CA Service Operations Insight

Event and Alert Management Best Practices Guide

r3.2

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 CA at any time. This Documentation is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

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 CA 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 CA that all copies and partial copies of the Documentation have been returned to CA 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 CA.

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.

CA Technologies Product References

This document references the following CA Technologies products:

- CA Application Performance Management
- CA Business Intelligence
- CA Clarity™ Project and Portfolio Manager
- CA CMDB
- CA Configuration Automation (formerly CA Application Configuration Manager)
- CA eHealth® Performance Manager (CA eHealth)
- CA Embedded Entitlements Manager (CA EEM)
- CA Event Integration
- CA Insight™ Database Performance Manager
- CA NSM
- CA Process Automation
- CA Service Desk
- CA Server Automation (formerly CA Spectrum® Automation Manager)
- CA SiteMinder®
- CA Spectrum®
- CA Systems Performance for Infrastructure Managers
- CA SystemEDGE
- CA Virtual Assurance

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 6: Working with Escalation Policy 87

Chapter 7: Event Management Overview 121

Chapter 8: Searching for and Viewing Events 137

Chapter 1: About This Guide

The *Event and Alert Management Best Practices Guide* contains information about viewing and managing the stream of events and alerts that CA SOI receives from connectors. This guide introduces the key concepts that are associated with event and alert management, describes the lifecycle of an alert that is imported into CA SOI, and contains architecture information. This guide shows you how to view, escalate, and group alerts; how to search for events; and how to create event policies.

This section contains the following topics:

Intended Audience

This guide is intended for any user who is responsible for managing any or all of the product alert or event 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.

Related Publications

The following publications, provided on the installation media and the CA SOI online bookshelf, provide complete information about CA SOI:

Administration Guide

Provides information about administering and maintaining the product after installation.

Connector Guide

Provides general information about connectors, the CA Catalyst infrastructure, and writing custom connectors.

Implementation Guide

Provides information about installing and implementing the product.

Online Help

Provides information about performing tasks in CA SOI user interfaces.

Readme

Provides information about known issues and information that is discovered after the guides were finalized. A CA SOI release may not have a Readme.

Release Notes

Provides information about operating system support, system requirements, database requirements, web browser support, and international support.

Service Modeling Best Practices Guide

Provides procedures and best practices for modeling services including the following methods: service imports, service discovery, and manual service modeling.

Troubleshooting Guide

Provides information and procedures to diagnose and resolve problems with CA SOI.

User Guide

Provides information for nonadministrative users about using the product, such as responding to alerts and viewing reports.

Web Services Reference Guide

Provides information about the CA SOI web services for interacting with resources such as CIs, services, alerts, relationships, and escalation policy.

The following publications provide information about the CA Catalyst infrastructure:

<Product Name> Connector Guide and Readme

Provides information about a specific CA Catalyst connector, including prerequisites, installation, configuration, data mapping, and known issues. The documentation for each CA Catalyst connector is included with its downloadable package.

Local Documentation and Online Bookshelf

CA SOI provides access to the documentation locally and online.

Local Documentation

The local documentation is installed in the SOI_HOME\Documentation folder and includes the PDFs for all guides. The online help is also installed with CA SOI and accessed through the Dashboard (PC and Mobile) and USM Web View. The local documentation is updated with specific releases only.

Online Bookshelf

The online bookshelf is on support.ca.com and provides the most current documentation set, which can be updated between releases. The online bookshelf also provides the documentation for the latest supported versions of CA Business Intelligence, CA EEM, and CA Process Automation. For a list of Bookshelf updates, click the Update History link on the Bookshelf.

CA SOI provides access to the online bookshelf in the following locations:

- The Dashboard provides a Bookshelf link.
- The Operations Console provides a menu link under Help, Bookshelf.

Note: If you are unable to access the online bookshelf, contact your system administrator to provide the documentation set PDFs.

Chapter 2: Introduction

This section introduces the role alerts and events play in the CA SOI product.

This section contains the following topics:

Event and Alert Management Overview

Managing the large stream of messages and fault conditions that the multitudes of data sources generate is a unique and difficult challenge in the modern IT infrastructure. Distributing the management of these conditions across multiple domain managers increases overhead, the potential for error or duplication of work, and time to resolution.

CA Service Operations Insight (CA SOI) collects and displays data from all domain managers in your enterprise. Therefore, CA SOI is the ideal product to use as a unified alert management tool across all managed domains. Operations personnel can use the unified alert view to manage all fault conditions in one place, drilling into the source product when necessary to resolve problems.

You can manage alerts in CA SOI from multiple perspectives:

Service-oriented

Alerts that are associated with managed services appear when viewing that service, and you can manage alerts from this perspective to ensure service health.

Queue-oriented

CA SOI also introduces the concept of alert queues to enable unified alert management with no reliance on existence in a managed service. CA SOI displays all collected alerts, including alerts that are not associated with any services. You can group alerts that share common characteristics into queues so that you can manage them together.

These distinct perspectives let you manage alerts in the manner that best suits your needs. If your enterprise is in the early stages of transitioning to a service-oriented paradigm, you can use alert queues to ease the transition to services.

Due to its unique position in the product hierarchy, CA SOI can also serve as the single escalation point for all alerts enterprise-wide. Notification emails, help desk tickets, and other escalations can all originate from CA SOI to consolidate and simplify the escalation and remediation process.

CA SOI also includes an event management layer that lets you define rules for processing raw and normalized event messages. You can define policies that influence how and when raw events appear as alerts, so that operators are presented with a quality set of actionable alert conditions.

The combined features of event and alert management let you effectively manage alert data throughout its lifecycle, from processing to assignment to management to escalation to resolution.

Concepts

This section introduces the core concepts and terms that are associated with event and alert management.

Alerts

An *alert* is a message on the Operations Console that reports a fault condition that is associated with a resource or service. Alerts affect CI severity and, when associated with a service, overall service health. Alerts let you monitor the health of your enterprise and take a corrective action when required.

Alerts can come from the following sources:

- CA Catalyst connectors send alerts from their integrated domain managers to CA SOI. Connectors are the primary source of alert data.

  Note: Domain managers have varying terms for the fault conditions they report, such as alarm, notification, or event. Connectors convert these conditions so that they all display as alerts in CA SOI and adhere to the USM schema properties for the alert type.

- CA SOI generates alerts on services that indicate service health degradation.
- The Event Management component lets you generate new alerts that are based on correlated event data.
- CA SOI connectors such as the Universal connector let you manually establish custom integrations that produce alert data.

Alert management includes the following features:

- Displaying all alerts in a single console view.
- Launching the source domain manager that generated the alert for more information.
- Alert assignment, annotation, and acknowledgment to track that status of an alert until it is cleared.
- Alert queues to group alerts that share common characteristics for consolidated management.
- Escalation policy to automate the actions to take in response to alert conditions.
- Full help desk integration so that CA SOI can serve as the single point of escalation to CA Service Desk or other help desk products.

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 (see page 21) as events (see page 19). 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.

Note: For more information about how alert quality and risk affect the service impact and health, see the *Service Modeling Best Practices Guide*.

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.

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.

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, CA NSM WorldView for system faults, and CA Application Performance Management for application performance
- High-volume raw event sources, such as CA NSM Event Management, 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.

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 (see page 20) 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.

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

Alert Lifecycle

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.

Example 3: CA NSM high CPU alert

- The CA NSM connector receives an alert from CA NSM indicating that CPU usage is high on a managed server.
- The alert is normalized and stored as an event.
- The event matches an Event Management enrichment policy that adds a contact name from an external database to the event.

- The enriched event becomes an infrastructure alert and displays associated with its computer system CI. The enrichment value appears in the property to which it was assigned.
- The alert matches the criteria for an alert queue that is based on the enriched contact name and is added to the queue.
- Before any escalation policy or manual action can occur, the CPU usage drops to acceptable levels on the server. The alert automatically clears when the connector receives notification of the normal CPU usage level.

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 and queues. You can perform all operations on alerts from the Operations Console and view comprehensive alert details. This guide primarily documents alert management operations from the Operations Console.

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.

Chapter 3: Alert Management Overview

This section contains introductory information about alert properties, how and where you can view alerts, and various available alert operations.

This section contains the following topics:

Introduction to Alert Management

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 (see page 29) 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 (see page 41), assign (see page 41), annotate (see page 42), escalate (see page 45), and clear (see page 42) 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 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 (see page 29) 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 the USM schema documentation. For information about how to access the USM schema documentation, see the *Connector Guide*.

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.

Severity

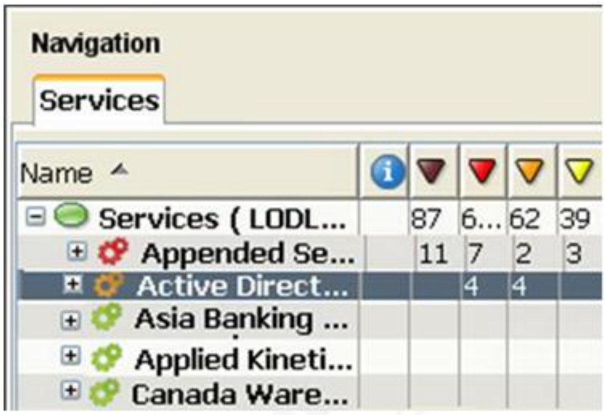
Severity indicates the condition of a CI as reported from the domain manager to CA SOI through alerts. If multiple domain managers send alerts for the same CI, the highest severity is used. CI severity helps determine the service impact by propagating the impact of the condition to related CIs in the 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

Severity	Color	Description
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).



Name ▲	ℹ	▼	▼	▼	▼
⊟ Services (LODL...		87	6...	62	39
⊞ Appended Se...		11	7	2	3
⊞ Active Direct...			4	4	
⊞ Asia Banking ...					
⊞ Applied Kineti...					
⊞ Canada Ware...					

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.

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).

View Alerts, Alert Details, and Extended Information

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.

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 (see page 87) and alert queue rules (see page 77).

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 Alerts History table contains the following information:

- Associated CI name
- Severity
- Creation and clear date
- 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 (see page 29). 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 (see page 38), select Matches regex from the Comparison Type drop-down list. Click Test Regex to open the Regex Tester (see page 39) 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.

Regular Expressions

Regular expressions are supported in the following areas:

- Alert filters (see page 36)
- Alert queue criteria (see page 77)
- Escalation policy definitions (see page 90) and action exception criteria (see page 112)
- Service Discovery dynamic services. For more information, see the *Service Modeling Best Practices Guide*.

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:

   - The 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.

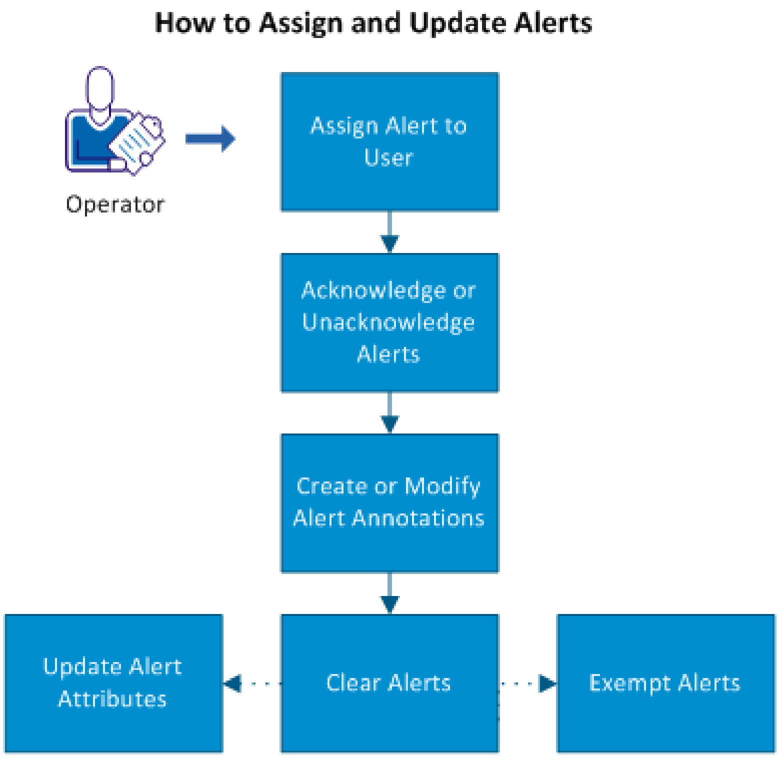
4. Perform one of the following actions:

   - 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.

How to Assign and Update Alerts

As an operator you can update alerts and 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:


How to Assign and Update Alerts
Operator
Assign Alert to User
Acknowledge or Unacknowledge Alerts
Create or Modify Alert Annotations
Update Alert Attributes
Clear Alerts
Exempt Alerts

1. Assign an alert to a user to resolve the issue (see page 41).
2. Acknowledge or unacknowledge the alert (see page 41).
3. Add an alert annotation (see page 42).
4. Clear an alert (see page 42).
5. (Optional) Update the alert attributes (see page 43).
6. (Optional) Exempt alerts (see page 44).

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 (see page 43).

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.

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.

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.

More information:

Create a Clear Alert Action (see page 112)

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.

Update an Alert from the Mobile Dashboard

The Mobile Dashboard lets you view alerts in the context of their associated services and alert queues and also perform certain update operations.

Follow these steps:

1. Access the Mobile Dashboard from the following URL:

   `http://uiserver:port/ssamobile`

   uiserver

   Defines the UI Server name.

   port

   Defines the UI Server port number.

   Default: 7070

   The login page opens.

2. Enter valid credentials and click Log In.
3. Click a service or alert queue, and click Alerts.
4. Click an alert.
5. Click Acknowledge Alert or Clear Alert and confirm the action.

   Note: If the alert is already acknowledged, you can unacknowledge the alert.

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 the *User Guide*.

The Alert Details tab lets you view and change the Exempt status of the selected alert.

How to Escalate Alerts

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 (see page 46)
- Send an email notification (see page 47)

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 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.

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: This feature is available only if you have appropriate access privileges. For more information, contact your CA SOI administrator. 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.
   - 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.

Escalate an Alert from the Mobile Dashboard

You can escalate an alert from the Mobile Dashboard when mobile interface access is most convenient.

Follow these steps:

1. Access the Mobile Dashboard from the following URL:

   http://*uiserver*:*port*/mobile

   uiserver

   Defines the UI Server name.

   port

   Defines the UI Server port number.

   Default: 7070

   The login page opens.

2. Enter valid credentials and click Log In.
3. Click a service or alert queue, and click Alerts.
4. Click an alert.
5. Perform one of the following actions:
   - Click an entry for any defined escalation action, and click Yes on the confirmation dialog.

     The escalation action runs. You can run any defined escalation action from the Mobile Dashboard, but you cannot create and run a new action.
   - Click E-mail.

     The E-mail dialog opens.
6. (E-mail only) Enter a subject, recipient address, and message and click Send.

   The email is sent.

Subscribe to Alert RSS Feeds from USM Web View

The USM Web View interface lets you subscribe to RSS feeds for alerts, services, and CIs.

Note: For more information about USM Web View, see the *User Guide* or Online Help.

Follow these steps:

1. Open the USM Web View interface from the CA SOI Dashboard.
2. Click Browse by CI Type on the USM Web View Home page.
3. Click Alert.

4. Click More Options, and click 'For updates to items matching this search' under Subscribe.

   The Alert RSS feed appears.

5. Click Subscribe to this feed.

You can subscribe to feeds that notify you when any new alert occurs or when a new alert occurs on a specific service or CI.

Follow these steps:

1. Select a CI or service using the search or browse functionality.

   A page opens with CI or service details and configuration options.

2. Click More Options, and click one of the following under Subscribe For:

   Alerts on this CI

   Displays alerts open on the selected CI or service.

   Alerts on this service this CI belongs to

   Displays alerts open on the service to which the CI or service belongs.

   The selected feed for that CI or service opens.

3. Click Subscribe to this feed.

   You are subscribed to a feed that notifies you when new alerts occur on this CI, service, or associated service.

Chapter 4: Alert Management Administration

This section provides alert management administration procedures.

This section contains the following topics:

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 (see page 99)

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.

Note: For more information about configuring each integration, see the *Administration Guide*.

Configure Alert Management Global Settings

Global settings are available that influence how alerts appear in the Operations Console and how CA SOI manages alerts.

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 alerts—it 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

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.

Rename Custom 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

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.

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: Five files correspond to the five 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:

   ```
   <?xml version="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 five 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.

You can use the CI user attributes only in escalation, queue or filter policy criteria with alerts on managed CIs or unmanaged alerts that are associated with a real CI that is part of a low granularity service. These attributes are not available for unmanaged alerts on unmanaged CIs.

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 1-5. For example, "CIuserAttribute1" populates what is shown as "CI User Attribute (1)" in the Operations Console.

Example

In the < Format > 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*].

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.

Hide Alerts in Maintenance Mode

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 parent's 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 *Administration Guide*.

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 the *Administration Guide*.

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 selected in Services tab	Alerts display for these CIs	Comments
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.

Alert Synchronization

CA SOI supports synchronization of the Cleared and Acknowledged properties from CA SOI to the following products and releases:

- CA Spectrum r9.2.0 H03 (Acknowledged and Cleared properties)
- CA NSM r11.2 SP2 (Acknowledged property only, DSM connector only)
- Microsoft SCOM 2007 R2 (Cleared and Acknowledged properties)

When you clear or acknowledge an alert in CA SOI from one of these domain managers, and update propagates to the source domain manager.

Note: For more information about how alerts are synchronized, see the *Administration Guide*.

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. Also, the connectors must have the Alert specified as an “InboundToConnector” type in their policies.

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. Restart the CA SAM Application Server service.

Exempt Alerts from Impact Analysis

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.

You can exempt alerts using any of the following methods:

- On the Operations Console (see page 66).
- With Event Management (see page 67).
- With customized connector policy (see page 68).
- On a mobile device (see page 69).

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 the *User Guide*.
   - 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 the Event Management sections.

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 the *Connector Guide*.

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.

Set Root Cause Analysis Mode

An administrator can set the root cause analysis mode, which 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 the *Administration Guide* or the *User Guide*.

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 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.

Configure Root Cause Analysis Global Setting

An administrator sets the root cause analysis source (see page 69) on the Dashboard Administration tab.

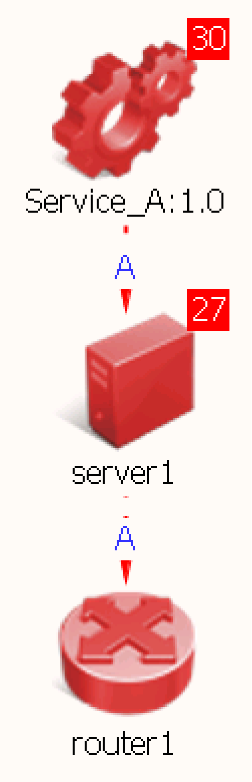
Follow these steps:

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. In the Root Cause Analysis Setting section, select the Root Cause Analysis Mode and click Save.
6. Restart the CA SAM Application Server service.

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:



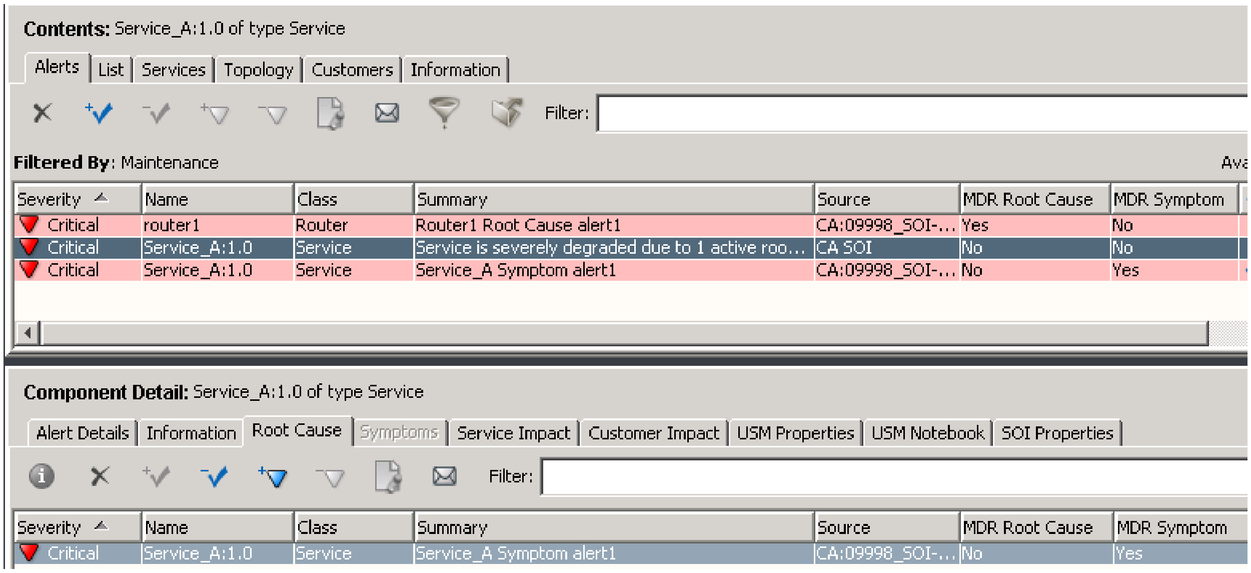
The domain manager is CA Spectrum, which is configured to determine root cause, and symptoms. In each example, you will see how changing the root cause analysis mode changes the root cause and symptoms on the Operations Console.

Example: CA SOI Derived Root Cause Analysis

In this example, the administrator configures the Root Cause Global Setting (see page 71) 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.



Contents: Service_A:1.0 of type Service

Filtered By: Maintenance

Severity	Name	Class	Summary	Source	MDR Root Cause	MDR Symptom
Critical	router1	Router	Router1 Root Cause alert1	CA:09998_SOI-...	Yes	No
Critical	Service_A:1.0	Service	Service is severely degraded due to 1 active roo...	CA SOI	No	No
Critical	Service_A:1.0	Service	Service_A Symptom alert1	CA:09998_SOI-...	No	Yes

Component Detail: Service_A:1.0 of type Service

Severity	Name	Class	Summary	Source	MDR Root Cause	MDR Symptom
Critical	Service_A:1.0	Service	Service_A Symptom alert1	CA:09998_SOI-...	No	Yes

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.

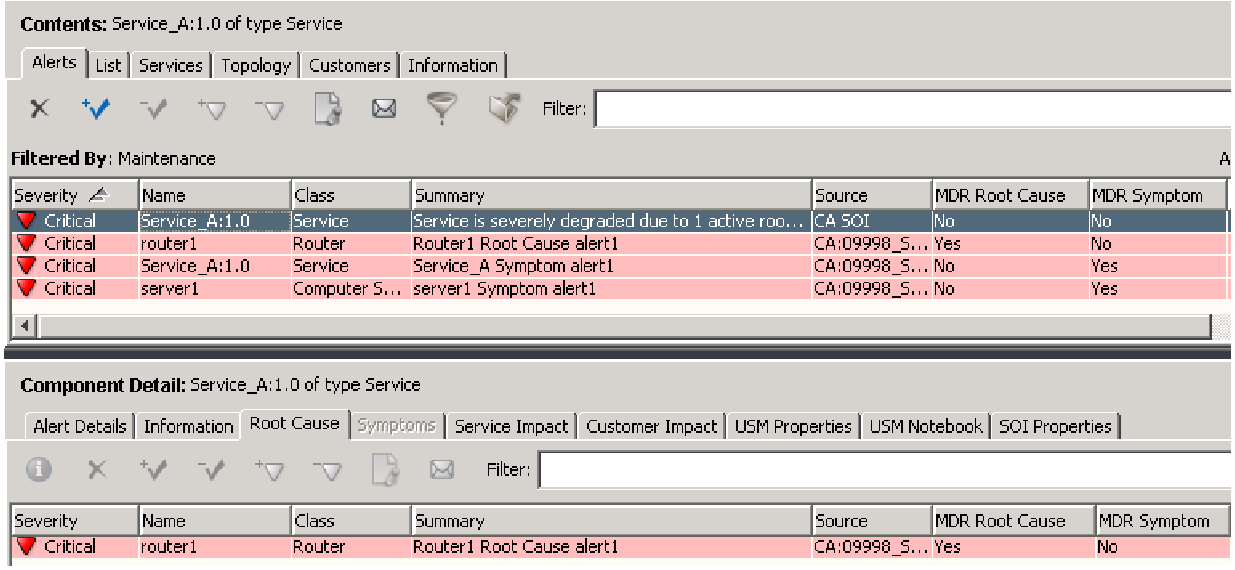
Therefore, in CA SOI mode, verify that you are viewing alerts with CA SOI as the Source in the column. CA Spectrum determines the symptoms in this example, not CA SOI. CA SOI calculated the root cause.

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 (see page 71) 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.



Contents: Service_A:1.0 of type Service

Filtered By: Maintenance

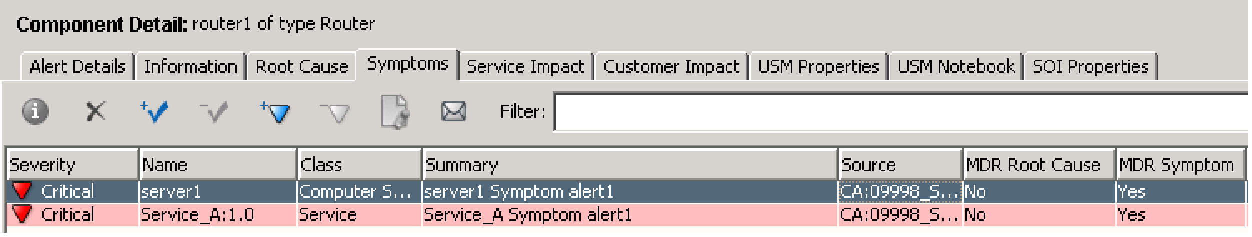
Severity	Name	Class	Summary	Source	MDR Root Cause	MDR Symptom
Critical	Service_A:1.0	Service	Service is severely degraded due to 1 active roo...	CA SOI	No	No
Critical	router1	Router	Router1 Root Cause alert1	CA:09998_S...	Yes	No
Critical	Service_A:1.0	Service	Service_A Symptom alert1	CA:09998_S...	No	Yes
Critical	server1	Computer S...	server1 Symptom alert1	CA:09998_S...	No	Yes

Component Detail: Service_A:1.0 of type Service

Severity	Name	Class	Summary	Source	MDR Root Cause	MDR Symptom
Critical	router1	Router	Router1 Root Cause alert1	CA:09998_S...	Yes	No

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 (see page 73), 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.



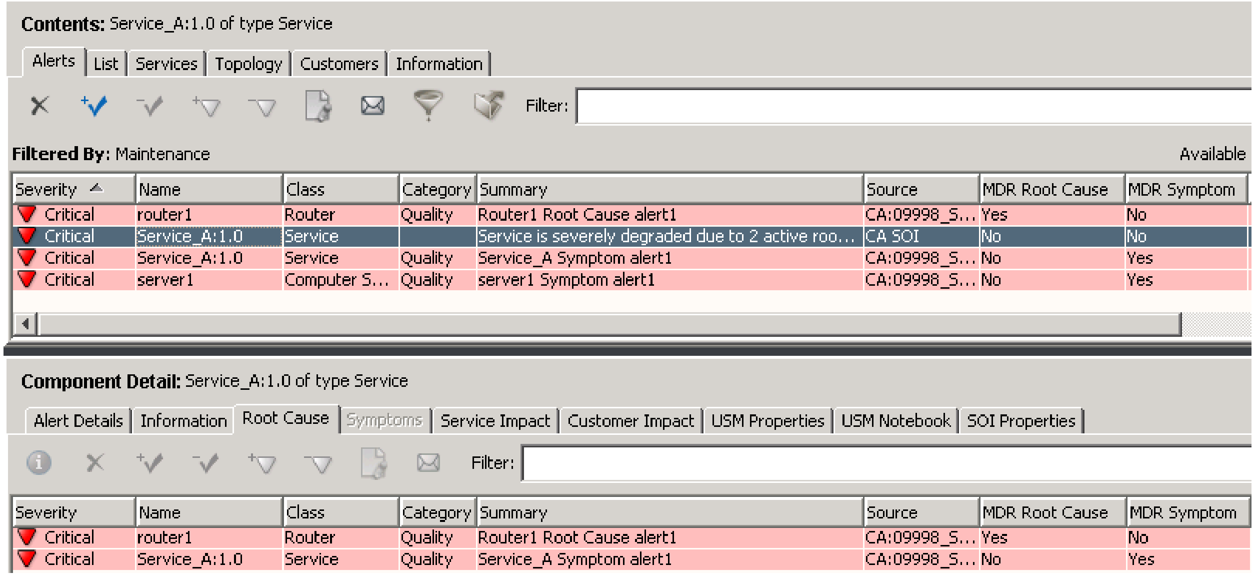
Component Detail: router1 of type Router

Severity	Name	Class	Summary	Source	MDR Root Cause	MDR Symptom
Critical	server1	Computer S...	server1 Symptom alert1	CA:09998_S...	No	Yes
Critical	Service_A:1.0	Service	Service_A Symptom alert1	CA:09998_S...	No	Yes

Example: Combination (CA SOI and Domain Manager) Derived Root Cause Analysis

In this example, the administrator configures the Root Cause Analysis Setting (see page 71) 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.



Contents: Service_A:1.0 of type Service

Filtered By: Maintenance

Available

Severity	Name	Class	Category	Summary	Source	MDR Root Cause	MDR Symptom
Critical	router1	Router	Quality	Router1 Root Cause alert1	CA:09998_S…	Yes	No
Critical	Service_A:1.0	Service		Service is severely degraded due to 2 active roo…	CA SOI	No	No
Critical	Service_A:1.0	Service	Quality	Service_A Symptom alert1	CA:09998_S…	No	Yes
Critical	server1	Computer S…	Quality	server1 Symptom alert1	CA:09998_S…	No	Yes

Component Detail: Service_A:1.0 of type Service

Severity	Name	Class	Category	Summary	Source	MDR Root Cause	MDR Symptom
Critical	router1	Router	Quality	Router1 Root Cause alert1	CA:09998_S…	Yes	No
Critical	Service_A:1.0	Service	Quality	Service_A Symptom alert1	CA:09998_S…	No	Yes

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.

Chapter 5: Working with Alert Queues

This section describes how to create, view, and manage alert queues.

This section contains the following topics:

How to Create and Manage Alert Queues

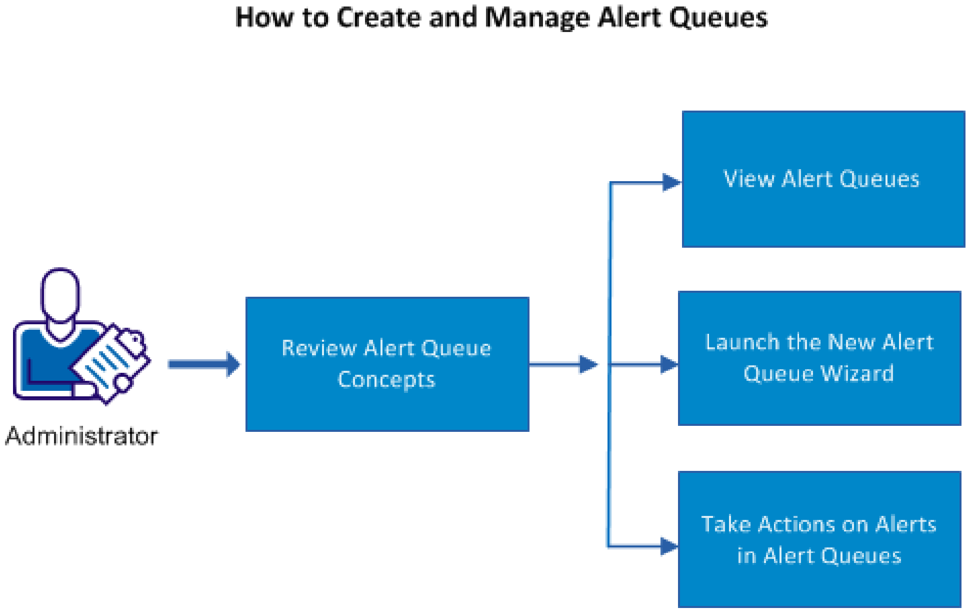
As an administrator, you can define alert queues then assign the escalation policies and user groups to the queues.

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:


How to Create and Manage Alert Queues
Administrator
Review Alert Queue Concepts
View Alert Queues
Launch the New Alert Queue Wizard
Take Actions on Alerts in Alert Queues

1. Review the alert queue concepts (see page 78) to learn about the default alert queue and how alert queue security works.
2. View your alert queues (see page 79).
3. Launch the New Alert Queue wizard (see page 80) to create alert queues.
4. Take actions on the alerts in your alert queues (see page 85).

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 see all non service-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 see alert queues only to which you have access privileges.
- You can see alerts in alert queues only on the services to which you have 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.

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 (see page 31).

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 (see page 38), select Matches regex from the Comparison Type drop-down list. Click Test Regex to open the Regex Tester (see page 39) 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 the *USM Schema Documentation* that is provided on the CA SOI Bookshelf.

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 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.

Take Actions on Alerts in Alert Queues

Depending on your user access privileges, you can trigger escalation policy actions on a selected alert in an alert queue.

Follow these steps:

1. Start the Operations Console and click the Alerts Queues tab in the Navigation pane.
2. Select an alert queue.

   The alerts in the selected alert queue display in the Contents pane. The Contents label displays the currently selected queue.
3. Right-click an alert and select Take Action.
4. (Optional) Enter a filter string to filter the available actions.
5. (Optional) Click Create to define a new escalation policy action (see page 90).
6. (Optional) Select the check box to use the selected action as the default without prompting again.
7. Select an action and click OK.

Chapter 6: Working with Escalation Policy

This section describes how to create escalation policy and actions to automate the response to alert conditions.

This section contains the following topics:

How to Create Escalation Policy

As an administrator, you define escalation policies 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.

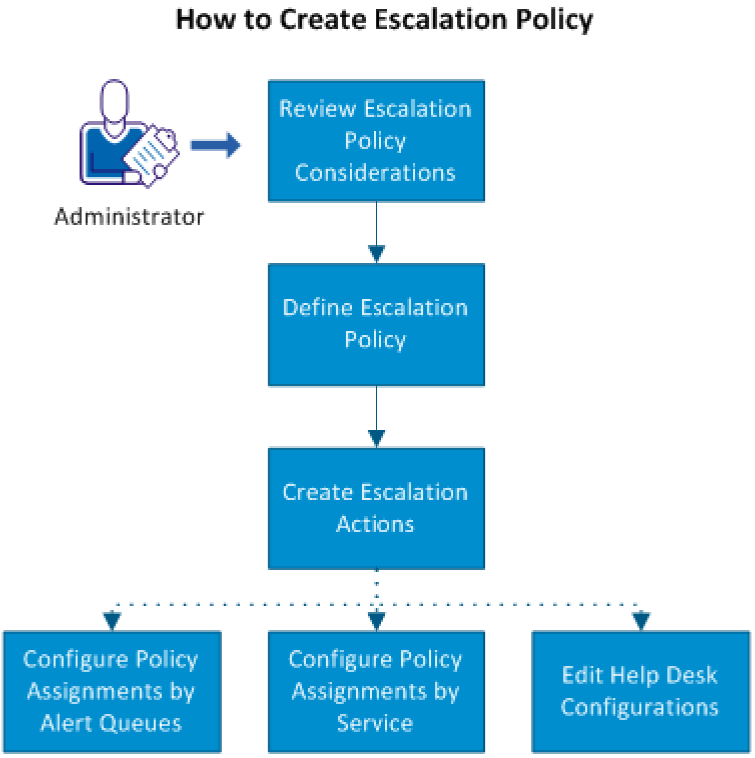
Global

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:


How to Create Escalation Policy
Administrator
Review Escalation Policy Considerations
Define Escalation Policy
Create Escalation Actions
Configure Policy Assignments by Alert Queues
Configure Policy Assignments by Service
Edit Help Desk Configurations

1. Review the escalation policy considerations (see page 89).
2. Define the escalation policy (see page 90).
3. Create escalation actions (see page 97).
4. (Optional) Configure policy assignments by alert queues (see page 118).
5. (Optional) Configure policy assignments by service (see page 119).
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 (see page 51). Configure email for both the SA Manager and UI Server.
- An alert only 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 an AND relationship.

Define Escalation Policy

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 (see page 119).

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 (see page 69)).

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 (see page 38), select Matches regex from the Comparison Type drop-down list. Click Test Regex to open the Regex Tester (see page 39) 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. 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 (see page 97).

    You can add multiple actions to the policy.

14. (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 (see page 119).

15. (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.

16. (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.

17. 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 the *USM Schema Documentation* that is provided on the CA SOI Bookshelf.

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 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 1-5. 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 the *Implementation Guide*.

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

As an administrator, you configure escalation policy to perform any of the following automated actions:

- Notification to a technician or operator by email (see page 100)
- Open a help desk ticket (see page 101)
- Update a help desk ticket (see page 104)
- Run a command (see page 105)
- Open an announcement (see page 106)
- Run a CA Process Automation process (see page 108)
- Clear an alert (see page 112)

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.

Important! Before you create escalation policy actions, consider the following items:

- Verify the configuration for email, help desk applications, and CA Process Automation Server configuration (see page 51).
- You can configure email notifications so that CA SOI notifies you when an escalation action fails. For more information about failure email notifications, see the *Troubleshooting Guide*.

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 (see page 101) or Create Announcement (see page 106) action
- When you populate the Message, Subject, Recipients, and From fields in a Send Email action (see page 100)
- When you populate the Command field in an Run a Command action (see page 105)
- When you add values to parameters in an Execute Automated Process action (see page 108)

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:

$[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 (see page 99).

$[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 (see page 99).

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 (see page 90).
   - 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 (see page 97) 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 (see page 99).

   - If you want to use the $[Mobile UI URL] token, you first enable its use (see page 99).
   - If you want to use the CI user attribute tokens, you first set the attributes. For more information about setting CI user attributes, see the *Implementation Guide*.

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 the *Implementation Guide*.

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 (see page 90).
   - 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 (see page 112).
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 (see page 97) 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 (see page 99).
   - If you want to use the $[Mobile UI URL] token, you first enable its use (see page 99).
   - If you want to use the CI user attribute tokens, you first set the attributes. For more information about setting CI user attributes, see the *Implementation Guide*.
   - 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 the *Implementation Guide*.

   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 (see page 112) 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 the *Implementation Guide*.

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 (see page 90).
   - 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 (see page 112).
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 (see page 112) 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 (see page 90).

   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 (see page 97) 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 (see page 99).
   - If you want to use the $[Mobile UI URL] token, you first enable its use (see page 99).
   - If you want to use the CI user attribute tokens, you first set the attributes. For more information about setting CI user attributes, see the *Implementation Guide*.
   - 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 (see page 90).
   - 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 (see page 112).

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 (see page 97) 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 (see page 99).
   - If you want to use the $[Mobile UI URL] token, you first enable its use (see page 99).
   - If you want to use the CI user attribute tokens, you first set the attributes. For more information about setting CI user attributes, see the *Implementation Guide*.
   - 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 the *Implementation Guide*. You also set the help desk configuration for in-context links to help tickets. For more information, see the *Administration Guide*.

   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.

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. The documentation is provided on the CA SOI bookshelf.

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 (see page 90).

   - 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, configure the SSL connection with CA Process Automation.

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 (see page 110).

   - If you want to enter a CA Process Automation process manually, see Use a Process without an Associated Form (see page 111).

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 (see page 97) 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 (see page 97) 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 (see page 90).
   - 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 (see page 38), select Matches regex from the Comparison Type drop-down list. Click Test Regex to open the Regex Tester (see page 39) 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 (see page 113).

   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.* This guide is provided on the CA SOI Bookshelf.

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 the *Implementation Guide*.

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 (see page 90), 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 (see page 90), 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 (see page 90) 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.

Chapter 7: Event Management Overview

This section introduces Event Management architecture and describes how events are stored.

This section contains the following topics:

Introduction to Event Management

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 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

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 levels—simple 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.

More information:

How Events are Processed in Event Management (see page 123)
Event Management Components (see page 124)

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 (see page 124)
- Mid-tier Connector (see page 131)
- Event Service (Query/Deploy Service) (see page 131)
- Event Management UI Service (see page 132)

More information:

How Events are Processed in Event Management (see page 123)

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* (see page 126) and *archive* (see page 126). 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-4f2
    9-4070-8704-34d214752646;publishAction=IMPORT;entitytype=Item'>
        <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_ComputerS
        ystem_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</ConnectorConfigMdrProdInsta
        nce>
        </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>
```

More information:

Event Store XML File Naming Convention (see page 126)
Archiving Event Store XML Files (see page 127)

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.

More information:

Archiving Event Store XML Files (see page 127)

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

More information:

How the Archiving Process Works (see page 127)

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.

More information:

Archiving Event Store XML Files (see page 127)

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.

More information:

Archiving Event Store XML Files (see page 127)
How the Archiving Process Works (see page 127)

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%.

   Important! 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.

More information:

Archiving Event Store XML Files (see page 127)

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 NSM, 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 ActiveMQ 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 (see page 158)
- Occurrence frequency (see page 159)
- Advanced search techniques (see page 159)
- Raw events (see page 161)
- Moving from simple to complex (see page 162)

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 (see page 171)
- Create event action (see page 178)
- Database enrichment (see page 188)
- Java method enrichment (see page 195)
- Script enrichment (see page 201)
- Map enrichment (see page 205)
- Normalization (see page 213)

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 (see page 230)
- Create a new event to indicate a crashing service (see page 232)
- Combine a create event action with an enrichment using reevaluation (see page 236)
- Normalize monitoring traps (see page 240)

Event Management KPIs

CA SOI captures various KPIs that enable you to pinpoint performance problems for specific event processing modules. You can use these KPIs to verify that all processing modules are operating correctly and efficiently. This information assists you in performing detailed diagnostics to assess the health of each module. Proper analysis of this information can further help you quickly identify and resolve associated issues. For example, the KPIs data can provide more performance-related information about why your environment is running slow, and can help you take appropriate measures.

The KPIs are written to individual connector-specific XML files, which are named as *mdrprod-mdrprodinstance*.xml (for example, CA-09998-LOD20.abc.com.xml, CA-09997-LOD20.abc.com.xml, and CA-09993-LOD20.abc.com.xml). Each connector-specific KPIs file is stored in the SOI_HOME\resources\Core\Kpi folder. These KPIs are collected for the type *Core*, which includes all core event processing modules (for example, Classifier and Evaluator). The following list describes all of the assessed KPIs:

ProcessTime

Specifies the time that the module has been processing events in milliseconds.

TotalEvents

Specifies the total number of events that processed since the module was last started.

LastTput

Specifies the most recent event throughput value in events per second.

AvgTput

Specifies the average event throughput in events per second since the module was last started.

MaxTput

Specifies the maximum event throughput in events per second since the module was last started.

MinTput

Specifies the minimum event throughput in events per second since the module was last started.

FilteredEvents

Specifies the number of events that have been filtered since the module was last started. This KPI covers events that are filtered explicitly according to the Filter operation in the policy and implicitly because of their inability to be classified.

QueueLength

Specifies the number of events queued for a given module to process since the module was last started. Longer queues indicate a backlog of events. A status of Warning indicates more than 300 queued events, and a status of Critical indicates more than 500 queued events.

ExceptionCount

Specifies the number of code exceptions since the core was last started.

ThreadCount

Specifies the number of active processing threads since the core was last started. Additional threads are created for performing normalization and enrichments. A status of Warning indicates more than 100 active threads, and a status of Critical indicates more than 200 active threads.

FqdnResolved

Specifies the total number of times a fully qualified domain name was successfully resolved and added to the internal cache since the core was last started.

Note: These names expire from the cache every 120 seconds by default.

FqdnUnresolved

Specifies the total number of times a fully qualified domain name was unable to be resolved since the core was last started.

Note: The name *null* is placed in the cache when this scenario occurs. These names expire from the cache every 120 seconds by default.

TotalRulesFired

(Evaluator only) Specifies the total number of Drools rules that are activated on an event since the module was last started.

PolicyName-RuleType

(Evaluator only) Specifies the total number and type of Drools rules that are activated on an event for a specific policy since the module was last started. For example, a line <kpi name='ABC-EnrichEvent' value='3' status='NORMAL' /> in a KPIs file specifies that the rule of type EnrichEvent was triggered three times for a policy with the name ABC.

Chapter 8: Searching for and Viewing Events

This section describes how to run the event searches from the Event Policies dialog.

This section contains the following topics:

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 the USM schema documentation on the CA SOI Bookshelf.

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 (see page 20) 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 (see page 151) 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 and CA NSM Event Management, 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 (see page 150).

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 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. It would exclude events written to the current 6:00 to 7:00 file, as the “last modified date” (of 6:35) is outside of the scoping criteria.

   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 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 of the last 4 hours would find all events stored from 2:00 to 6:00. It would exclude events written to the current 6:00 to 7:00 file, as the “last modified date” (of 6:35) is outside of the scoping criteria.

   The time range appears in the Time Range field.

6. 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'.

7. (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.

8. 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 (see page 150).

9. 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.

10. (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.

11. (Optional) Click Create Policy to save the search or create an event policy based on the search (see page 168).

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 (see page 210).

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

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 (see page 152)
- Operators and Expressions (see page 153)
- Functions (see page 154)
- Special Characters (see page 155)
- Scoping (see page 155)
- Raw Event Searches Limitations (see page 156)
- Normalized Event Searches Limitations (see page 157)

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(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 (see page 210).
- 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.

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 CA NSM events from the Tandem syslog

```
matches(evtlog_text,’^TAN$’)
```

The CA NSM Event Console collects events from multiple sources. The source policy that the event connector provides splits these event sources into separate classes. This search returns collected events from the Tandem syslog so that you can write specific mapping rules for these events.

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

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.

Chapter 9: Working with Event Policies and Actions

This section describes how to create event policies for performing actions on events that meet search criteria.

This section contains the following topics:

Event Policy with Actions

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

Lets you exclude 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

Lets you create 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

Lets you add 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

Lets you configure 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

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, CA NSM, 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

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 (see page 245).

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 (see page 168), 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.

Create an Event Policy with a Create Event Action

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 (see page 211) 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 (see page 168), 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. 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 (see page 184)
   - Java method (see page 192)
   - Script (see page 198)
   - Map only (see page 204)

   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 (see page 168), 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

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 (see page 181).

   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.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 (see page 188).

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 (see page 181).

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 (see page 188) 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 (see page 188) 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

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 (see page 181).

   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\epluscore.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 (see page 181).

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

Important! 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:
    - endpointref: http://*cmdbserver*:8080/axis/services/USD_R11_WebService?wsdl
    - userid: ${user}
    - password: ${password}
    - propertylist: location.address,location.city

      Note: For CA CMDB r12 and above, use location.address1 instead of location.address.
    - selectquery: dns_name like "%s"
    - 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.E
  Q.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:
    - i1: -searchmethod=by_name
    - i2: -broadcast=yes
    - i3: -landscapeout=*spectrumserver*
    - i4: landscapeuser=${user}
    - i5: -modeltype=Host_Device
    - i6: modelname=${AlertedMdrProdInstance }
    - 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

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 (see page 181).

   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.

   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.

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.

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,severi
    ty,%1
    )
else (if %1==Major
            (
            echo
    lname,Black,fname,Bill,email,email02@company.com,phone,631-002-0002,sever
    ity,%1
            )
else (if %1==Minor
            (
            echo
    lname,Unum,fname,Sven,email,email03@company.com,phone,631-003-0003,severi
    ty,%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}

7. 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.

8. Save or deploy the policy (see page 181).

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 descriptive information from CA NSM

This example shows how you can use the enrich event action to enrich events with output information from a script when search patterns match. The example uses the CA NSM enrichment provided with the Mid-tier connector. This enrichment could be useful in CA SOI for situations such as the following:

- You want to use CI properties in CA NSM in alert queue criteria or for escalation policy
- You are using custom properties in CA NSM that are not imported into CA SOI
- You want to use information from CA NSM in alerts or CIs that are not managed in CA NSM
- You want to parse partial information from a property for use in a different context

Important! The enrichments use .jar files that are only available with the Mid-tier connector. Deploy these provided enrichments on the Mid-tier connector only.

This example enriches events with comment information stored about the related CI in CA NSM. The information could help you resolve alerts related to the CI. For this enrichment to work, at the minimum, a CA NSM WorldView client must be installed with the remote WorldView repository on the connector system to which you deploy the event policy.

- Select Script on the Enrichment Configuration page, and select NSM 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:
    - i1: /r *nsmserver*
    - i2: /u ${user}
    - i3: /p ${password}
    - i4: /pf dnsname
    - i5: /pv ${pattern1.AlertedMdrProdInstance}

    This parameter configuration queries the defined CA NSM instance for CIs with a dnsname property that matches the AlertedMdrProdInstance property value and returns the comment property value of the matching CI. It uses substitution strings for the required CA NSM credentials (referencing the credentials entered on the previous page) to avoid entering the information unencrypted.
  - Enter ${comment} in the Assigned Value cell corresponding to the User Attribute 1 Event Property cell.

This enrichment queries the defined CA NSM instance for the comment property of the CI with a dnsname property value that matches the AlertedMdrProdInstance property value. If there is no match, the enrichment does not occur. If there is a match, the comment value is returned from the matching CI and assigned to the User Attribute 1 property in the enriched event.

For example, consider an event with the AlertedMdrProdInstance property value of server5. The server5 value matches a CA NSM WorldView managed object DNS name. The comment value in the matching CI is 'Reinstalling operating system', and this value appears in the enriched event for User Attribute 1. In this case, the logged comment would indicate that you can clear any alert associated with the server being down, because the operating system is currently being refreshed.

- Deploy the policy on the Mid-tier connector.

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

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 (see page 181).

   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 (see page 181).

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&q=${pattern1.AlertedMdrElementID }
  ```
- 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

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 (see page 181) 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 (see page 150), 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 (see page 210).
   - 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 (see page 211) 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 (see page 212) 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 (see page 212):

  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 (see page 212):

  - 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.

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.

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.

For detailed descriptions of the functions, see the *CA Catalyst Implementation Guide*.

Managing Event Policies

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

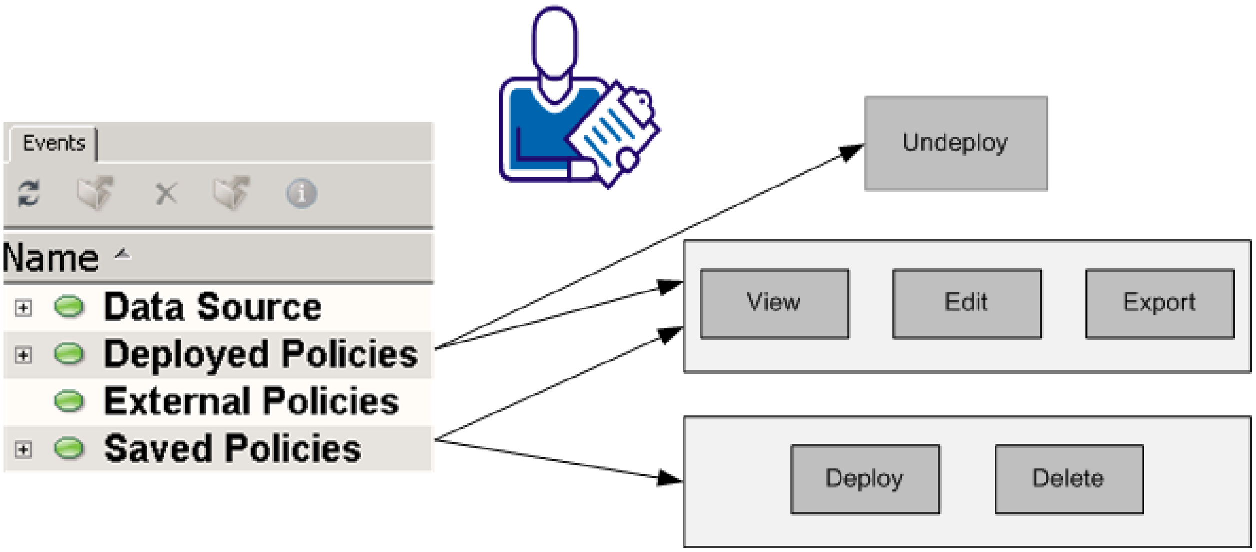
Contains policies that are created or refined manually outside of the user interface.

Saved Policies

Contains saved policies that are currently undeployed.

The following graphic shows the available policy management tasks for each type of event policy:

Event Policy Management Tasks by Policy Type



Administrators can manage event policies in several ways:

- View event policy summary (see page 224)
- Edit an event policy (see page 224)
- Deploy or activate a saved, deactivated, or imported event policy (see page 225)
- Undeploy or deactivate an event policy (see page 226)

- Delete an event policy (see page 226)
- Export an event policy (see page 227)

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 (see page 225) or you can delete it (see page 226).

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 (see page 226).

   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 (see page 225).

Manual Event Policy Customization

When you require 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 situations or unsupported use cases occurs, use the functionality provided in the Event Policy dialog to define and deploy event policies.

To refine the event policy manually, edit or create the appropriate event policy files using the <Evaluate> policy operation and embedded Drools rules.

For information about how to define the <Evaluate> operations and Drools rules in event policy files, see Manually Refining Event Policy (see page 245).

Chapter 10: Event and Alert Management Deployment Scenarios

This section contains real-world scenarios for deploying event policies and creating alert queues and escalation policies.

This section contains the following topics:

Event Management Scenarios

Event Management lets you manipulate the event stream so that only the events that you manage appear as alerts in the Operations Console. Also, those alerts contain all information necessary for prompt assignment and resolution.

This section includes the following end-to-end Event Management scenarios, which include search patterns, policy creation, and action definition:

- Filtering duplicate events from integrated domain managers so that one alert represents a condition reported in multiple places in the Operations Console
- Creating an event that is based on a correlated group of events that indicate a service is crashing immediately after startup
- Combining a create event action with an enrichment action using the Reevaluate option to chain event policies
- Normalizing traps from the CA SystemEDGE agent so that the normalized traps appear in the Operations Console as alerts with the information necessary for resolution

Previous sections include standalone examples for event search patterns and event policies:

- Event search patterns
    - Time-based correlation (see page 158)
    - Occurrence frequency (see page 159)
    - Advanced search techniques (see page 159)
    - Raw events (see page 161)
    - Moving from simple to complex (see page 162)

- Event policies
  - Filter action (see page 171)
  - Create event action (see page 178)
  - Database enrichment (see page 188)
  - Java method enrichment (see page 195)
  - Script enrichment (see page 201)
  - Normalization (see page 213)

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
- CA Spectrum and CA NSM
- CA eHealth and CA NSM

For example, CA Spectrum might already be feeding its alarms into CA NSM when the two products are integrated. If you have CA Spectrum and CA NSM connectors installed, you could receive an alert for the original CA Spectrum alarm and an alert for the CA NSM 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 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, such as the CA NSM connector, 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 (see page 188) 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 (see page 212):

   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.

Appendix A: Manually Refining Event Policy

This section describes how to make manual refinements to event policy to implement functionality that the Event Policy dialog does not support.

This section contains the following topics:

How to Manually Refine Event Policy

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:

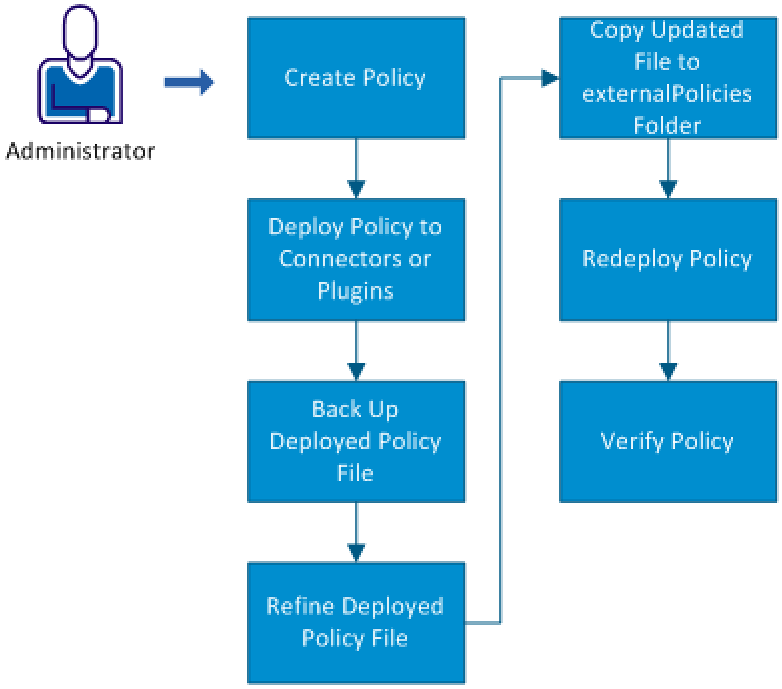
- Policies that require a combination of exclude and include filters
- Policies with an enrichment action that requires complex querying and enrichment assignments, involving operations such as SQL joins
- Policies that require more than three distinct search patterns
- Enrichment policies that are conditional 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:

How to Manually Refine Event Policy


Administrator
Create Policy
Deploy Policy to Connectors or Plugins
Back Up Deployed Policy File
Refine Deployed Policy File
Copy Updated File to externalPolicies Folder
Redeploy Policy
Verify 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 one of the SOI_HOME\resources\Core\Catalogpolicy\extensions directory 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 (see page 248) 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.

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 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, 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 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.

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.

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.

Glossary

alert

An *alert* is a message on the Operations Console that reports a fault condition that is associated with a resource or service.

alert escalation

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.

alert filter

An *alert filter* limits the number and alert types that are shown on the Operations Console.

alert queue

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.

Apache ActiveMQ

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

bidirectional connectors

A *bidirectional connector* supports both inbound and outbound operations.

CI

See *configuration item*.

configuration item (CI)

A *configuration item* (*CI*) is a managed resource such as a printer, software application, or database. Configuration items support services. A synonym for a configuration item is a *resource.*

Connector

A *connector* is software that provides the interface for the data exchange between the CA Catalyst infrastructure and a domain manager.

domain manager

A *domain manager* is a management application that provides information to CA Catalyst and CA SOI using a connector.

enrichment

An *enrichment* adds information to an event or alert from an outside source. An outside source can include a database, method, or script. You can add an enrichment to event policies to enrich matching events with important information.

escalation policy

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

event

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.

Event Management

Event Management provides a processing layer between raw connector USM alert data and alert management. You can influence the data that makes it into the Operations Console as alerts and how those alerts appear. Available Event Management actions include event correlation, filtering, creation, and enrichment.

event policy

An *event policy* is a combination of event search patterns and an action to perform when the patterns match. You can deploy event policies on all or specific sources, and available actions include filtering, event creation, and enrichment.

impact

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

inbound to connector operations

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.

JDBC

Java Database Connectivity is a programming interface that lets Java applications access an SQL database.

JMS (Java Message Service)

The *Java Message Service* (*JMS*) is a messaging standard that lets application components that are based on the Java 2 Platform Enterprise Edition (J2EE) create, send, receive, and read messages.

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. All events flow through the Mid-tier connector before reaching the Operations Console as alerts.

outbound from connector operations

Outbound from connector operations obtain data (such as services, CIs, topology, alerts, and status) from the source domain manager.

priority

Priority indicates the importance of a service to the business.

relationship

Relationships in a service model show how CIs are linked to form the service topology.

resource

A *resource* is a managed resource such as a printer, software application, or database. Resources support services. A synonym for a resource is a configuration item (CI).

root cause alert

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.

service

A *service* typically consists of several CIs, which are grouped to represent entities like web server farms or clusters. Services can also contain *subservices*, which are subordinate service models. Service models typically represent high-level abstract entities like a web-based retail transaction service, an application server service, or a source control service. You can define any service type with CA SOI as long as one of the integrated domain managers monitors the service components.

service model

A *service model* is a definition of a service or other entity in your enterprise. It is a logical grouping of resources, associations, dependencies, and policies.

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.

significance

Significance indicates the importance of a CI.

subservice

A *subservice* is used to indicate a subordinate service model.

synchronization

Synchronization is the CA Catalyst capability of updating source domain manager data due to CI changes. Changes are a result of reconciliation or other changes to data in the Persistent Store, including CI creation and deletion.

Unified Service Model (USM)

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