&#38;size=4" title="next-page" />
  <author>
    <name>CA SOI</name>
  </author>
  <id>http://server:7070/rest/user?unassigned=true</id>
  <updated>2012-03-29T13:36:17Z</updated>
  <restApi:totalCountHint xmlns:restApi="http://ca.com/2011/soi/rest">75</restApi:totalCountHint>
  <entry>
    <title>zaiwenuser</title>
    <link rel="entry" type="application/atom+xml" href="http://server:7070/rest/user/zaiwenuser/entry"
 title="entry" />
    <id>http://server:7070/rest/user/zaiwenuser/entry</id>
    <published>2012-03-29T13:36:17Z</published>
  </entry>
  <entry>
    <title>zaiwen400</title>
```

```
    <link rel="entry" type="application/atom+xml" href="http://server:7070/rest/user/zaiwen400/entry"
 title="entry" />
    <id>http://server:7070/rest/user/zaiwen400/entry</id>
    <published>2012-03-29T13:36:17Z</published>
  </entry>
  <entry>
    <title>zaiwen300</title>
    <link rel="entry" type="application/atom+xml" href="http://server:7070/rest/user/zaiwen300/entry"
 title="entry" />
    <id>http://server:7070/rest/user/zaiwen300/entry</id>
    <published>2012-03-29T13:36:17Z</published>
  </entry>
  <entry>
    <title>zaiwen200</title>
    <link rel="entry" type="application/atom+xml" href="http://server:7070/rest/user/zaiwen200/entry"
 title="entry" />
    <id>http://server:7070/rest/user/zaiwen200/entry</id>
    <published>2012-03-29T13:36:17Z</published>
  </entry>
</feed>
```

If you want to return only one value in your XPath query, for example, to get a link to the next page, you can use XPath::findvalue() method which returns the one element.

Example

The following code snippet shows how to get the next page URL from the ATOM feed.

```
my $xp = XML::XPath->new($response->content);
my $next_page_url = $xp->findvalue('/feed/link[@rel=\'next-page\']/@href')->value();
```

In case your XPath returns list of values, use XPath::findnodes() method.

Example

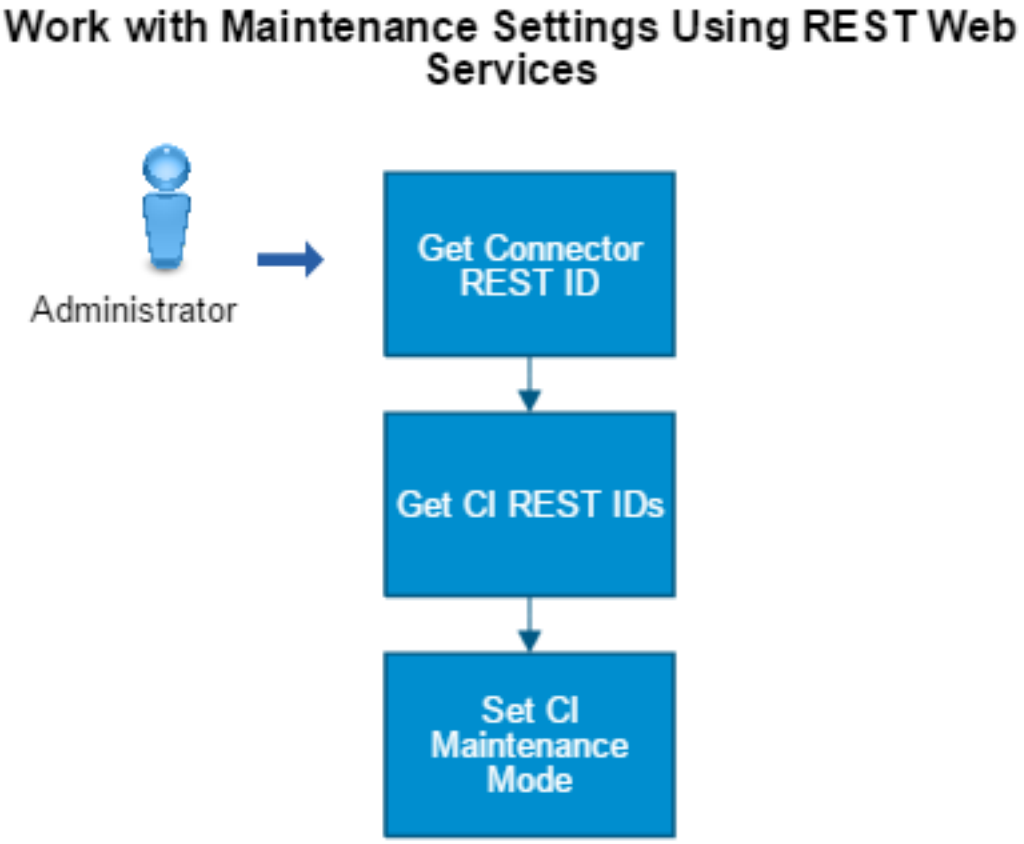
This code snippet shows how to get all titles from the entries in the ATOM feed.

```
my $xp = XML::XPath->new($response->content);
my @usernames = $xp->findnodes('/feed/entry/title');
foreach my $username (@usernames) {
   print $username->string_value;
}
```

Work with Maintenance Settings Using CA SOI REST Web Services

You can use CA SOI REST web services to work with the maintenance settings for managed and unmanaged CIs. This article provides a sample workflow for how you would retrieve and update maintenance settings and CI user attribute values using web services:

Figure 63: maintenancews



Use the following process to work with maintenance settings:

Get the Connector REST ID

Find the ID of the connector that manages the CIs you want to set in maintenance using the GET method of the Connector resource.

> **NOTE**
>
> If you are not sure which connector the CI is from, CA:00030, the reconciled CA Catalyst sheet, will contain all of the CIs.

```
?GET https://<UI_Server>:<HTTPS_PORT>/rest/connector?includeDetails=NONE
```

Available parameters:

includeDetails

A query that returns connector details. Set the parameter to STATUS to include connector status information.

Default: NONE

In this example, you set includeDetails to NONE, and you get a list of connectors in the result. Locate the entry section containing the title CA:00030@tenant0 and extract the ID of the connector to use to query and update a CI that is managed by the connector. The connector ID is in bold in the following example response:

```
. . .
 <entry>
 <title>CA:00030@tenant0</title>
 <link rel="entry" type="application/atom+xml" href="https://<host>:7403/rest/
connector/5277655813324805/entry" title="entry"/>
 <id>https://<host>:7403/rest/connector/5277655813324805/entry>
 <published>2015-01-30T18:15:31Z</published>
 </entry>
. . .
```

Find the CI REST ID

Use the connector ID that you retrieved to query CA SOI for a list of CI IDs that are managed by the connector. Use the Repository CI resource to query all managed and unmanaged CIs.

```
GET https://<UI_Server>:<HTTPS_PORT>/rest/repositoryCI?
connectorId=5277655813324805&classtype=Router
```

This query uses the retrieved connector ID and filters the returned CIs from the connector to include only those of the type Router.

> **NOTE**
>
> Available parameters not shown in the example include parameters for filtering, ordering, amount and size of results, detail inclusion, and whether to filter on maintenance status.

The result of the example query is a list of CIs that meet the criteria. From that list, you locate the CI that you want to edit and note its ID. The ID of the CI is in bold in the following example response:

```
<feed xmlns="http://www.w3.org/2005/Atom">
 <title>Repository CIs</title>
 <author>
 <name>CA SOI</name>
 </author>
 <id>https://<UI_Server>:<HTTPS_PORT>/rest/repositoryCI</id>
 <generator version="3.3.0.102.20141028"/>
 <updated>2015-01-30T19:19:20Z</updated>
 <restApi:totalCountHint xmlns:restApi="http://ca.com/2011/soi/rest">3>
 <entry>
 <title>CA SOI Service</title>
 <link rel="entry" type="application/atom+xml" href="https://<UI_Server>:<HTTPS_PORT>/
rest/repositoryCI/34B465A6B429482FA12C562073DD34B5/entry" title="entry"/>
 <id>https://<host>:7403/rest/repositoryCI/34B465A6B429482FA12C562073DD34B5/entry>
 <published>2015-01-30T19:19:20Z</published>
```

```
</entry> . . .
```

Set CI Maintenance Mode

Now that you have the ID for the CI, you can enter a PUT call on the CI to put it into maintenance.

```
PUT https://<UI_Server>:<HTTPS_PORT>/rest/repositoryCI/34B465A6B429482FA12C562073DD34B5
```

The following example PUT body also shows how you can set values for the CI user attributes.

```
?<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ci>
 <isInMaintenance>true</isInMaintenance>
 <CIuserAttribute1>New Attribute 1</CIuserAttribute1>
 <CIuserAttribute2>New Attribute 2</CIuserAttribute2>
 <CIuserAttribute3>New Attribute 3</CIuserAttribute3>
 <CIuserAttribute4>New Attribute 4</CIuserAttribute4>
 <CIuserAttribute5>New Attribute 5</CIuserAttribute5>
 <CIuserAttribute6>New Attribute 6</CIuserAttribute6>
 <CIuserAttribute7>New Attribute 7</CIuserAttribute7>
 <CIuserAttribute8>New Attribute 8</CIuserAttribute8>
 <CIuserAttribute9>New Attribute 9</CIuserAttribute9>
 <CIuserAttribute10>New Attribute 10</CIuserAttribute10>
</ci>
```

This request sets the CI's IsInMaintenance property to True and sets the CI user attributes to the specific values.

The response is the USM representation of the CI's reconciled projection sheet in CA SOI. The isInMaintenance and CI user attribute settings reflect the new values.

The request is validated as successful before a response is returned. Since this method is synchronous, we recommend using no more than 50 concurrent threads at a single time to have the best throughput.

> **TIP**
>
> To submit the query asynchronously, add async to the end of the query as follows:
>
> PUT https://<UI_Server>:<HTTPS_PORT>/rest/repositoryCI/34B465A6B429482FA12C562073DD34B5/async

View and Modify User Filters Using REST Web Services

You can use CA SOI REST web services to work with user preferences, or filters. This article provides a sample workflow for how you would retrieve and update user filters using web services:

1. Get the user filters for a specific user using a call similar to the following:
   ```
   ?GET https://<UI_Server>:<HTTPS_PORT>/rest/user/samuser/filters
   ```
   This call retrieves the current filter settings for samuser.
2. Update the user filters for a specific user using a call similar to the following:
   ```
   ? PUT https://<UI_Server>:<HTTPS_PORT>/rest/user/samuser/filters
   ```
   Paste the response from the GET call into the PUT body and make the necessary changes to the filters for samuser:
   ```
   ?<named-alarm-filters type="com.spectrum.app.alarm.client.preferences.NamedAlarmFiltersPreferences
    <pref>
   ```

```
    <complex-filter>
     <name>com.ca.cam.client.alarm.ServiceAlarms</name>
     <simple-filter>
      <show-model-families>3</show-model-families>
     </simple-filter>
    </complex-filter>
    <complex-filter>
     <name>com.ca.cam.client.alarm.ServiceAlarms</name>
     <simple-filter>
      <show-model-families>3</show-model-families>
     </simple-filter>
    </complex-filter>
   </pre>
  </named-alarm-filters>
```

This example body modifies the value of the settings that specify whether to show the model families for service and infrastructure alerts. You can update as many settings as needed.

WS-MAN Web Services

Contents

This section introduces the architecture of the CA SOI WS-MAN web services and the mechanisms used to interact with CA SOI data.

> **NOTE**
> The WS-MAN web services will be continued but will not be further enhanced. Instead, we recommend using the REST Web Services.

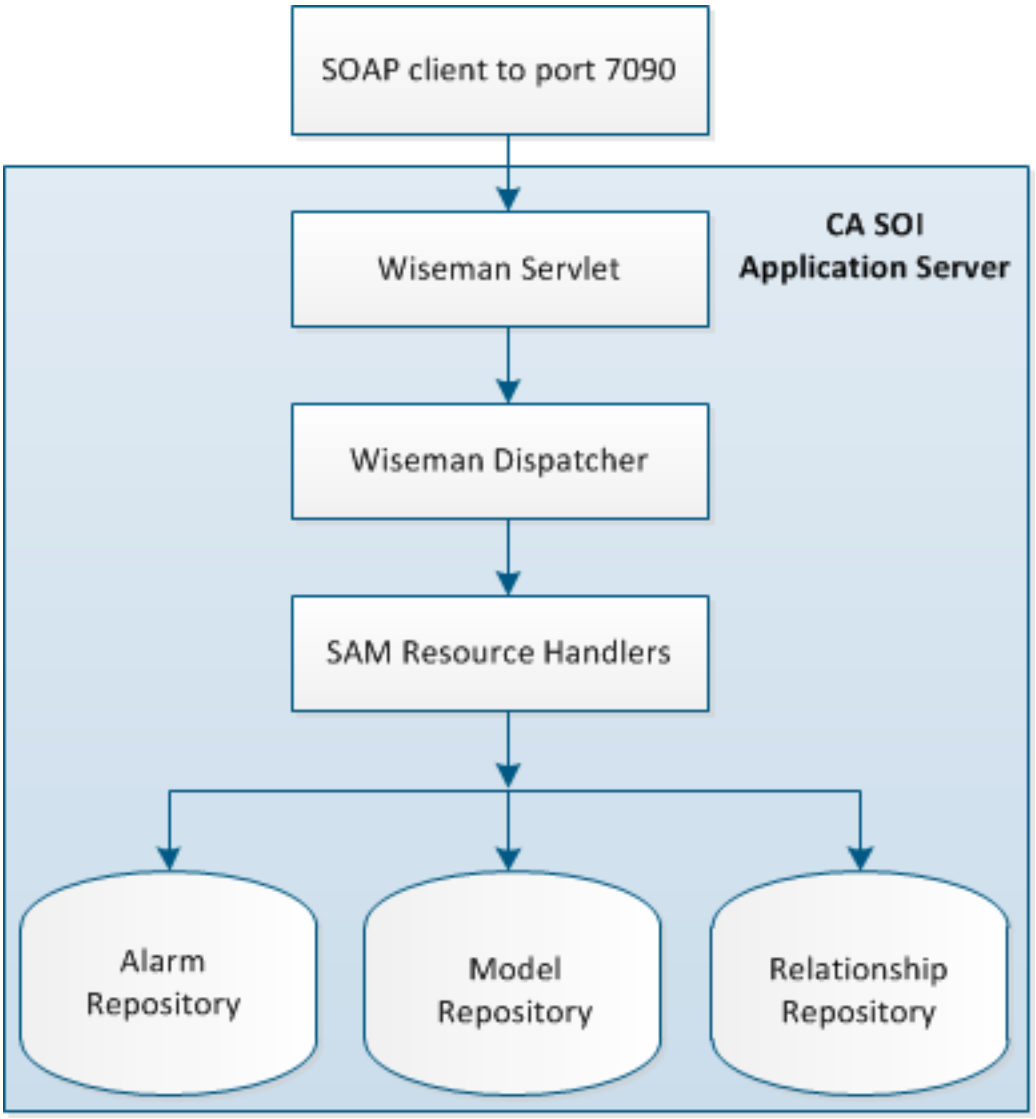
SOAP Support

For SOAP support information, see Software Support.

Web Services Architecture

CA SOI provides web services that comply with the WS-Management standard using a Wiseman implementation. The web services interface with the SA Manager to provide access to service, CI, alert, and other data. The web services use basic authentication to let clients access these resources through SOAP. Several Resource URIs are available through which the web services access the requested data.

The following diagram illustrates how the web service obtains data when it receives a request:



The WSManServlet intercepts client SOAP requests directed at the endpoint URL and dispatches them to the appropriate Resource Handler instance identified by the ResourceURI contained in the EPR element of the SOAP message. Wiseman automatically identifies the appropriate Resource Handler class in the following format: *Resource*_Handler.java.

Resources and Operations

The Endpoint Reference transport address for all CA SOI WS-Management web services is as follows:

```
http://<samanager>:<port>/sam/webservice
```

- **samanager**
  Defines the name of the server that contains the SA Manager.
- **port**
  Defines the SA Manager Tomcat port, which is 7090 by default.

WS-Management defines a resource addressing model based on the WS-Addressing standard. It uses the ReferenceParameter field in the WS-Addressing EndpointReference element to contain the following specific elements that identify the resource to act upon:

- **ResourceID**
  Defines the resource type or class.
- **SelectorSet**
  Defines the specific resource instance.

USM Schema Based Resources

The USM schema currently used as the CA SOI and CA Catalyst internal schema exposes CA SOI resources for web service operations using the following ResourceURI:

```
http://ns.ca.com/2009/07/usm-core/resource-class
```

- **resource-class**
  Defines the type of USM object, which can be Entity, BinaryRelationship, Alert, or Notification.

The following WSDL file outlines the available operations:

```
http://<samanager>:<port>/sam/webservice/wsdls/usm-core-200907.wsdl
```

Access the USM schema as follows:

```
http://<samanager>:<port>/sam/webservice/schemas/usm-core-200907.xsd
```

The initial USM schema web services implementation exposes the following USM resource types:

- **Entity**
  Refers to any CI in CA SOI, including services.
- **BinaryRelationship**
  Refers to the relationship between CIs. Note that this is different from the propagation type, which used the term relationship in previous releases.
- **Alert**
  Refers to CA SOI alerts.
- **Notification**
  Refers to the subscription to notifications for changes to Entities, Alerts, and BinaryRelationships.

  NOTE
  Notification resource is applicable only for CA SOI; it does not appear in CA Catalyst. By default, the USM schema does not contain any definition for the Notification resource. To ensure that it is available in the USM schema for CA SOI, special Java classes have been added explicitly for CA SOI.

Any of these resource types may require the following identifiers in the web service request as SelectorSets:

- **MdrProduct**
  Defines the connector data source. Each connector has a specific MdrProduct value formatted as a five-digit number prefixed by 'CA:'. For example, the MdrProduct value for resources created by web services is CA:09996. For a list of MdrProduct values, see Connector Identification Numbers.
- **MdrProdInstance**
  Defines the host name associated with the resource.
- **MdrElementID**
  Defines a value that uniquely identifies the resource.

You can retrieve these values from existing resources from the Operations Console or through an Enumerate web services operation.

USM 01-2009 Based Resources

A robust set of web services is available that uses a previous version of the USM schema (referred in the document as USM 01-2009) from the following ResourceURI:

```
http://ns.ca.com/2009/01/usm-data/resource-class
```

- **resource-class**
  Defines the type of USM object, which can be Queue, Customer, RelationshipPolicy, EscalationPolicy, or EscalationPolicyAction.

The following WSDL file outlines the available operations:

```
http://<samanager>:<port>/sam/webservice/wsdls/usm2.wsdl
```

Access the USM 01-2009 schema as follows:

```
http://<samanager>:<port>/sam/webservice/schemas/usm2.xsd
```

This implementation supports resources that are not accessible from the current USM schema web services, such as propagation policy, escalation policy, and escalation actions. Use this implementation to retrieve resources not available from the current USM schema web services, or for backward compatibility with previous web service implementations.

Any of these resource types may require one of the following identifiers in the web service request as the SelectorSet:

- **ASBOID**
  Defines the resource using the following properties:
  - **ASBOID.source**
    Defines the connector data source. Each data source has a unique value that you can obtain from the usm2.xsd file or an Enumerate operation.
  - **ASBOID.id**
    Defines the unique ID value for the resource, which you can obtain from an Enumerate operation.

  If you use ASBOID to uniquely identify the resource, include these properties separately.
- **USMID**
  Defines a value that uniquely identifies the resource by concatenating the ASBOID.source and ASBOID.id values.

WS-Transfer Operations

The web service resource handlers support all WS-Transfer operations as defined in the WS-Management specification:

- Get: http://schemas.xmlsoap.org/ws/2004/09/transfer/Get
- Put: http://schemas.xmlsoap.org/ws/2004/09/transfer/Put
- Create: http://schemas.xmlsoap.org/ws/2004/09/transfer/Create
- Delete: http://schemas.xmlsoap.org/ws/2004/09/transfer/Delete

The SelectorSet element is required to identify the appropriate resource instance. Each web service type (USM 01-2009, and USM) requires different SelectorSet values.

Fragment-Level WS-Transfer Operations

Fragment-level WS-Transfer is a WS-Management extension that lets you access a subset of resource properties. The web service resource handlers support fragment-level WS-Transfer for some operations where applicable using the default XPath dialect specified by the following URI:

http://www.w3.org/TR/1999/REC-xpath-19991116

The syntax is a list of property identifiers separated by "|" characters, such as the following:

```
<wsman:FragmentTransfer Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116"
 env:mustUnderstand="true">AlertID|SvcDeskTicket|SDTicketProp|AnnotationList
</wsman:FragmentTransfer>
```

WS-Enumeration Operations

The web service resource handlers support the following WS-Enumeration operations as defined in the WS-Management specification:

- Enumerate: http://schemas.xmlsoap.org/ws/2004/09/enumeration/Enumerate
- Pull: http://schemas.xmlsoap.org/ws/2004/09/enumeration/Pull
- Release: http://schemas.xmlsoap.org/ws/2004/09/enumeration/Release

Wiseman 1.0 does not implement the following operations that may be supported in the future:

- Renew: http://schemas.xmlsoap.org/ws/2004/09/enumeration/Renew
- GetStatus: http://schemas.xmlsoap.org/ws/2004/09/enumeration/GetStatus
- EnumerationEnd: http://schemas.xmlsoap.org/ws/2004/09/enumeration/EnumerationEnd

WS-Enumeration Filtering

WS-Enumeration defines XPath as the default filter dialect. XPath is designed to operate on XML elements, so all instances first must be retrieved, converted to XML, and then passed through the filter.

Configure an enumeration filter as follows:

```
final String xpathFilter = /alarm:alarm[alarm:ClassName='SA_Service']";
final Map<String, String> namespaces = new HashMap<String, String>(1);
namespaces.put("alarm","http://schemas.sam.ca.com/webservice/1/alarm.xsd");
```

This filter retrieves only those alerts that are associated with services.

WS-Eventing Operations

The web service resource handlers Entity, Alert, and BinaryRelationship support the following WS-Eventing operations as defined in the WS-Management specification (http://schemas.xmlsoap.org/ws/2004/08/eventing):

- Subscribe: http://schemas.xmlsoap.org/ws/2004/08/eventing/Subscribe
- Unsubscribe: http://schemas.xmlsoap.org/ws/2004/08/eventing/Unsubscribe

This release does not support the following operations:

- GetStatus: http://schemas.xmlsoap.org/ws/2004/08/eventing/GetStatus
- SubscriptionEnd: http://schemas.xmlsoap.org/ws/2004/08/eventing/SubscriptionEnd
- Renew: http://schemas.xmlsoap.org/ws/2004/08/eventing/Renew

Available Web Services

The available web services and their resource URLs are as follows:

- USM Entity: http://ns.ca.com/2009/07/usm-core/Entity
- USM BinaryRelationship: http://ns.ca.com/2009/07/usm-core/BinaryRelationship
- Alert: http://ns.ca.com/2009/07/usm-core/Alert
- Propagation Policy: http://ns.ca.com/2009/01/usm-data/RelationshipPolicy
- Escalation Policy: http://ns.ca.com/2009/01/usm-data/EscalationPolicy
- Escalation Action: http://ns.ca.com/2009/01/usm-data/EscalationpolicyAction
- Queue: http://ns.ca.com/2009/01/usm-data/Queue
- Notification: http://ns.ca.com/2009/07/usm-core/Notification
- Customer: http://ns.ca.com/2009/01/usm-data/Customer

Web Clients

The WS-MAN web services use a standard WS-MAN API and work with any applicable web client.

USM Entity Web Services

Contents

This section provides information about the operations performed in USM entity web services.

Entity Web Services Overview

Entity web services use the USM schema to perform operations on USM entities, which include CIs, services, and alerts. Use the following endpoint URI when invoking the entity web services resource:

```
http://ns.ca.com/2009/07/usm-core/Entity
```

The following WSDL file outlines the available operations:

```
http://<samanager>:<port>/sam/webservice/wsdls/usm-core-200907.wsdl
```

Access the USM schema as follows:

```
http://<samanager>:<port>/sam/webservice/schemas/usm-core-200907.xsd
```

Using the entity web services requires basic knowledge of the USM schema and its properties. In addition to the schema itself, HTML documentation is available. For more information, see How to Access the USM Schema Documentation.

Get an Entity

Use the Get request to retrieve a specific entity. The following selectors are required as part of the request to identify a unique instance of a CA SOI CI:

> **NOTE**
> This request also retrieves the KPI properties (Risk, Quality, and Health) of an entity.

- **MdrProduct**
  Defines the connector data source. Each connector has a specific MdrProduct value formatted as a five-digit number prefixed by 'CA:'. For example, the MdrProduct value for resources created by web services is CA:09996. For a list of MdrProduct values, see Connector Identification Numbers.
- **MdrProdInstance**
  Defines the host name associated with the resource.
- **MdrElementID**
  Defines a value that uniquely identifies the resource.

To get an entity, use the following properties in the request:

Operation: Get

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/07/usm-core/Entity

Selector: MdrProduct

Selector: MdrProdInstance

Selector: MdrElementID

The EntityHandlerImpl.Get() method returns an entity of the type of USM resource that represents the CA SOI CI in USM terms, including the CI type and its properties.

Get a List of Entities

To retrieve a list of active CIs, the web services use a combination of WS-Management Enumeration and Pull operations. You can filter the returned list using the WS-Management Filter element to pass a valid XPath expression to limit the number and type of CIs returned.

Use the className selector to filter the collection by USM type. For example, if you pass in a selector of className=Service, the collection is limited to entities of the Service type.

To get a list of entities, use the following properties in the request:

Operation: Enumerate & Pull

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/07/usm-core/Entity

Selector: className

The EntityIteratorImpl() method creates a collection of all the active CIs in CA SOI of the USM type defined in the className selector, and the Pull operation retrieves CIs in batches as defined by the MaxElements tag.

Create an Entity

Use the Create operation to create a CI in CA SOI. You define the CI type and USM property values for the new CI in the body of the request.

> **NOTE**
> For information about the required properties to include for an entity, see the USM schema documentation.

To create a CI, use the following properties in the request:

Operation: Create

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/07/usm-core/Entity

Selector: null

The EntityHandlerImpl.Create() method extracts all defined property values from the body of the SOAP message and passes a map of the property names and values to the Connector Manager, which is the same interface that all connectors use to create CIs.

> **NOTE**
> Refer to the USM schema for a list of USM properties and their appropriate values. For information see How to Access the USM Schema Documentation.

The following example SOAP message to create a ComputerSystem CI:

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns:usm="http://ns.ca.com/2009/07/usm-core"
xmlns:soap="http://schemas.xmlsoap.org/soap/evelope"
xmlns:mex="http://schemas.xmlsoap.org/ws/2004/09/mex"
xmlns:wsmeta="http://schemas.dmtf.org/wbem/wsman/1/wsman/version1.0.0.a/default-addressing-model.xsd"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding"
xmlns:wxf="http://schemas.xmlsoap.org/ws/2004/09/transfer"
xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:encodingStyle="http://schemas.xmlsoap.org/soap/encoding"
xmlns:wsen="http://schemas.xmlsoap.org/ws/2004/09/enumeration"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

```
xmlns:mdo="http://schemas.wiseman.dev.java.net/metadata/messagetypes"
xmlns:wse="http://schemas.xmlsoap.org/ws/2004/08/eventing">
<env:Header>
   <wsa:To>http://<sam manager>:<port>/sam/webservice</wsa:To>
   <wsman:ResourceURI>http://ns.ca.com/2009/07/usm-core/Entity</wsman:ResourceURI>
   <wsa:Action>http://schemas.xmlsoap.org/ws/2004/09/transfer/Create</wsa:Action>
   <wsa:ReplyTo>
    <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
   </wsa:ReplyTo>
   <wsa:MessageID>uuid:079e3c5a-7c5e-41a0-b5a8-992238aa55e3</wsa:MessageID>
</env:Header>
   <env:Body>
   <usm:ComputerSystem>
    <siloName></siloName>
    <usm:MdrProduct></usm:MdrProduct>
    <usm:MdrProdInstance></usm:MdrProdInstance>
    <usm:MdrElementID>MyComputerA</usm:MdrElementID>
    <usm:Label>MyComputerA</usm:Label>
    <usm:DeviceSysName>MyComputerA</usm:DeviceSysName>
    <usm:SysName>MyComputerA</usm:SysName>
     </usm:ComputerSystem>
   </env:Body>
</env:Envelope>
```

Update an Entity

Use the Put operation to update the writable properties of a CI. Perform the update by passing in all of the USM properties and their new values in the body of the request.

To update an entity, use the following properties in the request:

Operation: Put

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/07/usm-core/Entity

Selector: null

The EntityHandlerImpl.Put method extracts the entity property name value pairs and passes them into a CI Update request to the Connector Manager.

> **NOTE**
> Refer to the USM schema for a list of USM properties and their appropriate values. For information see How to Access the USM Schema Documentation.

Delete an Entity

Use the Delete operation to delete a CI. If the CI exists and has active alerts, the operation clears the associated alerts before deleting the CI. Pass the USM properties of the CI to delete in the body of the request.

> **NOTE**
> For information about the required properties to include for an entity, see the USM schema documentation.

To delete an entity, use the following properties in the request:

Operation: Delete

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/07/usm-core/Entity

Selector: Null

The EntityHandlerImpl.Delete() method verifies that the CI exists, clears any active associated alerts, and deletes the CI. The CI is deleted only if the connector is a trusted source, otherwise the CI is marked as 'Unmanaged.' A trusted source connector is the trusted source of record for a CI. Imported information uses the source connector as a trusted source unless you change the trusted source in the Operations Console.

The following example SOAP message to delete a ComputerSystem CI:

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns:usm="http://ns.ca.com/2009/07/usm-core"
xmlns:soap="http://schemas.xmlsoap.org/soap/evelope"
xmlns:mex="http://schemas.xmlsoap.org/ws/2004/09/mex"
xmlns:wsmeta="http://schemas.dmtf.org/wbem/wsman/1/wsman/version1.0.0.a/default-addressing-model.xsd"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding"
xmlns:wxf="http://schemas.xmlsoap.org/ws/2004/09/transfer"
xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:encodingStyle="http://schemas.xmlsoap.org/soap/encoding"
xmlns:wsen="http://schemas.xmlsoap.org/ws/2004/09/enumeration"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:mdo="http://schemas.wiseman.dev.java.net/metadata/messagetypes"
xmlns:wse="http://schemas.xmlsoap.org/ws/2004/08/eventing">
<env:Header>
   <wsa:To>http://<sam manager>:<port>/sam/webservice</wsa:To>
   <wsman:ResourceURI>http://ns.ca.com/2009/07/usm-core/Entity</wsman:ResourceURI>
   <wsa:Action>http://schemas.xmlsoap.org/ws/2004/09/transfer/Delete</wsa:Action>
   <wsa:ReplyTo>
    <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
   </wsa:ReplyTo>
   <wsa:MessageID>uuid:079e3c5a-7c5e-41a0-b5a8-992238aa55e3</wsa:MessageID>
</env:Header>
   <env:Body>
   <usm:ComputerSystem>
    <siloName></siloName>
    <usm:MdrProduct></usm:MdrProduct>
    <usm:MdrProdInstance></usm:MdrProdInstance>
    <usm:MdrElementID>MyComputerA</usm:MdrElementID>
    <usm:Label>MyComputerA</usm:Label>
    <usm:DeviceSysName>MyComputerA</usm:DeviceSysName>
    <usm:SysName>MyComputerA</usm:SysName>
     </usm:ComputerSystem>
   </env:Body>
</env:Envelope>
```

Entity Web Services Examples

The following example shows the SOAP messages of many of the available entity web services.

Example: Update a ComputerSystem CI

The following example SOAP message is a Put request to update a set of properties in a specific ComputerSystem CI:

```
<env:Envelope>
```

```
<env:Header>
   <wsa:ReplyTo xmlns:usm="http://ns.ca.com/2009/07/usm-core">
   <wsa:Address env:mustUnderstand="true">
    http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
   </wsa:Address>
   </wsa:ReplyTo>
   <wsa:MessageID xmlns:usm="http://ns.ca.com/2009/07/usm-core"
   env:mustUnderstand="true">
    uuid:4add8974-25a0-4fc4-84f6-c7f88158f8f7
   </wsa:MessageID>
   <wsa:To xmlns:usm="http://ns.ca.com/2009/07/usm-core" env:mustUnderstand="true">
    http://localhost:7090/sam/webservice
   </wsa:To>
   <wsman:ResourceURI xmlns:usm="http://ns.ca.com/2009/07/usm-core">
    http://ns.ca.com/2009/07/usm-core/Entity
   </wsman:ResourceURI>
   <wsman:OperationTimeout xmlns:usm="http://ns.ca.com/2009/07/usm-core">
    PT30.000S
   </wsman:OperationTimeout>
   <wsa:Action xmlns:usm="http://ns.ca.com/2009/07/usm-core"
   env:mustUnderstand="true">
    http://schemas.xmlsoap.org/ws/2004/09/transfer/Put
   </wsa:Action>
</env:Header>
<env:Body>
   <usm:ComputerSystem >
        <ssa_connector_name>server1.ca.com</ssa_connector_name>
        <ssa_silo_name>WebServiceForSSA_server1@server1.ca.com</ssa_silo_name>
        <usm:MdrProduct>CA:09996</usm:MdrProduct>
        <usm:MdrProdInstance>server1.ca.com</usm:MdrProdInstance>
        <usm:MdrElementID>ComputerSystem:server1</usm:MdrElementID>
        <usm:InstanceName>CA:server1</usm:InstanceName>
        <usm:Label>USM-WSServer3</usm:Label>
        <usm:Description>New description</usm:Description>
        <usm:AdministrativeStatus>Managed</usm:AdministrativeStatus>
        <usm:Vendor>Dell</usm:Vendor>
        <usm:PrimaryDnsName>server1.ca.com</usm:PrimaryDnsName>
        <usm:PrimaryIPV4Address>111.11.11.11</usm:PrimaryIPV4Address>
        <usm:ComputerName>server1</usm:ComputerName>
        <usm:MemoryInGB>512</usm:MemoryInGB>
   </usm:ComputerSystem>
</env:Body>
</env:Envelope>
```

The web service updates the properties listed in bold in the body of the request. The SOAP response to this request is as follows:

```
<env:Envelope>
<env:Header>
   <wsa:Action env:mustUnderstand="true"
   xmlns:ns14="http://ns.ca.com/2009/07/usm-core">
    http://schemas.xmlsoap.org/ws/2004/09/transfer/PutResponse
   </wsa:Action>
   <wsa:MessageID env:mustUnderstand="true"
```

```
    xmlns:ns14="http://ns.ca.com/2009/07/usm-core">
     uuid:f1ccc796-28a6-4afc-9766-d1438fe7d011
    </wsa:MessageID>
    <wsa:RelatesTo xmlns:ns14="http://ns.ca.com/2009/07/usm-core">
     uuid:4add8974-25a0-4fc4-84f6-c7f88158f8f7
    </wsa:RelatesTo>
    <wsa:To env:mustUnderstand="true" xmlns:ns14="http://ns.ca.com/2009/07/usm-core">
     http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
    </wsa:To>
  </env:Header>
  <env:Body/>
  </env:Envelope>
```

Subscribe to Notifications for Entity Changes

To subscribe to notifications for changes to Entity, use the Notification web services. The Entity resource supports the WS-Eventing functionality that enables a web client to subscribe to notification events when an entity is created, deleted, or updated.

> **NOTE**
> For more information, see Notification Web Services.

USM Binary Relationship Web Services

Contents

This section provides information about the operations performed in USM Binary Relationship web services.

BinaryRelationship Web Services Overview

BinaryRelationship web services use the USM schema to perform operations on USM BinaryRelationships, which define the relationships between CIs in CA SOI.

USM BinaryRelationships (referred to as relationships) are not the same as the entities known as relationships in previous releases of CA SOI. Propagation types define how the impact is propagated between related CIs (which was the previous role of relationships), and relationships are USM entities that show how CIs are linked. Every relationship in CA SOI has a corresponding propagation type. The USM Semantic property indicates the propagation type for each relationship. Therefore, when you interact with relationships, you can view, create, or edit their corresponding propagation types at the same time.

> **NOTE**
> For a list of the available relationships and propagation types, see Create Propagation Policy and Assign Types.

Use the following endpoint URI when invoking the relationship web services resource:

```
http://ns.ca.com/2009/07/usm-core/BinaryRelationship
```

The following WSDL file outlines the available operations:

```
http://<samanager>:<port>/sam/webservice/wsdls/usm-core-200907.wsdl
```

Access the USM schema as follows:

```
http://<samanager>:<port>/sam/webservice/schemas/usm-core-200907.xsd
```

Using the relationship web services requires basic knowledge of the USM schema and its properties. In addition to the schema itself, HTML documentation is available. For more information see How to Access the USM Schema Documentation.

Get a Relationship

Use the Get request to get a specific relationship. Each relationship belongs to one of the USM BinaryRelationship types. The following selectors are required as part of the USM definition to identify a unique instance of a relationship:

- **MdrProduct**
  Defines the connector data source. Each connector has a specific MdrProduct value formatted as a five-digit number prefixed by 'CA:'. For example, the MdrProduct value for resources created by web services is CA:09996. For a list of MdrProduct values, see Connector Identification Numbers.
- **MdrProdInstance**
  Defines the host name associated with the resource.
- **MdrElementID**
  Defines a value that uniquely identifies the resource.

To get a relationship, use the following properties in the request:

Operation: Get

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/07/usm-core/BinaryRelationship

Selector: MdrProduct

Selector: MdrProdInstance

Selector: MdrElementID

The BinaryRelationshipImpl.Get() method returns an entity of the type of USM BinaryRelationship that represents the CA SOI relationship in USM terms.

Get a List of Relationships

To retrieve a list of relationships for a given CA SOI service, the web services use a combination of WS-Management Enumeration and Pull operations. You can filter the returned list using the WS-Management Filter element to pass a valid XPath Expression to limit the number and type of relationships returned.

Use the serviceName selector to filter the collection by service.

To get a list of relationships, use the following properties in the request:

Operation: Enumerate & Pull

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/07/usm-core/BinaryRelationship

Selector: serviceName

The BinaryRelationshipIteratorImpl() method creates a collection of all the relationships for the service defined in the selector. The Pull operation retrieves relationships in batches as defined by the MaxElements tag.

Create a Relationship

Use the Create operation to create a relationship between two CIs in CA SOI. You define the type and USM property values for the new relationship in the body of the request. You should include the following information:

- Unique identifiers (MdrProduct, MdrProdInstance, MdrElementID)
- Unique identifiers for the source CI
- Unique identifiers for the target CI
- Unique identifiers for the service in which to include the relationship
- Semantic value to define the propagation type

NOTE
For information about the required properties to include for a relationship, see the USM schema documentation. For more information about property names and formatting, see USM Binary Relationship Web Services Examples.

To create a relationship, use the following properties in the request:

Operation: Create

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/07/usm-core/BinaryRelationship

Selector: null

The BinaryRelationshipHandlerImpl.Create() method extracts all defined properties from the body of the SOAP message and passes a map of the property names and values to the Connector Manager, which is the same interface that all connectors use to create relationships.

NOTE
Refer to the USM schema for a list of USM properties and their appropriate values. For information about how to access the USM schema documentation, see How to Access the USM Schema Documentation.

The following example SOAP message to create a relationship:

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns:usm="http://ns.ca.com/2009/07/usm-core"
xmlns:soap="http://schemas.xmlsoap.org/soap/evelope"
xmlns:mex="http://schemas.xmlsoap.org/ws/2004/09/mex"
xmlns:wsmeta="http://schemas.dmtf.org/wbem/wsman/1/wsman/version1.0.0.a/default-addressing-model.xsd"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding"
xmlns:wxf="http://schemas.xmlsoap.org/ws/2004/09/transfer"
xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:encodingStyle="http://schemas.xmlsoap.org/soap/encoding"
xmlns:wsen="http://schemas.xmlsoap.org/ws/2004/09/enumeration"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:mdo="http://schemas.wiseman.dev.java.net/metadata/messagetypes"
xmlns:wse="http://schemas.xmlsoap.org/ws/2004/08/eventing">
<env:Header>
   <wsa:To>http://<sam manager>:<port>/sam/webservice</wsa:To>
   <wsman:ResourceURI xmlns:usm="http://ns.ca.com/2009/07/usm-core">http://ns.ca.com/2009/07/usm-core/
BinaryRelationship</wsman:ResourceURI>
   <wsa:Action>http://schemas.xmlsoap.org/ws/2004/09/transfer/Create</wsa:Action>
   <wsa:ReplyTo>
    <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
   </wsa:ReplyTo>
   <wsa:MessageID>uuid:079e3c5a-7c5e-41a0-b5a8-992238aa55e3</wsa:MessageID>
</env:Header>
   <env:Body>
   <usm:BinaryRelationship>
    <siloName></siloName>
    <usm:MdrProduct></usm:MdrProduct>
    <usm:MdrElementID>My_relationA</usm:MdrElementID>
    <usm:MdrProdInstance></usm:MdrProdInstance>
    <usm:SourceMdrProduct>CA:00996</usm:SourceMdrProduct>
    <usm:SourceMdrProdInstance></usm:SourceMdrProdInstance>
```

```
    <usm:SourceMdrElementID>MyServiceA</usm:SourceMdrElementID>
    <usm:TargetMdrProduct>CA:00996</usm:TargetMdrProduct>
    <usm:TargetMdrProdInstance></usm:TargetMdrProdInstance>
    <usm:TargetMdrElementID>MyComputerA</usm:TargetMdrElementID>
    <usm:ScopeMdrProduct>CA:00996</usm:ScopeMdrProduct>
    <usm:ScopeMdrProdInstance></usm:ScopeMdrProdInstance>
    <usm:ScopeMdrElementID>MyServiceA</usm:ScopeMdrElementID>
    <usm:Semantic>Depends On</usm:Semantic>
    <usm:Significance>5</usm:Significance>
</usm:BinaryRelationship>
   </env:Body>
</env:Envelope>
```

Update a Relationship

Use the Put operation to update the writable properties of a relationship. Perform the update by passing in all of the USM BinaryRelationship properties and their new values in the body of the request. You should include the following information:

- Unique identifiers (MdrProduct, MdrProdInstance, MdrElementID)
- Unique identifiers for the source CI
- Unique identifiers for the target CI
- Unique identifiers for the service in which to include the relationship
- Semantic value to define the propagation type

> **NOTE**
> For more information, see USM Binary Relationship Web Services Examples.

To update a relationship, use the following properties in the request:

Operation: Put

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/07/usm-core/BinaryRelationship

Selector: null

The BinaryRelationshipHandlerImpl.Put method extracts the relationship property name value pairs and passes them into a Relationship Update request to the Connector Manager.

> **NOTE**
> Refer to the USM schema for a list of USM properties and their appropriate values. For information about how to access the USM schema documentation, see How to Access the USM Schema Documentation.

Delete a Relationship

Use the Delete operation to delete a relationship. Pass the USM properties of the relationship to delete in the body of the request.

> **NOTE**
> For information about the required properties to include for a relationship, see the USM schema documentation.

To delete a relationship, use the following properties in the request:

Operation: Delete

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/07/usm-core/BinaryRelationship

Selector: Null

The BinaryRelationshipHandlerImpl.Delete() method passes the delete request to the Connector Manager to process.

The following example SOAP message to delete a relationship:

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns:usm="http://ns.ca.com/2009/07/usm-core"
xmlns:soap="http://schemas.xmlsoap.org/soap/evelope"
xmlns:mex="http://schemas.xmlsoap.org/ws/2004/09/mex"
xmlns:wsmeta="http://schemas.dmtf.org/wbem/wsman/1/wsman/version1.0.0.a/default-addressing-model.xsd"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding"
xmlns:wxf="http://schemas.xmlsoap.org/ws/2004/09/transfer"
xmlns:wsman="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:encodingStyle="http://schemas.xmlsoap.org/soap/encoding"
xmlns:wsen="http://schemas.xmlsoap.org/ws/2004/09/enumeration"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:mdo="http://schemas.wiseman.dev.java.net/metadata/messagetypes"
xmlns:wse="http://schemas.xmlsoap.org/ws/2004/08/eventing">
<env:Header>
   <wsa:To>http://<sam manager>:<port>/sam/webservice</wsa:To>
   <wsman:ResourceURI xmlns:usm="http://ns.ca.com/2009/07/usm-core">http://ns.ca.com/2009/07/usm-core/
BinaryRelationship</wsman:ResourceURI>
   <wsa:Action>http://schemas.xmlsoap.org/ws/2004/09/transfer/Delete</wsa:Action>
   <wsa:ReplyTo>
    <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
   </wsa:ReplyTo>
   <wsa:MessageID>uuid:079e3c5a-7c5e-41a0-b5a8-992238aa55e3</wsa:MessageID>
</env:Header>
   <env:Body>
   <usm:BinaryRelationship>
    <siloName></siloName>
    <usm:MdrProduct>CA:00996</usm:MdrProduct>
    <usm:MdrElementID>My_relationA</usm:MdrElementID>
    <usm:MdrProdInstance></usm:MdrProdInstance>
    <usm:SourceMdrProduct>CA:00996</usm:SourceMdrProduct>
    <usm:SourceMdrProdInstance></usm:SourceMdrProdInstance>
    <usm:SourceMdrElementID>MyServiceA</usm:SourceMdrElementID>
    <usm:TargetMdrProduct>CA:00996</usm:TargetMdrProduct>
    <usm:TargetMdrProdInstance></usm:TargetMdrProdInstance>
    <usm:TargetMdrElementID>MyComputerA</usm:TargetMdrElementID>
    <usm:ScopeMdrProduct>CA:00996</usm:ScopeMdrProduct>
    <usm:ScopeMdrProdInstance></usm:ScopeMdrProdInstance>
    <usm:ScopeMdrElementID>MyServiceA</usm:ScopeMdrElementID>
    <usm:Semantic>Depends On</usm:Semantic>
    <usm:Significance>5</usm:Significance>
</usm:BinaryRelationship>
   </env:Body>
</env:Envelope>
```

Relationship Web Services Examples

The following examples show the SOAP messages of many of the available relationship web services.

Example: Get a list of relationships for a specific service

The following example SOAP messages are Enumerate and Pull requests to retrieve a list of relationships associated with the servicetest service:

```
<env:Envelope>
<env:Header>
   <wsa:Action xmlns:ns13="http://ns.ca.com/2009/07/usm-core" env:mustUnderstand="true">
    http://schemas.xmlsoap.org/ws/2004/09/enumeration/Enumerate
   </wsa:Action>
   <wsa:ReplyTo xmlns:ns13="http://ns.ca.com/2009/07/usm-core">
   <wsa:Address env:mustUnderstand="true">
    http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
   </wsa:Address>
   </wsa:ReplyTo>
   <wsa:MessageID xmlns:ns13="http://ns.ca.com/2009/07/usm-core" env:mustUnderstand="true">
    uuid:c189ea6b-b07f-48c4-8060-819c1dd2da12
   </wsa:MessageID>
   <wsa:To xmlns:ns13="http://ns.ca.com/2009/07/usm-core" env:mustUnderstand="true">
    http://localhost:7090/sam/webservice
   </wsa:To>
   <wsman:ResourceURI xmlns:ns13="http://ns.ca.com/2009/07/usm-core">
    http://ns.ca.com/2009/07/usm-core/BinaryRelationship
   </wsman:ResourceURI>
   <wsman:SelectorSet xmlns:ns13="http://ns.ca.com/2009/07/usm-core">
   <wsman:Selector Name="serviceName">SA_Service:servicetest
   </wsman:Selector>
   </wsman:SelectorSet>
   <wsman:RequestTotalItemsCountEstimate  xmlns:ns13="http://ns.ca.com/2009/07/usm-core"/>
   </env:Header>
   <env:Body>
    <wsen:Enumerate xmlns:ns13="http://ns.ca.com/2009/07/usm-core">
    <wsman:EnumerationMode>EnumerateObjectAndEPR</wsman:EnumerationMode>
    </wsen:Enumerate>
   </env:Body>
   </env:Envelope>
   <env:Envelope>
   <env:Header>
   <wsa:Action xmlns:ns13="http://ns.ca.com/2009/07/usm-core" env:mustUnderstand="true">
    http://schemas.xmlsoap.org/ws/2004/09/enumeration/Pull
   </wsa:Action>
   <wsa:ReplyTo xmlns:ns13="http://ns.ca.com/2009/07/usm-core">
   <wsa:Address env:mustUnderstand="true">
    http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
   </wsa:Address>
   </wsa:ReplyTo>
   <wsa:MessageID xmlns:ns13="http://ns.ca.com/2009/07/usm-core" env:mustUnderstand="true">
    uuid:630fa96e-7f01-46bd-b3aa-7f79dc57445c
   </wsa:MessageID>
   <wsa:To xmlns:ns13="http://ns.ca.com/2009/07/usm-core" env:mustUnderstand="true">
    http://localhost:7090/sam/webservice
   </wsa:To>
   <wsman:ResourceURI xmlns:ns13="http://ns.ca.com/2009/07/usm-core">
    http://ns.ca.com/2009/07/usm-core/BinaryRelationship
```

```
   </wsman:ResourceURI>
   <wsman:OperationTimeout xmlns:ns13="http://ns.ca.com/2009/07/usm-core">
    P0Y0M0DT0H0M30.000S
   </wsman:OperationTimeout>
   </env:Header>
   <env:Body>
   <wsen:Pull xmlns:ns13="http://ns.ca.com/2009/07/usm-core">
     <wsen:EnumerationContext>7a26ba28-7c8e-46dc-b456-f70c67d2f2b6
     </wsen:EnumerationContext>
     <wsen:MaxTime>P0Y0M0DT0H0M30.000S</wsen:MaxTime>
     <wsen:MaxElements>20</wsen:MaxElements>
   </wsen:Pull>
</env:Body>
</env:Envelope>
```

The bold SelectorSet syntax defines the service for which to list relationships. The SOAP response to this request is as follows:

```
<env:Envelope>
<env:Header>
   <wsa:Action env:mustUnderstand="true"
   xmlns:ns14="http://ns.ca.com/2009/07/usm-core">
    http://schemas.xmlsoap.org/ws/2004/09/enumeration/PullResponse
   </wsa:Action>
   <wsa:MessageID env:mustUnderstand="true">
    uuid:6bbbc6ad-3f91-4191-ac58-b17dce91d581
   </wsa:MessageID>
   <wsa:RelatesTo >uuid:630fa96e-7f01-46bd-b3aa-7f79dc57445c</wsa:RelatesTo>
   <wsa:To env:mustUnderstand="true" >
    http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
   </wsa:To>
   </env:Header>
   <env:Body>
   <wsen:PullResponse >
   <wsen:Items>
   <wsman:Item>
   <BinaryRelationship>
      <MdrProduct>CA:00047</MdrProduct>
             <MdrProdInstance>server1</MdrProdInstance>
             <MdrElementID>21:21:22</MdrElementID>
             <ssa_usm_rel_id>IsAffectedBy</ssa_usm_rel_id>
             <ssa_impact>0</ssa_impact>
             <ssa_policy_id>0</ssa_policy_id>
             <ssa_relevance>0</ssa_relevance>
             <ssa_root_cause>false</ssa_root_cause>
             <SourceMdrProduct>CA:00047</SourceMdrProduct>
             <SourceMdrProdInstance>server1</SourceMdrProdInstance>
             <SourceMdrElementID>21</SourceMdrElementID>
             <ns12:TargetMdrProduct>CA:00047</TargetMdrProduct>
             <ns12:TargetMdrProdInstance>server2</TargetMdrProdInstance>
             <ns12:TargetMdrElementID>22</TargetMdrElementID>
             <ns12:Semantic>Depends On</Semantic>
             <ns12:ScopeMdrProduct>CA:00047</ScopeMdrProduct>
             <ns12:ScopeMdrProdInstance>server1</ScopeMdrProdInstance>
```

```
      <ns12:ScopeMdrElementID>21</ScopeMdrElementID>
      <ns12:Significance>5</Significance>
  </BinaryRelationship>
  <wsa:EndpointReference>
  <wsa:Address env:mustUnderstand="true">
http://localhost:7090/sam/webservice</wsa:Address>
  <wsa:ReferenceParameters>
  <wsman:ResourceURI>
http://ns.ca.com/2009/07/usm-core/BinaryRelationship
  </wsman:ResourceURI>
  <wsman:SelectorSet>
  <wsman:Selector Name="MdrProduct">CA:00047</wsman:Selector>
  <wsman:Selector Name="MdrElementID">21:21:22</wsman:Selector>
  <wsman:Selector Name="MdrProdInstance">server1</wsman:Selector>
  </wsman:SelectorSet>
  </wsa:ReferenceParameters>
  </wsa:EndpointReference>
  </wsman:Item>
  <wsman:Item>
  <ns12:BinaryRelationship>
    <ns12:MdrProduct>CA:00047</ns12:MdrProduct>
    <ns12:MdrProdInstance>server3</ns12:MdrProdInstance>
    <ns12:MdrElementID>21:21:9</ns12:MdrElementID>
    <ssa_usm_rel_id>IsAffectedBy</ssa_usm_rel_id>
    <ssa_impact>0</ssa_impact>
    <ssa_policy_id>0</ssa_policy_id>
    <ssa_relevance>0</ssa_relevance>
    <ssa_root_cause>false</ssa_root_cause>
    <ns12:SourceMdrProduct>CA:00047</ns12:SourceMdrProduct>
    <ns12:SourceMdrProdInstance>server3</ns12:SourceMdrProdInstance>
    <ns12:SourceMdrElementID>21</ns12:SourceMdrElementID>
    <ns12:TargetMdrProduct>CA:00047</ns12:TargetMdrProduct>
    <ns12:TargetMdrProdInstance>server4</ns12:TargetMdrProdInstance>
    <ns12:TargetMdrElementID>9</ns12:TargetMdrElementID>
    <ns12:Semantic>Depends On</ns12:Semantic>
    <ns12:ScopeMdrProduct>CA:00047</ns12:ScopeMdrProduct>
    <ns12:ScopeMdrProdInstance>server3</ns12:ScopeMdrProdInstance>
    <ns12:ScopeMdrElementID>21</ns12:ScopeMdrElementID>
    <ns12:Significance>5</ns12:Significance>
  </BinaryRelationship>
  <wsa:EndpointReference>
  <wsa:Address
  env:mustUnderstand="true">http://localhost:7090/sam/webservice
  </wsa:Address>
  <wsa:ReferenceParameters>
  <wsman:ResourceURI>http://ns.ca.com/2009/07/usm-core/BinaryRelationship
  </wsman:ResourceURI>
  <wsman:SelectorSet>
  <wsman:Selector Name="MdrProduct">CA:00047</wsman:Selector>
  <wsman:Selector Name="MdrElementID">21:21:9</wsman:Selector>
  <wsman:Selector Name="MdrProdInstance">symbe01-5</wsman:Selector>
  </wsman:SelectorSet>
  </wsa:ReferenceParameters>
```

```
          </wsa:EndpointReference>
          </wsman:Item>
          </wsen:Items>
          <wsen:EndOfSequence xsi:type="xs:string" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
   </wsen:PullResponse>
</env:Body>
</env:Envelope>
```

The bold syntax shows the details of the returned relationships for the servicetest service. Note the returned properties, including the relationship type (ssa_usm_rel_id), propagation type (Semantic), and source and target CIs.

Example: Create a relationship

The following example SOAP message is a Create request to create a new relationship with Aggregates propagation:

```
<env:Envelope>
<env:Header>
   <wsa:ReplyTo xmlns:usm="http://ns.ca.com/2009/07/usm-core">
   <wsa:Address env:mustUnderstand="true">
    http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
   </wsa:Address>
   </wsa:ReplyTo>
   <wsa:MessageID xmlns:usm="http://ns.ca.com/2009/07/usm-core"
   env:mustUnderstand="true">
    uuid:4add8974-25a0-4fc4-84f6-c7f88158f8f7
   </wsa:MessageID>
   <wsa:To xmlns:usm="http://ns.ca.com/2009/07/usm-core" env:mustUnderstand="true">
    http://localhost:7090/sam/webservice
   </wsa:To>
   <wsman:ResourceURI xmlns:usm="http://ns.ca.com/2009/07/usm-core">
    http://ns.ca.com/2009/07/usm-core/BinaryRelationship
   </wsman:ResourceURI>
   <wsman:OperationTimeout xmlns:usm="http://ns.ca.com/2009/07/usm-core">
    PT30.000S
   </wsman:OperationTimeout>
   <wsa:Action xmlns:usm="http://ns.ca.com/2009/07/usm-core"
   env:mustUnderstand="true">
    http://schemas.xmlsoap.org/ws/2004/09/transfer/Create
   </wsa:Action>
   </env:Header>
   <env:Body>
   <BinaryRelationship xmlns:ns14="http://ns.ca.com/2009/07/usm-core">
      <MdrProduct>CA:09997</MdrProduct>
      <MdrProdInstance>server1.ca.com</MdrProdInstance>
      <MdrElementID>Agg_Service:Agg_Service:Dep_Server</MdrElementID>
      <SourceMdrProduct>CA:09997</SourceMdrProduct>
      <SourceMdrProdInstance>server1.ca.com</SourceMdrProdInstance>
      <SourceMdrElementID>SA_Service:Agg_Service</SourceMdrElementID>
      <TargetMdrProduct>CA:09997</TargetMdrProduct>
      <TargetMdrProdInstance>server2.ca.com</TargetMdrProdInstance>
      <TargetMdrElementID>SA_Server:Dep_Server</TargetMdrElementID>
      <Semantic>Aggregates</Semantic>
      <ScopeMdrProduct>CA:09997</ScopeMdrProduct>
      <ScopeMdrProdInstance>server1.ca.com</ScopeMdrProdInstance>
      <ScopeMdrElementID>SA_Service:Agg_Service</ScopeMdrElementID>
```

```
      <Significance>5</Significance>
      <ssa_connector_name>server1.ca.com</ssa_connector_name>
      <ssa_silo_name>WebServiceForSSA_server1.ca.com@server1.ca.com</ssa_silo_name>
   </BinaryRelationship>
</env:Body>
</env:Envelope>
```

The web service creates the relationship using the properties listed in bold in the body of the request. Note that the Semantic property defines the propagation type for the relationship as Aggregates. No relationship type is defined, so the relationship automatically obtains the default type that maps to Aggregates propagation. The SOAP response to this request is as follows:

```
<env:Envelope>
<env:Header>
   <wsa:Action env:mustUnderstand="true"
   xmlns:ns14="http://ns.ca.com/2009/07/usm-core">
    http://schemas.xmlsoap.org/ws/2004/09/transfer/CreateResponse
   </wsa:Action>
   <wsa:MessageID env:mustUnderstand="true"
   xmlns:ns14="http://ns.ca.com/2009/07/usm-core">
    uuid:d23a181b-5e2d-4dae-85a0-8ee4415c227b
   </wsa:MessageID>
   <wsa:RelatesTo xmlns:ns14="http://ns.ca.com/2009/07/usm-core">
    uuid:4add8974-25a0-4fc4-84f6-c7f88158f8f7
   </wsa:RelatesTo>
   <wsa:To env:mustUnderstand="true" xmlns:ns14="http://ns.ca.com/2009/07/usm-core">
    http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
   </wsa:To>
   </env:Header>
   <env:Body>
   <wxf:ResourceCreated xmlns:ns14="http://ns.ca.com/2009/07/usm-core">
   <wsa:Address env:mustUnderstand="true">
   http://localhost:7090/sam/webservice/</wsa:Address>
   <wsa:ReferenceParameters>
   <wsman:ResourceURI>http://ns.ca.com/2009/07/usm-core/BinaryRelationship
   </wsman:ResourceURI>
   </wsa:ReferenceParameters>
   </wxf:ResourceCreated>
</env:Body>
</env:Envelope>
```

Subscribe to Notifications for Relationship Changes

To subscribe to notifications for changes to Relationship, use the Notification web services. The Relationship resource supports the WS-Eventing functionality that enables a web client to subscribe to notification events when a relationship is created, deleted, or updated.

> **NOTE**
> For more information, see Notification Web Services.

Notification Web Services

Contents

This section provides information about the operations performed in Notification web services.

Notification Web Services Overview

Notification web services are used to subscribe to notifications for changes to Entities, Alerts, and Relationships.

Use the following endpoint URI when invoking the Notification web services resource:

```
http://ns.ca.com/2009/07/usm-core/Notification
```

The following WSDL file outlines the available operations:

```
http://<samanager>:<port>/sam/webservice/wsdls/usm-core-200907.wsdl
```

Access the USM schema as follows:

```
http://<samanager>:<port>/sam/webservice/schemas/usm-core-200907.xsd
```

Using the Notification web services requires basic knowledge of the USM schema and its properties. In addition to the schema itself, HTML documentation is available. For information see How to Access the USM Schema Documentation.

How to Subscribe to Notifications for Entity, Relationship, and Alert Changes

Addition of the WS-Eventing functionality to the Entity, Relationship, and Alert resources allows a web client to subscribe to notification events when an entity (CI), relationship, or alert is created, deleted, or updated. The web client sends a subscription request with a filter detailing what notification events the client wants to receive and the mode of delivery. If the request is successful, the web service sends a response with a subscription ID. The subscription provides two delivery modes: Pull and Push. In case of Pull, a client periodically polls for notification events, and in case of Push, the notification events are published to an *event sink*, where the web client can process them.

You can perform the following steps to complete the task:

1. Send a Subscribe request for the Notification resource.
2. Add a filter with the following events as appropriate:
   - entityCreated, entityModified, and entityDeleted for the Entity resource
   - binaryrelationshipCreated, binaryrelationshipModified, and binaryrelationshipDeleted for the Relationship resource
   - alertCreated, alertModified, and alertCleared for the Alert resource
3. Update authentication details (user ID and password).
4. Run the request, and review the response for the subscription ID.

   NOTE
   For Pull mode, send a Pull request for the Notification resource.

Create a Subscription

Use the Subscribe operation to create a subscription request for notification. This request is sent to the event source.

To create a subscription, use the following properties in the request:

Operation: Subscribe

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/07/usm-core/Notification

Creating a subscription request does not require any selector; it requires a filter. Therefore, the web client must provide the following information:

- Address of the web service
- Resource URL
- Action (Subscribe)
- Reply-to Address (default)
- MessageID (uuid)
- Delivery Mode (Pull in this case)
- Heartbeats (Timeout value for subscription)
- Filter (what events to subscribe for)
- Bookmark (from what point to start sending events)

> **NOTE**
> For more information about how to create this request, see the first example in the Notification Web Services Examples section.

Delete a Subscription

Use the Unsubscribe operation to explicitly delete a subscription when you do not want notifications associated with the subscription.

To delete a subscription, use the following properties in the request:

Operation: Unsubscribe

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/07/usm-core/Notification

Deleting a subscription does not require any selector; it requires a filter. Therefore, the web client must provide the following information:

- Address of the web service
- Resource URL
- Action (Unsubscribe)
- Reply-to Address (default)
- MessageID (uuid)
- Identifier (Subscription ID)

> **NOTE**
> For more information about how to create this request, see the second example in the Notification Web Services Examples section.

Pull a Subscription Notification

Use the WS-Management Pull operation in combination with the delivery mode *Pull* to retrieve periodically any notification events such as entityModified, entityCreated, and entityDeleted for Entity; binaryrelationshipModified, binaryrelationshipCreated, and binaryrelationshipDeleted for Relationship; and alertModified, alertCreated, and alertDeleted for Alert.

To pull a subscription notification, use the following properties in the request:

Operation: Pull

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/07/usm-core/Notification

EnumerationContext: subscriptionID

NOTE
For more information about how to create this request, see the third example in the Notification Web Services Examples section.

Notification Web Services Examples

The following examples show the SOAP requests to create a subscription, retrieve notification events, and delete a subscription for Entity.

Example 1: Create a subscription

The following example SOAP request shows how you can create a subscription request:

```
<env:Envelope>
<env:Header>
   <wsa:To env:mustUnderstand="true">
    http://localhost:7090/sam/webservice
   </wsa:To>
   <wsman:ResourceURI>
    http://ns.ca.com/2009/07/usm-core/Notification
   </wsman:ResourceURI>
   <wsa:Action  env:mustUnderstand="true">
    http://schemas.xmlsoap.org/ws/2004/08/eventing/Subscribe
   </wsa:Action>
   <wsa:ReplyTo>
   <wsa:Address env:mustUnderstand="true">
    http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
   </wsa:Address>
   </wsa:ReplyTo>
   <wsa:MessageID env:mustUnderstand="true">
    uuid:afb3d3ee-cdb2-4587-9087-d09d77ab5d8d
   </wsa:MessageID>
</env:Header>
   <env:Body>
   <wse:Subscribe>
    <wse:Delivery Mode="http://schemas.dmtf.org/wbem/wsman/1/wsman/Pull">
    <wsman:Heartbeats>PT5M0.000S</wsman:Heartbeats>
   </wse:Delivery>
   <wse:Filter Dialect="http://ns.ca.com/2009/07/usm-core/NotificationFilter">
    entityModified;entityCreated;entityDeleted
   </wse:Filter>
   <wsman:Bookmark>
   <ns15:Bookmark>
    http://schemas.dmtf.org/wbem/wsman/1/wsman/bookmark/earliest
   </ns15:Bookmark>
   </wsman:Bookmark>
   </wse:Subscribe>
   </env:Body>
</env:Envelope>
```

Example 2: Delete a Subscription

The following example SOAP request shows how to delete a subscription:

```
<env:Envelope >
<env:Header>
```

```
   <wsa:To env:mustUnderstand="true">
    http://localhost:7090/sam/webservice
   </wsa:To>
   <wsman:ResourceURI >
    http://ns.ca.com/2009/07/usm-core/Notification
   </wsman:ResourceURI>
   <wsa:Action env:mustUnderstand="true">
    http://schemas.xmlsoap.org/ws/2004/08/eventing/Unsubscribe
   </wsa:Action>
   <wsa:ReplyTo>
   <wsa:Address env:mustUnderstand="true">
    http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
   </wsa:Address>
   </wsa:ReplyTo>
   <wsa:MessageID env:mustUnderstand="true">
    uuid:0f589af8-1b38-4948-a171-f4dc419f49db
   </wsa:MessageID>
   <wse:Identifier >
    5264d7de-c6ac-46f4-80ea-fd59c11a7561
   </wse:Identifier>
</env:Header>
   <env:Body>
    <wse:Unsubscribe/>
   </env:Body>
</env:Envelope>
```

Example 3: Retrieve Notification Events

The following example SOAP request shows how to retrieve notification events:

```
<env:Envelope>
<env:Header>
   <wsa:To env:mustUnderstand="true">
    http://localhost:7090/sam/webservice
   </wsa:To>
   <wsman:ResourceURI >
    http://ns.ca.com/2009/07/usm-core/Notification
   </wsman:ResourceURI>
   <wsman:OperationTimeout>
    PT0.100S
   </wsman:OperationTimeout>
   <wsa:Action env:mustUnderstand="true">
    http://schemas.xmlsoap.org/ws/2004/09/enumeration/Pull
   </wsa:Action>
   <wsa:ReplyTo >
   <wsa:Address env:mustUnderstand="true">
    http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
   </wsa:Address>
   </wsa:ReplyTo>
   <wsa:MessageID env:mustUnderstand="true">
    uuid:7f71fa61-1e94-4265-97e4-f4bf0ddd7e68
   </wsa:MessageID>
</env:Header>
   <env:Body>
```

```
    <wsen:Pull >
    <wsen:EnumerationContext>
        5264d7de-c6ac-46f4-80ea-fd59c11a7561
    </wsen:EnumerationContext>
    <wsen:MaxTime>
        PT0.100S
    </wsen:MaxTime>
    <wsen:MaxElements>
        10
    </wsen:MaxElements>
    </wsen:Pull>
   </env:Body>
</env:Envelope>
```

Queue Web Services

Contents

This section provides information about the operations performed in queue web services.

> **NOTE**
> WS-MAN web services should be considered obsolete. We recommend using the Alert Queue REST web services instead.

Queue Web Services Overview

Queue web services use the USM 01-2009 schema to perform operations on alert management queues in CA SOI. Alert queues collect and logically organize closely related groups of alerts in CA SOI.

Use the following endpoint URI when invoking the queue web services resource:

```
http://ns.ca.com/2009/01/usm-data/Queue
```

The following WSDL file outlines the available operations:

```
http://<samanager>:<port>/sam/webservice/wsdls/usm2.wsdl
```

Access the USM 01-2009 schema as follows:

```
http://<samanager>:<port>/sam/webservice/schemas/usm2.xsd
```

> **NOTE**
> For more information about queues, see How to Create and Manage Alert Queues.

Get a Queue

Use the Get request to retrieve a specific queue. The following selectors are required as part of the request to identify a unique instance of a queue:

- **ASBOID.id**
  Uniquely identifies the queue using the Action ID value. Derive this value using an Enumerate operation.
- **ASBOID.source**
  Defines the DomainID of the CA SOI model repository. This value is constant for the SA Manager. Derive the value using an Enumerate operation.

To get a queue, use the following properties in the request:

Operation: Get

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/01/usm-data/Queue

Selector: ASBOID.id

Selector: ASBOID.source

The QueueHandlerImpl.Get() method returns the queue based on the selectors.

Get a List of Queues

To retrieve a list of queues, the web services use a combination of WS-Management Enumeration and Pull operations. You can filter the returned list using the WS-Management Filter element to pass a valid XPath expression to limit the number and type of queues returned.

To get a list of queues, use the following properties in the request:

Operation: Enumerate & Pull

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/01/usm-data/Queue

Selector: Null

The QueueIteratorImpl() method creates a collection of all alert queues in CA SOI. The Pull operation retrieves queues in batches as defined by the MaxElements tag.

Create a Queue

Use the Create operation to create an alert queue in CA SOI. You define the queue type and property values for the new queue in the body of the request.

> **NOTE**
> For information about the required properties to include for a queue, see the USM 01-2009 schema.

To create a queue, use the following properties in the request:

Operation: Create

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/01/usm-data/Queue

Selector: Null

The QueueHandlerImpl.Create() method extracts all defined property values from the body of the SOAP message and creates the queue in CA SOI.

Update a Queue

Use the Put operation to update the writeable properties of a queue. Perform the update by passing in all of the properties and their new values in the body of the request.

To update a queue, use the following properties in the request:

Operation: Put

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/01/usm-data/Queue

Selector: ASBOID.id

Selector: ASBOID.source

Fragment: property (Description indicating which property to update)

The QueueHandlerImpl.Put() method inspects the SOAP Header and determines if it is a fragment-based update. If so, only the attributes specified in the Fragment are updated; otherwise, all of the writable attributes are updated.

> **NOTE**
> For information about the required properties to include for a queue, see the USM 01-2009 schema.

Delete a Queue

Use the Delete operation to delete a queue. Use the queue ASBOID.id and ASBOID.source values as selectors.

To delete a queue, use the following properties in the request:

Operation: Delete

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/01/usm-data/Queue

Selector: ASBOID.id

Selector: ASBOID.source

The QueueHandlerImpl.Delete() method verifies that the queue exists and deletes it.

Queue Web Services Examples

The following examples show the SOAP messages of many of the available Queue web services.

Example 1: Get a Queue

The following example SOAP message is a Get request to retrieve a queue:

```
<env:Envelope>
<env:Header>
   <wsa:To>http://localhost:7090/sam/webservice</wsa:To>
   <wsman:ResourceURI >
    http://ns.ca.com/2009/01/usm-data/Queue
   </wsman:ResourceURI>
   <wsman:SelectorSet>
    <wsman:Selector Name="ASBOID.source">4503599627370496</wsman:Selector>
    <wsman:Selector Name="ASBOID.id">2</wsman:Selector>
   </wsman:SelectorSet>
   <wsa:Action>
    http://schemas.xmlsoap.org/ws/2004/09/transfer/Get
   </wsa:Action>
   <wsa:ReplyTo>
   <wsa:Address>
    http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
   </wsa:Address>
   </wsa:ReplyTo>
    <wsa:MessageID>uuid:78372ad8-46e7-4d27-b632-7e2de827f29a</wsa:MessageID>
</env:Header>
<env:Body/>
</env:Envelope>
```

Example 2: Get a Queue Response

The following example SOAP message shows how you can get a queue response:

```
<env:Envelope>
<env:Header>
   <wsa:Action>
    http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse
   </wsa:Action>
   <wsa:MessageID>
    uuid:607204b7-daf5-4ba1-966a-ebab3628b498
   </wsa:MessageID>
   <wsa:RelatesTo>uuid:78372ad8-46e7-4d27-b632-7e2de827f29a</wsa:RelatesTo>
   <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
</env:Header>
<env:Body>
   <ns13:Queue>
   <ssa_escalation_policy_id_1>2</ssa_escalation_policy_id_1>
   <sam_alert_queue_id>0x13400000000002</sam_alert_queue_id>
   <USMID>4503599627370496:2</USMID>
   <ASBOID>
    <source>4503599627370496</source>
    <id>2</id>
   </ASBOID>
   <ns13:item_name>WebServices Created Queue</ns13:item_name>
   <ns13:item_description>WS alarm queue10</ns13:item_description>
   <ns13:item_creation_date>2012-02-15T13:17:54.713+11:00</ns13:item_creation_date>
   <ns13:item_creation_user>symbe01</ns13:item_creation_user>
   <ns13:CriteriaXml><![CDATA[<attr-filter><and><equals-ignore-case><attribute id="0x11f57"><value>symbe01</
value></attribute></equals-ignore-case><equals-ignore-case><attribute id="0x12a08"><value>Windows</value></
attribute></equals-ignore-case></and></attr-filter>]] ></ns13:CriteriaXml>
   <ns13:CriteriaDrool>package com.ca.sam.manager.rules
   import com.ca.sam.manager.rules.AlarmObject;
   rule "Queue 2"
    when
        $alarm : AlarmObject( ((assignedTo matches "(?i)symbe01" &#38;&#38; situationType matches "(?
i)Windows" ))  )
    then
        $alarm.assignQueue(2);
   end</ns13:CriteriaDrool>
   <ns13:CompileTime>2012-02-28T08:21:21.797+11:00</ns13:CompileTime>
   <ns13:Priority>2</ns13:Priority>
   <ns13:NumberOfAlerts>0</ns13:NumberOfAlerts>
   <ns13:CountMinor>0</ns13:CountMinor>
   <ns13:CountMajor>0</ns13:CountMajor>
   <ns13:CountCritical>0</ns13:CountCritical>
   <ns13:CountDown>0</ns13:CountDown>
   </ns13:Queue>
</env:Body>
</env:Envelope>
```

Example 3: Create a Queue

The following example SOAP message is a Create request to create a queue:

```
<env:Envelope>
```

```
<env:Header>
   <wsa:ReplyTo>
   <wsa:Address>
    http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
   </wsa:Address>
   </wsa:ReplyTo>
   <wsa:MessageID>uuid:697da93f-3bc4-479f-8007-fcd0d860d2cb</wsa:MessageID>
   <wsa:To>http://au-symbe01-w8e2:7090/sam/webservice</wsa:To>
   <wsman:ResourceURI>http://ns.ca.com/2009/01/usm-data/Queue
   </wsman:ResourceURI>
   <wsman:OperationTimeout>P0Y0M0DT0H0M30.000S</wsman:OperationTimeout>
   <wsa:Action>http://schemas.xmlsoap.org/ws/2004/09/transfer/Create</wsa:Action>
   </env:Header>
   <env:Body>
   <ns13:Queue>
    <ns13:item_name>NewQueue</ns13:item_name>
    <ns13:item_description>My New Queue description</ns13:item_description>
    <ns13:item_creation_user>symbe01</ns13:item_creation_user>
    <ns13:CriteriaXml><![CDATA[<attr-filter><and><equals-ignore-case><attribute id="0x11f57"><value>symbe01</
value></attribute></equals-ignore-case><equals-ignore-case><attribute id="0x12a08"><value>Windows</value></
attribute></equals-ignore-case></and></attr-filter>]] ></ns13:CriteriaXml>
   </ns13:Queue>
</env:Body>
</env:Envelope>
```

Example 4: Update a Queue

The following example SOAP message is a Put request that shows how you can update a queue:

```
<env:Envelope>
<env:Header>
   <wsa:To>http://localhost:7090/sam/webservice</wsa:To>
   <wsman:ResourceURI>http://ns.ca.com/2009/01/usm-data/Queue</wsman:ResourceURI>
   <wsman:OperationTimeout>P0Y0M0DT0H0M30.000S</wsman:OperationTimeout>
   <wsman:SelectorSet>
    <wsman:Selector Name="ASBOID.source">4503599627370496</wsman:Selector>
    <wsman:Selector Name="ASBOID.id">2</wsman:Selector>
   </wsman:SelectorSet>
   <wsa:Action>http://schemas.xmlsoap.org/ws/2004/09/transfer/Put</wsa:Action>
   <wsa:ReplyTo>
    <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
   </wsa:ReplyTo>
   <wsa:MessageID>uuid:f4ea8e55-a4bc-4faf-9dd1-010d1767c8f4</wsa:MessageID>
   <wsman:FragmentTransfer>item_description</wsman:FragmentTransfer>
</env:Header>
<env:Body>
   <ns11:Queue>
    <ASBOID>
    <source>4503599627370496</source>
    <id>2</id>
    </ASBOID>
    <ns11:item_description>Updated desc</ns11:item_description>
   </ns11:Queue>
</env:Body>
```

```
</env:Envelope>
```

Example 5: Delete a Queue

The following example SOAP message is a Delete request that shows how you can delete a queue:

```
<env:Envelope>
<env:Header>
   <wsa:To>http://localhost:7090/sam/webservice</wsa:To>
   <wsman:ResourceURI>http://ns.ca.com/2009/01/usm-data/Queue</wsman:ResourceURI>
   <wsman:OperationTimeout>P0Y0M0DT0H0M30.000S</wsman:OperationTimeout>
   <wsman:SelectorSet>
   <wsman:Selector Name="ASBOID.source">4503599627370496</wsman:Selector>
   <wsman:Selector Name="ASBOID.id">2</wsman:Selector>
   </wsman:SelectorSet>
    <wsa:Action>http://schemas.xmlsoap.org/ws/2004/09/transfer/Delete</wsa:Action>
   <wsa:ReplyTo>
    <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
   </wsa:ReplyTo>
   <wsa:MessageID>uuid:f4ea8e55-a4bc-4faf-9dd1-010d1767c8f4</wsa:MessageID>
   <wsman:FragmentTransfer>item_description</wsman:FragmentTransfer>
</env:Header>
<env:Body/>
</env:Envelope>
```

Customer Web Services

Contents

This section provides information about the operations that you can perform in CA SOI using the Customer web services resource.

> **NOTE**
> WS-MAN web services should be considered obsolete. We recommend using the Customer REST web services instead.

Customer Web Services Overview

Customer web services resource uses the USM 01-2009 schema to perform customer-related operations in CA SOI. A customer in CA SOI is any consumer of a managed service. The CA SOI administrator creates customers and associates them with service models to see the impact of service degradation on the customer.

Use the following endpoint URI when invoking the customer web services resource:

```
http://ns.ca.com/2009/01/usm-data/Customer
```

The following WSDL file outlines the available operations:

```
http://<samanager>:<port>/sam/webservice/wsdls/usm2.wsdl
```

Access the USM 01-2009 schema as follows:

```
http://<samanager>:<port>/sam/webservice/schemas/usm2.xsd
```

> **NOTE**
> For more information about customers, see How to Create and Manage Customers.

Get a Customer

Use the Get request to retrieve a specific customer. The following selectors are required as part of the request to identify a unique customer:

- **ASBOID.id**
  Uniquely identifies the customer using the Action ID value. Derive this value using an Enumerate operation.
- **ASBOID.source**
  Defines the DomainID of the CA SOI model repository. This value is constant for the SA Manager. Derive the value using an Enumerate operation.

To get a customer, use the following properties in the request:

Operation: Get

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/01/usm-data/Customer

Selector: ASBOID.id

Selector: ASBOID.source

The CustomerHandlerImpl.Get() method returns the customer based on the selectors.

Get a List of Customers

To retrieve a list of customers, the web services use a combination of WS-Management Enumeration and Pull operations. You can filter the returned list using the WS-Management Filter element to pass a valid XPath expression to limit the number of customers returned.

To get a list of customers, use the following properties in the request:

Operation: Enumerate & Pull

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/01/usm-data/Customer

Selector: Null

The CustomerIteratorImpl() method creates a collection of all customers in CA SOI. The Pull operation retrieves customers in batches as defined by the MaxElements tag.

Customer Web Services Examples

The following example shows the SOAP messages of the customer web services.

Example: Get a Customer

The following example SOAP message is a Get request to retrieve a customer:

```
<env:Envelope>
  <env:Header>
    <wsa:To>http://localhost:7090/sam/webservice</wsa:To>
    <wsman:ResourceURI>
         http://ns.ca.com/2009/01/usm-data/Customer
    </wsman:ResourceURI>
    <wsman:SelectorSet>
         <wsman:Selector Name="ASBOID.source">4503599627370496
    </wsman:Selector>
         <wsman:Selector Name="ASBOID.id">1</wsman:Selector>
```

```
    </wsman:SelectorSet>
    <wsa:Action>
         http://schemas.xmlsoap.org/ws/2004/09/transfer/Get
    </wsa:Action>
    <wsa:ReplyTo>
      <wsa:Address>
         http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
      </wsa:Address>
    </wsa:ReplyTo>
    <wsa:MessageID>
         uuid:78372ad8-46e7-4d27-b632-7e2de827f29a
    </wsa:MessageID>
  </env:Header>
<env:Body/>
</env:Envelope>
```

The response of the Get request is as follows:

```
<env:Envelope>
  <env:Header>
     <wsa:Action>
    http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse
     </wsa:Action>
     <wsa:MessageID>
    uuid:6b58e83b-b02a-4a5e-8650-61fdc8842a77
     </wsa:MessageID>
     <wsa:RelatesTo>
    uuid:78372ad8-46e7-4d27-b632-7e2de827f29a
     </wsa:RelatesTo>
     <wsa:To>
    http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
     </wsa:To>
  </env:Header>
  <env:Body>
     <ns13:Customer>
    <USMID>4503599627370496:1</USMID>
    <ASBOID>
        <source>4503599627370496</source>
        <id>1</id>
    </ASBOID>
    <ns13:item_name>customer1</ns13:item_name>
    <ns13:item_description>first customer</ns13:item_description>
    <ns13:priority>10</ns13:priority>
    <ns13:identifier>c12345</ns13:identifier>
    <ns13:quality>0</ns13:quality>
    <ns13:risk>5</ns13:risk>
    <ns13:health>1</ns13:health>
    <ns13:topLevel>true</ns13:topLevel>
    <ns13:services>3</ns13:services>
    <ns13:services>16</ns13:services>
      </ns13:Customer>
   </env:Body>
</env:Envelope>
```

Alert Web Services

Contents

This section describes the operations performed in alert web services.

> **NOTE**
> WS-MAN web services should be considered obsolete. We recommend using the Alert REST web services instead.

Alert Web Services Overview

Alert web services let you interact with CA SOI alerts. Use the following endpoint URI when invoking the alert web services resource:

```
http://ns.ca.com/2009/07/usm-core/Alert
```

The following WSDL file outlines the available operations:

```
http://<samanager>:<port>/sam/webservice/wsdls/usm-core-200907.wsdl
```

Access the USM schema as follows:

```
http://<samanager>:<port>/sam/webservice/schemas/usm-core-200907.xsd
```

Using the alert web services requires basic knowledge of the USM schema and its properties. In addition to the schema itself, HTML documentation is available. For more information see How to Access the USM Schema Documentation.

Get an Alert

Use the Get request to retrieve an alert. The following selectors are required as part of the request to identify a unique instance of an alert:

- **MdrProduct**
  Defines the connector data source. Each connector has a specific MdrProduct value formatted as a five-digit number prefixed by 'CA:'. For example, the MdrProduct value for resources created by web services is CA:09996. For a list of MdrProduct values, see the *Connector Guide*.
- **MdrProdInstance**
  Defines the host name associated with the resource.
- **MdrElementID**
  Defines a value that uniquely identifies the resource.

> **NOTE**
> This request also retrieves the root cause property ssa_rootcause_alert_id of service-related alerts to indicate the alert that is the root cause alert, and ssa_is_rootcause property of CI alerts to indicate whether CI alerts are root cause alerts.

To get an alert, use the following properties in the request:

Operation: Get

Endpoint: http://<samanager>:<port>/sam/webservice

ResourceURI: http://ns.ca.com/2009/07/usm-core/Alert

Selector: MdrProduct

Selector: MdrProdInstance

Selector: MdrElementID

The AlertHandlerImpl.Get() method returns an alert of the type of USM resource that represents the CA SOI alert in USM terms.

Get a List of Alerts

To retrieve a list of alerts, the web services use a combination of WS-Management Enumeration and Pull operations. You can filter the returned list using the WS-Management Filter element to pass a valid XPath expression to limit the number and type of alerts returned.

When you retrieve a list of alerts, the web service returns the AlertID property for each alert in the MdrElementID property. For more information about a specific alert, use the retrieved AlertID value to get an alert.

To get a list of alerts, use the following properties in the request:

Operation: Enumerate & Pull

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/07/usm-core/Alert

Selector: Null

> **NOTE**
> You can also retrieve a list of alerts associated with a service and all of its children CIs. Use the Recursive selector to achieve this task. For example, if you specify MdrElementID=2 and Recursive=True, then if MdrElementID represents a service, the alerts for the service and all of its children CIs are returned in the list.

The AlertIteratorImpl.java program retrieves the list of current alerts from the alert repository context and the CI with which the alert is associated from the Model Repository and creates a collection object that contains the alert list. An enumeration context is returned, which you can use to retrieve the alerts through the Pull operation.

Create an Alert

Use the Create operation to create an alert and associate it with a CI in CA SOI. You define the alert type and USM property values for the new alert in the body of the request.

> **NOTE**
> For information about the required properties to include for an alert, see the USM schema documentation.

To create an alert, use the following properties in the request:

Operation: Create

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/07/usm-core/Alert

Selector: Null

The AlertHandlerImpl.Create() method extracts the alert detail from the body of the SOAP message, and creates the alert in CA SOI. If successful, the CreateResponse includes a unique AlertID in the MdrElementID of the response.

Update an Alert

Use the Put operation to update the writable attributes of an alert. Perform the update by passing in all of the USM properties and their new values in the body of the request.

> **NOTE**
> Refer to the USM schema for a list of USM properties and their appropriate values. For more information, see How to Access the USM Schema Documentation

To update an alert, use the following properties in the request:

Operation: Put

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/07/usm-core/Alert

Selector: Null

The AlertHandlerImpl.Put() method examines the contents of the SOAP header, and if the request contains a Fragment element, only the attributes specified in the Fragment are updated. Otherwise, all writable alert attributes are updated. A PutResponse Operation is returned after successful updates.

Clear an Alert

Use the Delete operation to clear an alert. Pass the USM properties of the alert to delete in the body of the request.

> **NOTE**
> For information about the required properties to include for an alert, see the USM schema documentation.

To clear an alert, use the following properties in the request:

Operation: Delete

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/07/usm-core/Alert

Selector: MdrElementID

The AlertHandlerImpl.Delete() method uses MdrElementID to select the appropriate alert and runs the clear backend function to delete the alert from the Operations Console.

If the alert is successfully cleared, the DeleteResponse operation is returned. Otherwise, a SOAP fault exception is returned.

> **NOTE**
>
> If the clear alert operation fails, it may be due to the 'Respect Underlying MDR Clear Alert Setting' option. This option prevents you from clearing alerts in CA SOI that are not clearable in the source domain manager.

Alert Web Services Examples

The following examples show the SOAP messages of many of the available alert web services.

Example: Get an alert

The following example SOAP message is a Get request to retrieve a specific alert:

```
<env:Envelope>
<env:Header>
   <wsa:To>http://localhost:7090/sam/webservice</wsa:To>
   <wsman:ResourceURI>http://ns.ca.com/2009/07/usm-core/Alert</wsman:ResourceURI>
   <wsman:SelectorSet>
    <wsman:Selector Name="AlertID">32</wsman:Selector>
   </wsman:SelectorSet>
   <wsa:Action>http://schemas.xmlsoap.org/ws/2004/09/transfer/Get</wsa:Action>
   <wsa:ReplyTo>
    <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
   </wsa:ReplyTo>
   <wsa:MessageID>uuid:079e3c5a-7c5e-41a0-b5a8-992238aa55e3</wsa:MessageID>
</env:Header>
<env:Body/>
```

```
</env:Envelope>
```

The SOAP response to this request is as follows:


```
<env:Envelope>
<env:Header>
   <wsa:Action>http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse</wsa:Action>
   <wsa:MessageID>uuid:09af3775-21b1-475e-91ca-4a3b2da059c8</wsa:MessageID>
   <wsa:RelatesTo>uuid:079e3c5a-7c5e-41a0-b5a8-992238aa55e3</wsa:RelatesTo>
   <wsa:To>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:To>
</env:Header>
<env:Body>
   <ns14:Alert>
    <ns14:MdrProduct>CA:00047</ns14:MdrProduct>
    <ns14:MdrProdInstance>AB-XY01-W8</ns14:MdrProdInstance>
    <ns14:MdrElementID>32</ns14:MdrElementID>
    <ns14:UrlParams>http://AB-XY01-W8.ca.com:8080?id=A00002</ns14:UrlParams>
    <ns14:OccurrenceTimestamp>2012-03-05T09:58:59.701+11:00</ns14:OccurrenceTimestamp>
    <ns14:ReportTimestamp>2012-03-05T09:58:59.000+11:00</ns14:ReportTimestamp>
    <ns14:AlertType>Risk</ns14:AlertType>
    <ns14:Severity>Minor</ns14:Severity>
    <ns14:AlertedMdrProduct>CA:00047</ns14:AlertedMdrProduct>
    <ns14:AlertedMdrProdInstance>AB-XY01-W8</ns14:AlertedMdrProdInstance>
    <ns14:AlertedMdrElementID>19</ns14:AlertedMdrElementID>
    <ns14:Summary>UC_Server has an infrastructure alarm..</ns14:Summary>
    <ns14:Message>The Detailed message associated with this alert..</ns14:Message>
    <ns14:IsAcknowledged>false</ns14:IsAcknowledged>
    <ns14:Assignee/>
    <ns14:RelatedIncident/>
    <ssa_instance_id>ComputerSystem:dnsname,ucserver.ca.com:UCServer</ssa_instance_id>
    <ssa_classname>SA_Server</ssa_classname>
    <ssa_class_id>21</ssa_class_id>
    <ssa_connector_id>2</ssa_connector_id>
    <ssa_connector_name>UniversalConnector running on host AB-XY01-W8.ca.com</ssa_connector_name>
    <ssa_ticket_props/>
    <ssa_ticket_id_url/>
    <ssa_ticket_url/>
    <ssa_userattribute_1/>
    <ssa_userattribute_2/>
    <ssa_userattribute_3/>
    <ssa_userattribute_4/>
    <ssa_userattribute_5/>
    <ssa_is_rootcause>true</ssa_is_rootcause>
    <ssa_domain_id>4503599627370496</ssa_domain_id>
    <ssa_queue_id>2</ssa_queue_id>
   </ns14:Alert>
</env:Body>
</env:Envelope>
```


NOTE

The tag structure for the root cause property of service-related alert and CI alert is as follows. In the following example code snippet, false specifies that the CI alerts are not the root cause alerts and 3 implies the ID of the service-related alert that is the root cause alert :

```
<ssa_is_rootcause>false</ssa_is_rootcause>
<ssa_rootcause_alert_id>3</ssa_rootcause_alert_id>
```

Example: Update the ticket number of an alert

The following example SOAP message is a Put request for the RelatedIncident property of the Alert:

```
<env:Envelope>
<env:Header>
   <wsa:ReplyTo>
    <wsa:Address>http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous</wsa:Address>
   </wsa:ReplyTo>
   <wsa:MessageID>uuid:4add8974-25a0-4fc4-84f6-c7f88158f8f7</wsa:MessageID>
   <wsa:To>http://localhost:7090/sam/webservice</wsa:To>
   <wsman:ResourceURI>http://ns.ca.com/2009/07/usm-core/Alert</wsman:ResourceURI>
   <wsman:OperationTimeout>PT30.000S</wsman:OperationTimeout>
   <wsa:Action>http://schemas.xmlsoap.org/ws/2004/09/transfer/Put</wsa:Action>
</env:Header>
<env:Body>
   <ns14:Alert>
    <ns14:MdrProduct>CA:00047</ns14:MdrProduct>
    <ns14:MdrProdInstance>symbe01-5</ns14:MdrProdInstance>
    <ns14:MdrElementID>6</ns14:MdrElementID>
    <ns14:UrlParams>http://symbe01-5.ca.com:8080?id=mdr_id</ns14:UrlParams>
    <ns14:AlertType>Quality</ns14:AlertType>
    <ns14:Severity>Critical</ns14:Severity>
    <ns14:AlertedMdrProduct>CA:00047</ns14:AlertedMdrProduct>
    <ns14:AlertedMdrProdInstance>symbe01-5</ns14:AlertedMdrProdInstance>
    <ns14:AlertedMdrElementID>4</ns14:AlertedMdrElementID>
    <ns14:Summary>UC_Server has a updated client alarm..</ns14:Summary>
    <ns14:Message>UC_Server has a detailed client alarm..</ns14:Message>
    <ns14:IsAcknowledged>false</ns14:IsAcknowledged>
    <ns14:Assignee/>
    <ns14:RelatedIncident>T12345</ns14:RelatedIncident>
    <ssa_ticket_props/>
    <ssa_userattribute_1/>
    <ssa_userattribute_2/>
    <ssa_userattribute_3/>
    <ssa_userattribute_4/>
    <ssa_userattribute_5/>
   </ns14:Alert>
</env:Body>
</env:Envelope>
```

Subscribe to Notifications for Alert Changes

To subscribe to notifications for changes to Alert, use the Notification web services. The Alert resource supports the WS-Eventing functionality that enables a web client to subscribe to notification events when an alert is created, deleted, or updated.

> **NOTE**
> For more information, see Notification Web Services.

Propagation Policy Web Services

Contents

This section provides information about the operations performed in propagation policy web services.

Propagation Policy Web Services Overview

Propagation policy web services use the USM 01-2009 schema to perform operations on propagation policies, which define specific policy instructions for how impact propagates in related CIs in a service.

Propagation policy was known as relationship policy in previous versions of CA SOI. The following propagation types require propagation policy:

- Custom (formerly DependsOn)
- Operative (formerly Requires)

The propagation policy web services let you interact with Custom propagation policy assigned to a relationship and propagation between CIs. Use the term DependsOn in web service requests to refer to Custom propagation policy.

> **NOTE**
> The web services cannot interact with Operative propagation policy.

Use the following endpoint URI when invoking the propagation policy web services resource:

```
http://ns.ca.com/2009/01/usm-data/RelationshipPolicy
```

The following WSDL file outlines the available operations:

```
http://<samanager>:<port>/sam/webservice/wsdls/usm2.wsdl
```

Access the USM 01-2009 schema as follows:

```
http://<samanager>:<port>/sam/webservice/schemas/usm2.xsd
```

> **NOTE**
> For more information about propagation policy, see Create Propagation Policy and Assign Types

Get a Propagation Policy

Use the Get request to retrieve a propagation policy associated with a custom propagation type. View the propagation policy in the Service Modeler on the Policies tab.

The following selectors are required to identify a unique instance of a propagation policy:

- **ASBOID.id**
  Uniquely identifies the propagation policy using the Policy ID value. Derive this value using an Enumerate operation.
- **ASBOID.source**
  Defines the DomainID of the CA SOI model repository. This value is constant for the SA Manager. Derive the value using an Enumerate operation.

To get a propagation policy, use the following properties in the request:

Operation: Get

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/01/usm-data/RelationshipPolicy

Selector: ASBOID.id

Selector: ASBOID.source

Get a List of Propagation Policies

To retrieve a list of propagation policies for a service, the web services use a combination of WS-Management Enumeration and Pull operations. You can filter the returned list using the WS-Management Filter element to pass a valid XPath Expression to limit the number and type of propagation policies returned.

Use the item_name selector to filter the collection by service. Format the selector as follows:

```
SA_Service:servicename
```

- **servicename**
  Defines the name of the service.

To get a list of propagation policies, use the following properties in the request:

Operation: Enumerate & Pull

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/01/usm-data/RelationshipPolicy

Selector: item_name

Create a Propagation Policy

Use the Create operation to create a propagation policy between CIs in CA SOI that belong to an existing service. Include the following information in the body of the request:

- CI ID for the service in which the propagation exists
- CI ID for the source CI
- CI ID for the target CI or CIs
- Policy properties, such as impact and thresholds

The associated propagation type must be custom (referred to as DependsOn in the web service request).

> **NOTE**
> For information about the required properties to include for a propagation policy, see the USM 01-2009 schema. For more information about property names and formatting, see Propagation Policy Web Services Examples.

To create a propagation policy, use the following properties in the request:

Operation: Create

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/01/usm-data/RelationshipPolicy

Selector: null

Update a Propagation Policy

Use the Put operation to update the writable attributes of a propagation policy. You can update all writable attributes or only certain attributes that you specify in a Fragment statement. To identify the unique propagation policy, pass the ASBOID.id and ASBOID.source propagation policy values as selectors.

You can update the following propagation policy attributes:

- item_description
- sam_policy_type
- sam_policy_threshold_1 (Rule 1 threshold)
- sam_policy_threshold_2 (Rule 2 threshold)
- sam_policy_threshold_3 (Rule 3 threshold)
- sam_policy_threshold_4 (Rule 4 threshold)
- sam_policy_action_1 (Rule 1 impact)
- sam_policy_action_2 (Rule 2 impact)
- sam_policy_action_3 (Rule 3 impact)
- sam_policy_action_4 (Rule 4 impact)
- sam_policy_bnode_id (CIs that the policy applies to)

The valid sam_policy_type values are as follows:

- **Any**
  Sets the impact of the parent item when any CIs associated with the policy have the impact specified in the rule.
- **All**
  Sets the impact of the parent item when all CIs have the impact specified in the rule.
- **Percentage**
  Sets the impact of the parent item based on a percentage of CIs that have the impact specified in the rule. For this type of policy the “sam_policy_threshold_X” values have a four-digit format PPTT with the first two digits representing the percentage of CIs and the last two the impact threshold. A value of 100TT is allowed. For all other policy types the “sam_policy_threshold_X” value has to be a two-digit number from 1-40 representing the impact threshold. See the example on this page for an example of the four digit format.
- **Average**
  Sets the impact of the parent item based on the average impact values of CIs associated with the policy.

To update a propagation policy, use the following properties in the request:

Operation: Put

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/01/usm-data/RelationshipPolicy

Selector: ASBOID.id

Selector: ASBOID.source

Fragment: any attributes

Delete a Propagation Policy

Use the Delete operation to delete a propagation policy. Use the propagation policy ASBOID.id and ASBOID.source values as selectors.

To delete a propagation policy, use the following properties in the request:

Operation: Delete

Endpoint *:* http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/01/usm-data/RelationshipPolicy

Selector: ASBOID.id

Selector: ASBOID.source

Propagation Policy Web Services Examples

The following examples show the SOAP messages of many of the available propagation policy web services.

Example: Create a custom propagation policy

The following example SOAP message is a Create request to create a custom propagation policy:

```
<env:Header>
   <wsa:ReplyTo xmlns:ns13="http://ns.ca.com/2009/01/usm-data">
   <wsa:Address env:mustUnderstand="true">
    http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
   </wsa:Address>
   </wsa:ReplyTo>
   <wsa:MessageID xmlns:ns13="http://ns.ca.com/2009/01/usm-data"
   env:mustUnderstand="true">
    uuid:e5d2bbd4-04dc-42d0-ab7a-2bd1692c87a0
   </wsa:MessageID>
   <wsa:To xmlns:ns13="http://ns.ca.com/2009/01/usm-data" env:mustUnderstand="true">
    http://localhost:7090/sam/webservice
   </wsa:To>
   <wsman:ResourceURI xmlns:ns13="http://ns.ca.com/2009/01/usm-data">
    http://ns.ca.com/2009/01/usm-data/RelationshipPolicy
   </wsman:ResourceURI>
   <wsman:OperationTimeout xmlns:ns13="http://ns.ca.com/2009/01/usm-data" >
    PT30.000S
   </wsman:OperationTimeout>
   <wsa:Action xmlns:ns13="http://ns.ca.com/2009/01/usm-data"
   env:mustUnderstand="true">
    http://schemas.xmlsoap.org/ws/2004/09/transfer/Create
   </wsa:Action>
</env:Header>
<env:Body>
   <ns11:RelationshipPolicy xmlns:ns10="http://www.w3.org/2003/05/soap-envelope" xmlns:ns11="http://
ns.ca.com/2009/01/usm-data" xmlns:ns2="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:ns3="http://
schemas.xmlsoap.org/ws/2004/08/eventing"
   xmlns:ns4="http://schemas.xmlsoap.org/ws/2004/09/enumeration" xmlns:ns5="http://schemas.xmlsoap.org/
ws/2004/09/transfer" xmlns:ns6="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd"
   xmlns:ns7="http://schemas.xmlsoap.org/ws/2004/09/mex" xmlns:ns8="http://schemas.wiseman.dev.java.net/
metadata/messagetypes" xmlns:ns9="http://schemas.sam.ca.com/webservice/1/alarm.xsd">
            <sam_policy_type>Percentage</sam_policy_type>
            <sam_policy_service_id>22</sam_policy_service_id>
            <sam_policy_anode_id>22</sam_policy_anode_id>
            <sam_policy_action_1>2</sam_policy_action_1>
            <sam_policy_action_2>3</sam_policy_action_2>
            <sam_policy_threshold_1>3015</sam_policy_threshold_1>
            <sam_policy_threshold_2>6035</sam_policy_threshold_2>
            <sam_usm_reltype_id>105</sam_usm_reltype_id>
            <sam_policy_bnode_id>23</sam_policy_bnode_id>
            <ns11:item_name>SAM2818_2</ns11:item_name>
            <ns11:item_description>SAM2818_2 Policy</ns11:item_description>
   </ns11:RelationshipPolicy>
</env:Body>
</env:Envelope>
```

The bold syntax shows the properties defined for the custom propagation policy. The SOAP response to this request is as follows:

```
<env:Header>
   <wsa:Action env:mustUnderstand="true"
   xmlns:ns13="http://ns.ca.com/2009/01/usm-data" >
    http://schemas.xmlsoap.org/ws/2004/09/transfer/CreateResponse
   </wsa:Action>
   <wsa:MessageID env:mustUnderstand="true"
   xmlns:ns13="http://ns.ca.com/2009/01/usm-data" >
    uuid:449e3af7-12e2-4f91-a3f4-360a4dffe484
   </wsa:MessageID>
   <wsa:RelatesTo xmlns:ns13="http://ns.ca.com/2009/01/usm-data">
    uuid:e5d2bbd4-04dc-42d0-ab7a-2bd1692c87a0
   </wsa:RelatesTo>
   <wsa:To env:mustUnderstand="true" xmlns:ns13="http://ns.ca.com/2009/01/usm-data" >
    http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
   </wsa:To>
</env:Header>
   <env:Body>
   <wxf:ResourceCreated xmlns:ns13="http://ns.ca.com/2009/01/usm-data" >
   <wsa:Address env:mustUnderstand="true">
    http://localhost:7090/sam/webservice/
   </wsa:Address>
   <wsa:ReferenceParameters>
   <wsman:ResourceURI>
    http://ns.ca.com/2009/01/usm-data/RelationshipPolicy
   </wsman:ResourceURI>
   <wsman:SelectorSet>
      <wsman:Selector Name="ASBOID.id">8</wsman:Selector>
      <wsman:Selector Name="ASBOID.source">4503599627370496</wsman:Selector>
   </wsman:SelectorSet>
   </wsa:ReferenceParameters>
   </wxf:ResourceCreated>
</env:Body>
</env:Envelope>
```

The bold syntax shows the returned selector properties for the created custom propagation policy.

Example: Delete a propagation policy

The following example SOAP message is a Delete request to delete a propagation policy:

```
<env:Header>
   <wsa:ReplyTo xmlns:ns13="http://ns.ca.com/2009/01/usm-data">
   <wsa:Address env:mustUnderstand="true">
    http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
   </wsa:Address>
   </wsa:ReplyTo>
   <wsa:MessageID xmlns:ns13="http://ns.ca.com/2009/01/usm-data"
   env:mustUnderstand="true">
    uuid:202fc0d4-7bf6-4e28-87cf-1a11afef97db
   </wsa:MessageID>
   <wsa:To xmlns:ns13="http://ns.ca.com/2009/01/usm-data" env:mustUnderstand="true">
   http://localhost:7090/sam/webservice</wsa:To>
```

```
   <wsman:ResourceURI xmlns:ns13="http://ns.ca.com/2009/01/usm-data">
    http://ns.ca.com/2009/01/usm-data/RelationshipPolicy
   </wsman:ResourceURI>
   <wsman:OperationTimeout xmlns:ns13="http://ns.ca.com/2009/01/usm-data">
    PT30.000S
   </wsman:OperationTimeout>
   <wsman:SelectorSet xmlns:ns13="http://ns.ca.com/2009/01/usm-data">
      <wsman:Selector Name="ASBOID.source">4503599627370496</wsman:Selector>
      <wsman:Selector Name="ASBOID.id">8</wsman:Selector>
   </wsman:SelectorSet>
   <wsa:Action xmlns:ns13="http://ns.ca.com/2009/01/usm-data"
   env:mustUnderstand="true">
    http://schemas.xmlsoap.org/ws/2004/09/transfer/Delete
   </wsa:Action>
</env:Header>
<env:Body/>
</env:Envelope>
```

The bold syntax shows the selectors that uniquely identify the propagation policy. The SOAP response to this request is as follows:

```
<env:Header>
   <wsa:Action env:mustUnderstand="true"
   xmlns:ns13="http://ns.ca.com/2009/01/usm-data" >
    http://schemas.xmlsoap.org/ws/2004/09/transfer/DeleteResponse
   </wsa:Action>
   <wsa:MessageID env:mustUnderstand="true"
   xmlns:ns13="http://ns.ca.com/2009/01/usm-data" >
    uuid:baa81209-3bc6-4e67-aed4-2050a89b8279
   </wsa:MessageID>
   <wsa:RelatesTo xmlns:ns13="http://ns.ca.com/2009/01/usm-data">
   uuid:202fc0d4-7bf6-4e28-87cf-1a11afef97db</wsa:RelatesTo>
   <wsa:To env:mustUnderstand="true"
   xmlns:ns13="http://ns.ca.com/2009/01/usm-data" >
    http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
   </wsa:To>
</env:Header>
<env:Body/>
</env:Envelope>
```

Escalation Policy Web Services

Contents

This section provides information about the operations performed in escalation policy web services.

> **NOTE**
> WS-MAN web services should be considered obsolete. We recommend using the Escalation Policy REST web services instead.

Escalation Policy Web Services Overview

Escalation policy web services use the USM 01-2009 schema to perform operations on escalation policies, which define specific policy instructions for when to escalate alerts. Escalation policies can be global or service-specific.

Use the following endpoint URI when invoking the escalation policy web services resource:

```
http://ns.ca.com/2009/01/usm-data/EscalationPolicy
```

The following WSDL file outlines the available operations:

```
http://<samanager>:<port>/sam/webservice/wsdls/usm2.wsdl
```

Access the USM 01-2009 schema as follows:

```
http://<samanager>:<port>/sam/webservice/schemas/usm2.xsd
```

> **NOTE**
> For more information about escalation policy, see How to Create Escalation Policy.

Get an Escalation Policy

Use the Get request to retrieve an escalation policy. The following selectors are required to identify a unique instance of an escalation policy:

- **ASBOID.id**
  Uniquely identifies the escalation policy using the Policy ID value. Derive this value using an Enumerate operation.
- **ASBOID.source**
  Defines the DomainID of the CA SOI model repository. This value is constant for the SA Manager. Derive the value using an Enumerate operation.

To get an escalation policy, use the following properties in the request:

Operation: Get

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/01/usm-data/EscalationPolicy

Selector: ASBOID.id

Selector: ASBOID.source

Get a List of Escalation Policies

To retrieve a list of escalation policies, the web services use a combination of WS-Management Enumeration and Pull operations. You can filter the returned list using the WS-Management Filter element to pass a valid XPath Expression to limit the number and type of escalation policies returned.

To get a list of escalation policies, use the following properties in the request:

Operation: Enumerate & Pull

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/01/usm-data/EscalationPolicy

Selector: null

Create an Escalation Policy

Use the Create operation to create an escalation policy. You define escalation policy properties in the body of the request, such as the following:

- Policy type (global or non-global)
- The types of alerts to which the policy applies

NOTE
Global policy type specifies that the policy applies to all alerts. Non-global type specifies that the policy applies to alerts of an assigned service or alert queue. When you create a queue, you can associate it with an existing escalation policy by using the attribute <ssa_escalation_policy_id_1>.

- Strings that indicate policy rules

NOTE
For information about the required properties to include for an escalation policy, see the USM 01-2009 schema. For more information about property names and formatting, see Escalation Policy Web Services Examples.

To create an escalation policy, use the following properties in the request:

Operation: Create

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/01/usm-data/EscalationPolicy

Selector: null

Update an Escalation Policy

Use the Put operation to update the writable attributes of an escalation policy. You can update all writable attributes or only certain attributes that you specify in a Fragment statement. To identify the unique escalation policy, pass the ASBOID.id and ASBOID.source escalation policy values as selectors.

You can update the following escalation policy attributes:

- sam_policy_enabled (Enable)
- sam_is_global (Global or Non-global)

NOTE
You can create an escalation policy as a non-global policy and add the attribute <sam_is_local_service_name> to identify the service.

- sam_root_cause_alarm
- sam_symptom_alarm
- sam_service_alarm
- sam_infrastructure_alarm
- sam_maintenance_alarm
- sam_service_maintenance_alarm

To update an escalation policy, use the following properties in the request:

Operation: Put

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/01/usm-data/EscalationPolicy

Selector: ASBOID.id

Selector: ASBOID.source

Fragment: any property

Delete an Escalation Policy

Use the Delete operation to delete an escalation policy. Use the escalation policy ASBOID.id and ASBOID.source values as selectors.

To delete an escalation policy, use the following properties in the request::

Operation: Delete

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/01/usm-data/EscalationPolicy

Selector: ASBOID.id

Selector: ASBOID.source

Escalation Policy Web Services Examples

The following examples show the SOAP messages of many of the available escalation policy web services.

Example: Create a global escalation policy

The following example SOAP message is a Create request to create a global escalation policy:

```
<env:Header>
   <wsa:ReplyTo xmlns:ns13="http://ns.ca.com/2009/01/usm-data">
   <wsa:Address env:mustUnderstand="true">
    http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
   </wsa:Address>
   </wsa:ReplyTo>
   <wsa:MessageID xmlns:ns13="http://ns.ca.com/2009/01/usm-data"
   env:mustUnderstand="true">
    uuid:51f6d9ca-767f-4be7-bd3e-d1f13cdd6759
   </wsa:MessageID>
   <wsa:To xmlns:ns13="http://ns.ca.com/2009/01/usm-data"  env:mustUnderstand="true">
    http://localhost:7090/sam/webservice
   </wsa:To>
   <wsman:ResourceURI xmlns:ns13="http://ns.ca.com/2009/01/usm-data" >
   http://ns.ca.com/2009/01/usm-data/EscalationPolicy
   </wsman:ResourceURI>
   <wsman:OperationTimeout xmlns:ns13="http://ns.ca.com/2009/01/usm-data" >
    PT30.000S
   </wsman:OperationTimeout>
   <wsa:Action xmlns:ns13="http://ns.ca.com/2009/01/usm-data"
   env:mustUnderstand="true">
    http://schemas.xmlsoap.org/ws/2004/09/transfer/Create
   </wsa:Action>
</env:Header>
<env:Body>
   <ns11:EscalationPolicy xmlns:ns10="http://www.w3.org/2003/05/soap-envelope" xmlns:ns11="http://
ns.ca.com/2009/01/usm-data"
   xmlns:ns12="http://ns.ca.com/2009/07/usm-core" xmlns:ns2="http://schemas.xmlsoap.org/ws/2004/08/addressing"
 xmlns:ns3="http://schemas.xmlsoap.org/ws/2004/08/eventing"
   xmlns:ns4="http://schemas.xmlsoap.org/ws/2004/09/enumeration" xmlns:ns5="http://schemas.xmlsoap.org/
ws/2004/09/transfer" xmlns:ns6="http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd" xmlns:ns7="http://
schemas.xmlsoap.org/ws/2004/09/mex" xmlns:ns8="http://schemas.wiseman.dev.java.net/metadata/messagetypes"
 xmlns:ns9="http://schemas.sam.ca.com/webservice/1/alarm.xsd">
       <sam_is_global>true</sam_is_global>
       <sam_policy_enabled>true</sam_policy_enabled>
       <sam_root_cause_alarm>true</sam_root_cause_alarm>
       <sam_symptom_alarm>true</sam_symptom_alarm>
```

```
        <sam_service_alarm>true</sam_service_alarm>
        <sam_infrastructure_alarm>true</sam_infrastructure_alarm>
        <sam_maintenance_mode>true</sam_maintenance_mode>
        <sam_service_maintenance_mode>false</sam_service_maintenance_mode>
        <sam_schedule_type>0</sam_schedule_type>
        <sam_calendar_id>0</sam_calendar_id>
        <sam_xml_rule_string>&#38;lt;esc-policy&#38;gt;&#38;lt;/esc-policy&#38;gt;
        </sam_xml_rule_string>
        <sam_drl_rule_string>package com.ca.sam.manager.rules
         import com.ca.sam.manager.rules.AlarmObject;
         rule "d68f435d-9a7e-4926-b383-d2e4676920ae"
            when
            $alarm : AlarmObject()</sam_drl_rule_string>
        <ns11:item_name>Escalation_Policy_eleven</ns11:item_name>
        <ns11:item_description>Policy created by the
        webservice...</ns11:item_description>
    </ns11:EscalationPolicy>
 </env:Body>
 </env:Envelope>
```

The bold syntax defines the following properties for the escalation policy:

- It is global and enabled
- It includes root cause, symptom, service, and infrastructure alerts
- It includes the rules defined in the sam_xml_rule_string and sam_drl_rule_string properties

The SOAP response to this request is as follows:

```
 <env:Header>
    <wsa:Action env:mustUnderstand="true"
    xmlns:ns13="http://ns.ca.com/2009/01/usm-data" >
     http://schemas.xmlsoap.org/ws/2004/09/transfer/CreateResponse
    </wsa:Action>
    <wsa:MessageID env:mustUnderstand="true"
    xmlns:ns13="http://ns.ca.com/2009/01/usm-data" >
     uuid:be232d8e-d755-4627-9365-643dbe47cf85
    </wsa:MessageID>
    <wsa:RelatesTo xmlns:ns13="http://ns.ca.com/2009/01/usm-data">
     uuid:51f6d9ca-767f-4be7-bd3e-d1f13cdd6759
    </wsa:RelatesTo>
    <wsa:To env:mustUnderstand="true" xmlns:ns13="http://ns.ca.com/2009/01/usm-data" >
     http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
    </wsa:To>
 </env:Header>
    <env:Body>
       <wxf:ResourceCreated xmlns:ns13="http://ns.ca.com/2009/01/usm-data">
       <wsa:Address
       env:mustUnderstand="true">http://localhost:8888/sam/webservice/
       </wsa:Address>
       <wsa:ReferenceParameters>
       <wsman:ResourceURI>
        http://ns.ca.com/2009/01/usm-data/EscalationPolicy
       </wsman:ResourceURI>
       <wsman:SelectorSet>
```

```
            <wsman:Selector Name="ASBOID.id">32</wsman:Selector>
            <wsman:Selector Name="ASBOID.source">4503599627370496</wsman:Selector>
        </wsman:SelectorSet>
        </wsa:ReferenceParameters>
        </wxf:ResourceCreated>
    </env:Body>
</env:Envelope>
```

The bold syntax shows the returned selector properties for the created escalation policy.

Example: Get an escalation policy

The following example SOAP message is a Get request to retrieve a specific escalation policy:

```
<env:Header>
   <wsa:To xmlns:ns13="http://ns.ca.com/2009/01/usm-data" env:mustUnderstand="true">
    http://localhost:7090/sam/webservice
   </wsa:To>
   <wsman:ResourceURI xmlns:ns13="http://ns.ca.com/2009/01/usm-data">
    http://ns.ca.com/2009/01/usm-data/EscalationPolicy
   </wsman:ResourceURI>
   <wsman:SelectorSet xmlns:ns13="http://ns.ca.com/2009/01/usm-data">
      <wsman:Selector Name="ASBOID.source">0</wsman:Selector>
      <wsman:Selector Name="ASBOID.id">27</wsman:Selector>
   </wsman:SelectorSet>
   <wsa:Action xmlns:ns13="http://ns.ca.com/2009/01/usm-data"
   env:mustUnderstand="true">
    http://schemas.xmlsoap.org/ws/2004/09/transfer/Get
   </wsa:Action>
   <wsa:ReplyTo xmlns:ns13="http://ns.ca.com/2009/01/usm-data">
   <wsa:Address env:mustUnderstand="true">
    http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
   </wsa:Address>
   </wsa:ReplyTo>
   <wsa:MessageID xmlns:ns13="http://ns.ca.com/2009/01/usm-data"
   env:mustUnderstand="true">
   uuid:78372ad8-46e7-4d27-b632-7e2de827f29a
   </wsa:MessageID>
</env:Header>
<env:Body/>
</env:Envelope>
```

The request retrieves the escalation policy with the ASBOID values in the bold SelectorSet syntax. The SOAP response to this request is as follows:

```
<env:Header>
   <wsa:Action env:mustUnderstand="true"
   xmlns:ns13="http://ns.ca.com/2009/01/usm-data" >
    http://schemas.xmlsoap.org/ws/2004/09/transfer/GetResponse
   </wsa:Action>
   <wsa:MessageID env:mustUnderstand="true"
   xmlns:ns13="http://ns.ca.com/2009/01/usm-data" >
    uuid:a608934d-17a5-4e7a-b5eb-d983bc6e1d8d
   </wsa:MessageID>
   <wsa:RelatesTo xmlns:ns13="http://ns.ca.com/2009/01/usm-data">
```

```
    uuid:78372ad8-46e7-4d27-b632-7e2de827f29a
   </wsa:RelatesTo>
   <wsa:To env:mustUnderstand="true" xmlns:ns13="http://ns.ca.com/2009/01/usm-data" >
    http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
   </wsa:To>
   </env:Header>
   <env:Body>
      <ns13:EscalationPolicy xmlns:ns13="http://ns.ca.com/2009/01/usm-data" >
         <sam_unique_id>4edb0241-f08a-416d-a7ab-16e32c920651</sam_unique_id>
         <sam_compiled_time>2010-03-16 09:00:01.27</sam_compiled_time>
         <sam_policy_enabled>true</sam_policy_enabled>
         <sam_root_cause_alarm>false</sam_root_cause_alarm>
         <sam_symptom_alarm>false</sam_symptom_alarm>
         <sam_service_alarm>true</sam_service_alarm>
         <sam_infrastructure_alarm>true</sam_infrastructure_alarm>
         <sam_maintenance_mode>true</sam_maintenance_mode>
         <sam_is_global>true</sam_is_global>
         <sam_service_maintenance_mode>false</sam_service_maintenance_mode>
         <sam_schedule_type>2</sam_schedule_type>
         <sam_escalation_schedule_id>0x11000000000001</sam_escalation_schedule_id>
         <sam_escalation_schedule_desc>Daily</sam_escalation_schedule_desc>
         <sam_calendar_id>0</sam_calendar_id>
         <sam_xml_rule_string><![CDATA[&#60;esc-policy&#62;&#60;time-filter&#62;&#60;or&#62;&#60;greater-
than&#62;&#60;attribute id=&#34;0x20027&#34;&#62;&#60;value&#62;30&#60;/value&#62;&#60;/attribute&#62;&#60;/
greater-than&#62;&#60;/or&#62;&#60;/time-filter&#62;&#60;attr-filter&#62;&#60;and&#62;&#60;equals-
ignore-case&#62;&#60;attribute id=&#34;0x11f57&#34;&#62;&#60;value&#62;symbe01&#60;/value&#62;&#60;/
attribute&#62;&#60;/equals-ignore-case&#62;&#60;/and&#62;&#60;/attr-filter&#62;&#60;/esc-policy&#62;]] >
         </sam_xml_rule_string>
         <sam_drl_rule_string>package com.ca.sam.manager.rules
   import com.ca.sam.manager.rules.AlarmObject;
   rule "4edb0241-f08a-416d-a7ab-16e32c920651"
      when
         $alarm : AlarmObject(</sam_drl_rule_string>
         <USMID>0:27</USMID>
         <ASBOID>
            <source>0</source>
            <id>27</id>
         </ASBOID>
         <ns13:item_name>global_3</ns13:item_name>
         <ns13:item_description/>
         <ns13:item_creation_date>2010-03-09T14:28:45.693+11:00
         </ns13:item_creation_date>
         <ns13:item_creation_user>Web Service</ns13:item_creation_user>
   </ns13:EscalationPolicy>
</env:Body>
</env:Envelope>
```


The bold syntax shows the details of the returned escalation policy.

Escalation Action Web Services

Contents

This section provides information about the operations performed in escalation action web services.

> **NOTE**
> WS-MAN web services should be considered obsolete. We recommend using the Escalation Policy Action REST web services instead.

Escalation Action Web Services Overview

Escalation action web services use the USM 01-2009 schema to perform operations on escalation actions, which define specific actions to perform when associated escalation policy criteria are met. Examples of available actions include the following:

- Send an email
- Create a help desk ticket
- Run a command
- Create a help desk announcement
- Run the CA Process Automation process
- Clear an alert

Use the following endpoint URI when invoking the escalation action web services resource:

```
http://ns.ca.com/2009/01/usm-data/EscalationpolicyAction
```

The following WSDL file outlines the available operations:

```
http://<samanager>:<port>/sam/webservice/wsdls/usm2.wsdl
```

Access the USM 01-2009 schema as follows:

```
http://<samanager>:<port>/sam/webservice/schemas/usm2.xsd
```

> **NOTE**
> For more information about escalation actions, see How to Create Escalation Policy.

Get an Escalation Action

Use the Get request to retrieve an escalation action. The following selectors are required to identify a unique instance of an escalation action:

- **ASBOID.id**
  Uniquely identifies the escalation action using the Action ID value. Derive this value using an Enumerate operation.
- **ASBOID.source**
  Defines the DomainID of the CA SOI model repository. This value is constant for the SA Manager. Derive the value using an Enumerate operation.

To get an escalation action, use the following properties in the request:

Operation: Get

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/01/usm-data/EscalationpolicyAction

Selector: ASBOID.id

Selector: ASBOID.source

Get a List of Escalation Actions

To retrieve a list of escalation actions, the web services use a combination of WS-Management Enumeration and Pull operations. You can filter the returned list using the WS-Management Filter element to pass a valid XPath Expression to limit the number and type of escalation actions returned.

To get a list of escalation actions, use the following properties in the request:

Operation: Enumerate & Pull

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/01/usm-data/EscalationpolicyAction

Selector: null

You can also retrieve a list of escalation actions based on the action name and action type. You can achieve this by using the WS-Management Filter parameter in the Enumerate operation as explained in the following two examples:

- The first example SOAP message snippet explains how you can get a list of escalation types based on the action type. This example limits the list to those action types that are equal to 1, which represents *Tickets*:
```
<env:Body xmlns:EscalationpolicyAction="http://ns.ca.com/2009/01/usm-data">
<wsen:Enumerate xmlns:ns11="http://ns.ca.com/2009/01/usm-data" >
<wsman:Filter Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">/
EscalationpolicyAction:EscalationpolicyAction[sam_action_type='1']</wsman:Filter>
<wsman:EnumerationMode>EnumerateObjectAndEPR</wsman:EnumerationMode>
</wsen:Enumerate>
</env:Body>
```
- The second example SOAP message snippet explains how you can get a list of escalation types based on the action name. This example limits the list to those escalation actions that have a name starting with *Tick*:
```
<env:Body xmlns:EscalationpolicyAction="http://ns.ca.com/2009/01/usm-data">
<wsen:Enumerate xmlns:ns11="http://ns.ca.com/2009/01/usm-data" >
<wsman:Filter Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">/
EscalationpolicyAction:EscalationpolicyAction[starts-with(EscalationpolicyAction:item_name,'Tick')]</
wsman:Filter>
<wsman:EnumerationMode>EnumerateObjectAndEPR</wsman:EnumerationMode>
</wsen:Enumerate>
</env:Body>
```

Create an Escalation Action

Use the Create request to create an escalation action. You define escalation action properties in the body of the request, such as the following:

- Action type (0-5)
- Action data, such as an email address

> **NOTE**
> For more information about property names and formatting, see Escalation Action Web Services Examples.

The valid action type properties are as follows:

- **0**
  Corresponds to the Send Email escalation action type.
- **1**
  Corresponds to the Create Ticket escalation action type.
- **2**

Corresponds to the Execute Command escalation action type.

- **3**
  Corresponds to the Create Announcement escalation action type.
- **4**
  Corresponds to the Execute Automated Process escalation action type.
- **5**
  Corresponds to the Clear Alert escalation action type.

To create an escalation action, use the following properties in the request:

Operation: Create

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/01/usm-data/EscalationpolicyAction

Selector: null

Delete an Escalation Action

Use the Delete operation to delete an escalation action. Use the escalation action ASBOID.id and ASBOID.source values as selectors.

To delete an escalation action, use the following properties in the request:

Operation: Delete

Endpoint: http://<samanager>:<port>/sam/webservice

Resource: http://ns.ca.com/2009/01/usm-data/EscalationpolicyAction

Selector: ASBOID.id

Selector: ASBOID.source

Escalation Action Web Services Examples

The following examples show the SOAP messages of many of the available escalation action web services.

Example: Create an email escalation action

The following example SOAP message is a Create request to create an escalation action that sends an email when the associated escalation policy criteria are met:

```
<env:Header>
   <wsa:ReplyTo xmlns:ns11="http://schemas.sam.ca.com/webservice/1/alarm.xsd" xmlns:ns13="http://
ns.ca.com/2009/01/usm-data"
   xmlns:ns14="http://ns.ca.com/2009/07/usm-core">
   <wsa:Address env:mustUnderstand="true">
    http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
   </wsa:Address>
   </wsa:ReplyTo>
   <wsa:MessageID xmlns:ns11="http://schemas.sam.ca.com/webservice/1/alarm.xsd" xmlns:ns13="http://
ns.ca.com/2009/01/usm-data"
   xmlns:ns14="http://ns.ca.com/2009/07/usm-core" env:mustUnderstand="true">
   uuid:4a909925-71c2-4317-82e4-b27c0b4f89d0
   </wsa:MessageID>
   <wsa:To xmlns:ns11="http://schemas.sam.ca.com/webservice/1/alarm.xsd" xmlns:ns13="http://ns.ca.com/2009/01/
usm-data"
   xmlns:ns14="http://ns.ca.com/2009/07/usm-core" env:mustUnderstand="true">
```

```
   http://localhost:7090/sam/webservice
   </wsa:To>
   <wsman:ResourceURI xmlns:ns11="http://schemas.sam.ca.com/webservice/1/alarm.xsd" xmlns:ns13="http://
ns.ca.com/2009/01/usm-data"
   xmlns:ns14="http://ns.ca.com/2009/07/usm-core">
   http://ns.ca.com/2009/01/usm-data/EscalationpolicyAction
   </wsman:ResourceURI>
   <wsman:OperationTimeout
   xmlns:ns11="http://schemas.sam.ca.com/webservice/1/alarm.xsd" xmlns:ns13="http://ns.ca.com/2009/01/usm-
data"
   xmlns:ns14="http://ns.ca.com/2009/07/usm-core">
   PT30.000S
   </wsman:OperationTimeout>
   <wsa:Action xmlns:ns11="http://schemas.sam.ca.com/webservice/1/alarm.xsd" xmlns:ns13="http://
ns.ca.com/2009/01/usm-data"
   xmlns:ns14="http://ns.ca.com/2009/07/usm-core" env:mustUnderstand="true">
   http://schemas.xmlsoap.org/ws/2004/09/transfer/Create
</wsa:Action>
</env:Header>
<env:Body>
   <ns11:EscalationpolicyAction xmlns:ns10="http://www.w3.org/2003/05/soap-envelope" xmlns:ns11="http://
ns.ca.com/2009/01/usm-data"
   xmlns:ns12="http://ns.ca.com/2009/07/usm-core" xmlns:ns2="http://schemas.xmlsoap.org/ws/2004/08/addressing"
   xmlns:ns3="http://schemas.xmlsoap.org/ws/2004/08/eventing" xmlns:ns4="http://schemas.xmlsoap.org/
ws/2004/09/enumeration"
   xmlns:ns5="http://schemas.xmlsoap.org/ws/2004/09/transfer" xmlns:ns6="http://schemas.dmtf.org/
wbem/wsman/1/wsman.xsd" xmlns:ns7="http://schemas.xmlsoap.org/ws/2004/09/mex" xmlns:ns8="http://
schemas.wiseman.dev.java.net/metadata/messagetypes" xmlns:ns9="http://schemas.sam.ca.com/webservice/1/
alarm.xsd">
      <sam_action_type>0</sam_action_type>
      <sam_action_data>emailaddress@ca.com</sam_action_data>
      <sam_action_subject>new web service subject</sam_action_subject>
      <sam_action_msg>the message of the email</sam_action_msg>
      <ns11:item_name>MyEmailAction</ns11:item_name>
      <ns11:item_description>Escalation Action created via WS
      </ns11:item_description>
   </ns11:EscalationpolicyAction>
</env:Body>
</env:Envelope>
```

The bold syntax defines the following properties for the escalation action:

- A type of 0 indicates an email escalation action
- The action sends an email to the address emailaddress@ca.com with a subject of 'new web service subject'.

The SOAP response to this request is as follows:

```
<env:Header>
   <wsa:Action env:mustUnderstand="true"
   xmlns:ns11="http://schemas.sam.ca.com/webservice/1/alarm.xsd" xmlns:ns13="http://ns.ca.com/2009/01/usm-
data"
   xmlns:ns14="http://ns.ca.com/2009/07/usm-core">
   http://schemas.xmlsoap.org/ws/2004/09/transfer/CreateResponse
   </wsa:Action>
```

```
   <wsa:MessageID env:mustUnderstand="true"
   xmlns:ns11="http://schemas.sam.ca.com/webservice/1/alarm.xsd" xmlns:ns13="http://ns.ca.com/2009/01/usm-
data"
   xmlns:ns14="http://ns.ca.com/2009/07/usm-core">
   uuid:ec2a8c6a-4cd1-4b89-8380-05ffc9dbd488
   </wsa:MessageID>
   <wsa:RelatesTo xmlns:ns11="http://schemas.sam.ca.com/webservice/1/alarm.xsd" xmlns:ns13="http://
ns.ca.com/2009/01/usm-data"
   xmlns:ns14="http://ns.ca.com/2009/07/usm-core">
   uuid:4a909925-71c2-4317-82e4-b27c0b4f89d0
   </wsa:RelatesTo>
   <wsa:To env:mustUnderstand="true" x
   mlns:ns11="http://schemas.sam.ca.com/webservice/1/alarm.xsd" xmlns:ns13="http://ns.ca.com/2009/01/usm-
data"
   xmlns:ns14="http://ns.ca.com/2009/07/usm-core">
   http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
   </wsa:To>
</env:Header>
<env:Body>
   <wxf:ResourceCreated
   xmlns:ns11="http://schemas.sam.ca.com/webservice/1/alarm.xsd" xmlns:ns13="http://ns.ca.com/2009/01/usm-
data"
   xmlns:ns14="http://ns.ca.com/2009/07/usm-core">
   <wsa:Address
   env:mustUnderstand="true">http://localhost:8888/sam/webservice/</wsa:Address>
   <wsa:ReferenceParameters>
   <wsman:ResourceURI>
   http://ns.ca.com/2009/01/usm-data/EscalationpolicyAction
   </wsman:ResourceURI>
      <wsman:SelectorSet>
         <wsman:Selector Name="ASBOID.id">18</wsman:Selector>
         <wsman:Selector Name="ASBOID.source">4503599627370496</wsman:Selector>
      </wsman:SelectorSet>
   </wsa:ReferenceParameters>
   </wxf:ResourceCreated>
</env:Body>
</env:Envelope>
```

The bold syntax shows the returned selector properties for the created escalation action.

Example: Delete an escalation action

The following example SOAP message is a Delete request to delete an escalation action:

```
<env:Header>
   <wsa:ReplyTo xmlns:ns13="http://ns.ca.com/2009/01/usm-data">
   <wsa:Address env:mustUnderstand="true">
    http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
   </wsa:Address>
   </wsa:ReplyTo>
   <wsa:MessageID xmlns:ns13="http://ns.ca.com/2009/01/usm-data"
   env:mustUnderstand="true">
   uuid:4f440ab3-9c4b-46db-824a-426faf11e9bd
   </wsa:MessageID>
   <wsa:To xmlns:ns13="http://ns.ca.com/2009/01/usm-data" env:mustUnderstand="true">
```

```
   http://localhost:7090/sam/webservice
   </wsa:To>
   <wsman:ResourceURI xmlns:ns13="http://ns.ca.com/2009/01/usm-data">
   http://ns.ca.com/2009/01/usm-data/EscalationpolicyAction
   </wsman:ResourceURI>
   <wsman:OperationTimeout xmlns:ns13="http://ns.ca.com/2009/01/usm-data">
   PT30.000S
   </wsman:OperationTimeout>
   <wsman:SelectorSet xmlns:ns13="http://ns.ca.com/2009/01/usm-data">
       <wsman:Selector Name="ASBOID.id">18</wsman:Selector>
       <wsman:Selector Name="ASBOID.source">0</wsman:Selector>
   </wsman:SelectorSet>
   <wsa:Action xmlns:ns13="http://ns.ca.com/2009/01/usm-data"
   env:mustUnderstand="true">
   http://schemas.xmlsoap.org/ws/2004/09/transfer/Delete
   </wsa:Action>
</env:Header>
<env:Body/>
</env:Envelope>
```

The bold syntax shows the selectors that uniquely identify the escalation action. The SOAP response to this request is as follows:

```
<env:Header>
   <wsa:Action env:mustUnderstand="true"
   xmlns:ns11="http://schemas.sam.ca.com/webservice/1/alarm.xsd" xmlns:ns13="http://ns.ca.com/2009/01/usm-
data"
   xmlns:ns14="http://ns.ca.com/2009/07/usm-core">
   http://schemas.xmlsoap.org/ws/2004/09/transfer/DeleteResponse
   </wsa:Action>
   <wsa:MessageID env:mustUnderstand="true"
   xmlns:ns11="http://schemas.sam.ca.com/webservice/1/alarm.xsd" xmlns:ns13="http://ns.ca.com/2009/01/usm-
data"
   xmlns:ns14="http://ns.ca.com/2009/07/usm-core">
   uuid:5179bae4-8d0e-42a5-a944-b7f6c7c29eec
   </wsa:MessageID>
   <wsa:RelatesTo xmlns:ns11="http://schemas.sam.ca.com/webservice/1/alarm.xsd" xmlns:ns13="http://
ns.ca.com/2009/01/usm-data"
   xmlns:ns14="http://ns.ca.com/2009/07/usm-core">
   uuid:4f440ab3-9c4b-46db-824a-426faf11e9bd
   </wsa:RelatesTo>
   <wsa:To env:mustUnderstand="true"
   xmlns:ns11="http://schemas.sam.ca.com/webservice/1/alarm.xsd" xmlns:ns13="http://ns.ca.com/2009/01/usm-
data"
   xmlns:ns14="http://ns.ca.com/2009/07/usm-core">
   http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
   </wsa:To>
</env:Header>
<env:Body/>
</env:Envelope>
```

Additional Resources

This section includes links to additional helpful CA SOI information sources.

SOI Connectors Additional Resources

This section includes links to additional helpful CA SOI information sources. Communities and Social Media This page provides links to CA SOI user communities and other CA SOI social media destinations:

- CA SOI Community Forum

Product Education

This page provides links to product education resources:

- eduCAte on YouTube
- CA SOI Videos on YouTube

Additional Product Information

This page provides additional sources for CA SOI product information:

- CA SOI Product Page
- Product Release and Support Announcements

Green Book

Green Books provide knowledge focused on CA solution implementations and deployments. They deliver best practices and considerations based on real-world scenarios and knowledge compiled from global CA team experiences. Cross-functional teams of CA technical employees from field services, support and education collaborate with Technical Information to apply their expertise and create publications that deliver practical knowledge that goes beyond an "out-of-the-box" installation. Here is the link to the CA Service Operations Insight Green Book:

CA SOI Troubleshooting Guide for CA Support Partners.

Documentation Legal Notice

This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the “Documentation”) is for your informational purposes only and is subject to change or withdrawal by Broadcom at any time. This Documentation is proprietary information of Broadcom and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of Broadcom.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all Broadcom copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to Broadcom that all copies and partial copies of the Documentation have been returned to Broadcom or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is Broadcom Inc.

Provided with “Restricted Rights.” Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © Broadcom. All Rights Reserved. The term “Broadcom” refers to Broadcom Inc. and/or its subsidiaries. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

ca technologies
A Broadcom Company