IBM Security Access Manager for Web
Version 7.0

Auditing Guide

SC23-6511-02

IBM Security Access Manager for Web
Version 7.0

Auditing Guide

SC23-6511-02

Note

Before using this information and the product it supports, read the information in “Notices” on page 419.

Edition notice

Note: This edition applies to version 7, release 0, modification 0 of IBM Security Access Manager (product number 5724-C87) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Part 7. Troubleshooting 351

Chapter 25. Problem determination 353

Part 8. Appendixes 369

Appendix A. Routing files 371

Appendix B. Configuration stanzas 375

Appendix C. Commands and utilities 405

Notices 419

Index 423

Figures

Tables

About this publication

IBM Security Access Manager for Web, formerly called IBM Tivoli Access Manager for e-business, is a user authentication, authorization, and web single sign-on solution for enforcing security policies over a wide range of web and application resources.

The *IBM Security Access Manager for Web Auditing Guide* provides conceptual, procedural, and reference information for the auditing operations of Security Access Manager using native Security Access Manager auditing and the Common Audit Service.

Intended audience

This publication is for system administrators who must perform the following auditing tasks:

- Installing and configuring the Common Audit Service
- Configuring and generating audit reports

Readers must be familiar with the following topics:

- Microsoft Windows, AIX, Linux, and Solaris operating systems
- Database architecture and concepts
- Security management
- Authentication and authorization
- Security Access Manager security model and its capabilities

Access to publications and terminology

This section provides:

- A list of publications in the "IBM Security Access Manager for Web library."
- Links to "Online publications" on page xv.
- A link to the "IBM Terminology website" on page xv.

IBM Security Access Manager for Web library

The following documents are in the IBM Security Access Manager for Web library:

- *IBM Security Access Manager for Web Quick Start Guide*, GI11-9333-01

  Provides steps that summarize major installation and configuration tasks.
- *IBM Security Web Gateway Appliance Quick Start Guide* – Hardware Offering

  Guides users through the process of connecting and completing the initial configuration of the WebSEAL Hardware Appliance, SC22-5434-00
- *IBM Security Web Gateway Appliance Quick Start Guide* – Virtual Offering

  Guides users through the process of connecting and completing the initial configuration of the WebSEAL Virtual Appliance.
- *IBM Security Access Manager for Web Installation Guide*, GC23-6502-02

  Explains how to install and configure Security Access Manager.
- *IBM Security Access Manager for Web Upgrade Guide*, SC23-6503-02

Provides information for users to upgrade from version 6.0, or 6.1.x to version 7.0.

- *IBM Security Access Manager for Web Administration Guide*, SC23-6504-02

  Describes the concepts and procedures for using Security Access Manager. Provides instructions for performing tasks from the Web Portal Manager interface and by using the **pdadmin** utility.
- *IBM Security Access Manager for Web WebSEAL Administration Guide*, SC23-6505-02

  Provides background material, administrative procedures, and reference information for using WebSEAL to manage the resources of your secure Web domain.
- *IBM Security Access Manager for Web Plug-in for Web Servers Administration Guide*, SC23-6507-02

  Provides procedures and reference information for securing your Web domain by using a Web server plug-in.
- *IBM Security Access Manager for Web Shared Session Management Administration Guide*, SC23-6509-02

  Provides administrative considerations and operational instructions for the session management server.
- *IBM Security Access Manager for Web Shared Session Management Deployment Guide*, SC22-5431-00

  Provides deployment considerations for the session management server.
- *IBM Security Web Gateway Appliance Administration Guide*, SC22-5432-00

  Provides administrative procedures and technical reference information for the WebSEAL Appliance.
- *IBM Security Web Gateway Appliance Configuration Guide for Web Reverse Proxy*, SC22-5433-00

  Provides configuration procedures and technical reference information for the WebSEAL Appliance.
- *IBM Security Web Gateway Appliance Web Reverse Proxy Stanza Reference*, SC27-4442-00

  Provides a complete stanza reference for the IBM® Security Web Gateway Appliance Web Reverse Proxy.
- *IBM Security Access Manager for Web WebSEAL Configuration Stanza Reference*, SC27-4443-00

  Provides a complete stanza reference for WebSEAL.
- *IBM Global Security Kit: CapiCmd Users Guide*, SC22-5459-00

  Provides instructions on creating key databases, public-private key pairs, and certificate requests.
- *IBM Security Access Manager for Web Auditing Guide*, SC23-6511-02

  Provides information about configuring and managing audit events by using the native Security Access Manager approach and the Common Auditing and Reporting Service. You can also find information about installing and configuring the Common Auditing and Reporting Service. Use this service for generating and viewing operational reports.
- *IBM Security Access Manager for Web Command Reference*, SC23-6512-02

  Provides reference information about the commands, utilities, and scripts that are provided with Security Access Manager.
- *IBM Security Access Manager for Web Administration C API Developer Reference*, SC23-6513-02

Provides reference information about using the C language implementation of the administration API to enable an application to perform Security Access Manager administration tasks.

- *IBM Security Access Manager for Web Administration Java Classes Developer Reference*, SC23-6514-02

  Provides reference information about using the Java™ language implementation of the administration API to enable an application to perform Security Access Manager administration tasks.
- *IBM Security Access Manager for Web Authorization C API Developer Reference*, SC23-6515-02

  Provides reference information about using the C language implementation of the authorization API to enable an application to use Security Access Manager security.
- *IBM Security Access Manager for Web Authorization Java Classes Developer Reference*, SC23-6516-02

  Provides reference information about using the Java language implementation of the authorization API to enable an application to use Security Access Manager security.
- *IBM Security Access Manager for Web Web Security Developer Reference*, SC23-6517-02

  Provides programming and reference information for developing authentication modules.
- *IBM Security Access Manager for Web Error Message Reference*, GI11-8157-02

  Provides explanations and corrective actions for the messages and return code.
- *IBM Security Access Manager for Web Troubleshooting Guide*, GC27-2717-01

  Provides problem determination information.
- *IBM Security Access Manager for Web Performance Tuning Guide*, SC23-6518-02

  Provides performance tuning information for an environment that consists of Security Access Manager with the IBM Tivoli Directory Server as the user registry.

Online publications

IBM posts product publications when the product is released and when the publications are updated at the following locations:

IBM Security Access Manager for Web Information Center

The http://pic.dhe.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.isam.doc_70/welcome.html site displays the information center welcome page for this product.

IBM Publications Center

The http://www-05.ibm.com/e-business/linkweb/publications/servlet/pbi.wss site offers customized search functions to help you find all the IBM publications that you need.

IBM Terminology website

The IBM Terminology website consolidates terminology for product libraries in one location. You can access the Terminology website at http://www.ibm.com/software/globalization/terminology.

Related publications

This section lists the IBM products that are related to and included with the Security Access Manager solution.

Note: The following middleware products are not packaged with IBM Security Web Gateway Appliance.

IBM Global Security Kit

Security Access Manager provides data encryption by using Global Security Kit (GSKit) version 8.0.x. GSKit is included on the *IBM Security Access Manager for Web Version 7.0* product image or DVD for your particular platform.

GSKit version 8 includes the command-line tool for key management, GSKCapiCmd (**gsk8capicmd_64**).

GSKit version 8 no longer includes the key management utility, iKeyman (**gskikm.jar**). iKeyman is packaged with IBM Java version 6 or later and is now a pure Java application with no dependency on the native GSKit runtime. Do not move or remove the bundled *java*/jre/lib/gskikm.jar library.

The *IBM Developer Kit and Runtime Environment, Java Technology Edition, Version 6 and 7, iKeyman User's Guide for version 8.0* is available on the Security Access Manager Information Center. You can also find this document directly at:

> http://download.boulder.ibm.com/ibmdl/pub/software/dw/jdk/security/60/iKeyman.8.User.Guide.pdf

Note:

GSKit version 8 includes important changes made to the implementation of Transport Layer Security required to remediate security issues.

The GSKit version 8 changes comply with the Internet Engineering Task Force (IETF) Request for Comments (RFC) requirements. However, it is not compatible with earlier versions of GSKit. Any component that communicates with Security Access Manager that uses GSKit must be upgraded to use GSKit version 7.0.4.42, or 8.0.14.26 or later. Otherwise, communication problems might occur.

IBM Tivoli Directory Server

IBM Tivoli Directory Server version 6.3 FP17 (6.3.0.17-ISS-ITDS-FP0017) is included on the *IBM Security Access Manager for Web Version 7.0* product image or DVD for your particular platform.

You can find more information about Tivoli Directory Server at:

> http://www.ibm.com/software/tivoli/products/directory-server/

IBM Tivoli Directory Integrator

IBM Tivoli Directory Integrator version 7.1.1 is included on the *IBM Tivoli Directory Integrator Identity Edition V 7.1.1 for Multiplatform* product image or DVD for your particular platform.

You can find more information about IBM Tivoli Directory Integrator at:

http://www.ibm.com/software/tivoli/products/directory-integrator/

IBM DB2 Universal Database™

IBM DB2 Universal Database Enterprise Server Edition, version 9.7 FP4 is provided on the *IBM Security Access Manager for Web Version 7.0* product image or DVD for your particular platform. You can install DB2® with the Tivoli Directory Server software, or as a stand-alone product. DB2 is required when you use Tivoli Directory Server or z/OS® LDAP servers as the user registry for Security Access Manager. For z/OS LDAP servers, you must separately purchase DB2.

You can find more information about DB2 at:

http://www.ibm.com/software/data/db2

IBM WebSphere® products

The installation packages for WebSphere Application Server Network Deployment, version 8.0, and WebSphere eXtreme Scale, version 8.5.0.1, are included with Security Access Manager version 7.0. WebSphere eXtreme Scale is required only when you use the Session Management Server (SMS) component.

WebSphere Application Server enables the support of the following applications:

- Web Portal Manager interface, which administers Security Access Manager.
- Web Administration Tool, which administers Tivoli Directory Server.
- Common Auditing and Reporting Service, which processes and reports on audit events.
- Session Management Server, which manages shared session in a Web security server environment.
- Attribute Retrieval Service.

You can find more information about WebSphere Application Server at:

http://www.ibm.com/software/webservers/appserv/was/library/

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

Visit the IBM Accessibility Center for more information about IBM's commitment to accessibility.

Technical training

For technical training information, see the following IBM Education website at http://www.ibm.com/software/tivoli/education.

Support information

IBM Support provides assistance with code-related problems and routine, short duration installation or usage questions. You can directly access the IBM Software Support site at http://www.ibm.com/software/support/probsub.html.

The *IBM Security Access Manager for Web Troubleshooting Guide* provides details about:

- What information to collect before you contact IBM Support.
- The various methods for contacting IBM Support.
- How to use IBM Support Assistant.
- Instructions and problem-determination resources to isolate and fix the problem yourself.

Note: The **Community and Support** tab on the product information center can provide more support resources.

Part 1. Introduction

Chapter 1. Introduction to auditing

Auditing is the process of maintaining detailed, secure logs of critical activities in a business environment. These activities can be related to security, content management, business transactions, or other such activities.

For example, the following activities can be audited:

- Login failures
- Unauthorized access to protected resources
- Modification to security policy

Security Access Manager provides two methods for managing audit events. One method uses the native Security Access Manager approach, and the other method uses the Common Auditing and Reporting Service.

- Use the method provided in Part 4, “Common Auditing Service auditing,” on page 135 to manage audit events with the Common Auditing and Reporting Service.
- Use the method provided in Part 5, “Native Security Access Manager auditing,” on page 171 to manage audit events with the native Security Access Manager approach.

For information about managing statistical events, see Chapter 21, “Working with statistics,” on page 205. For information about WebSEAL HTTP events, see Chapter 20, “WebSEAL HTTP logging,” on page 197.

Auditing versus diagnostics

Security Access Manager provides ways to collect events that you can use for diagnostic and auditing purposes of the servers. Events for diagnostics and auditing pertain to the operations of the servers.

To enable diagnostics and auditing, define which types of events to capture. You can write recorded events to one or a combination of the following files or devices:

- Log file.
- Standard output (STDOUT) device.
- Standard error (STDERR) device.

Beyond these destinations, when events are captured, they can be redirected to a remote authorization server or redirected to an application for processing. When using the Common Auditing and Reporting Service, audit events are written to the XML data store for processing.

During the installation of Security Access Manager servers, installation logs capture messages for the installation.

When using the installation wizard, each server has its own log file. When using a native installation, the installation uses its operating system-specific logs.

For information about installation logs, except for the installation of the Common Auditing and Reporting Service, see the *IBM Security Access Manager for Web*

Troubleshooting Guide. Information about installing, configuring, and uninstalling the Common Auditing and Reporting Service is contained in this document.

Audit events

For auditing purposes, define which audit, statistic, or other type of events to capture.

You can use events to create snapshots of various server activities. You can record audit events by using either the native Security Access Manager approach or Common Auditing and Reporting Service.

To configure auditing events, define stanza entries in the configuration files. Depending on your approach, you define different stanza entries in different configuration files.

When you enable the Common Auditing Service, use the following guidelines for defining the auditing configuration:

- For audit events that you want to record by using the Common Auditing Service, define entries in the `[cars-filter]` stanza of the server-specific `pdaudit.conf` configuration file. When events are sent to the Common Auditing Service audit server, you can generate and view operational reports through a reporting interface.
- For audit events that you want to record by using the native Security Access Manager mechanisms, define entries in the `[pdaudit-filter]` stanza of the server-specific `pdaudit.conf` configuration file.
- For HTTP request events, define entries in the `[aznapi-configuration]` and `[logging]` stanzas of the WebSEAL configuration files.

If you do not enable the Common Auditing Service, use the following guidelines for defining the auditing configuration:

- For audit events, define `logcfg` entries in the `[aznapi-configuration]` stanza of the server configuration file.
- For HTTP request events, define entries in the `[aznapi-configuration]` and `[logging]` stanzas of the WebSEAL configuration files for HTTP events that you want to record.

Diagnostic events

For diagnostic information, define which message events and which trace events to capture. These events can help you troubleshoot problems.

To configure diagnostic events, you define statements in the server-specific routing files. Each server has an associated routing file. The statements in these routing files are for both message events and trace events. You define the statements for message events by severity level. You define the statements for trace events by trace level and optionally by component.

For additional information about message and trace events, see the *IBM Security Access Manager for Web Troubleshooting Guide*.

Audit trails

IT organizations can use information contained in audit trails to help them show compliance with government regulations such as the following regulations:

- Sarbanes-Oxley (SOX) Act.
- The Health Insurance Portability and Accountability Act (HIPAA).
- The Basel II international banking accord.

For these reasons, such audit trails must be sometimes maintained for years.

Audit trails are useful to check enforcement and effectiveness of IT controls, for accountability and vulnerability, and for risk analysis. IT organizations can also use auditing of security-related critical activities to aid in forensic investigations of security incidents.

When a security incident occurs, audit trails enable analysis of the history of activities that occurred before the security incident. This analysis might answer questions such as who did what, when, where, and how. Based on this analysis, appropriate corrective actions can be taken. For these reasons, audit trails must be archived and accessible for years.

Audit trails can be established in relational databases that are easily queried to generate reports. When audit trails are written to relational databases, reporting tools, such as Tivoli® Common Reporting, can be used to display reports. Reports can fall into the following categories:

- Trend reports provide summarized audit data that you can use to assess whether there is any long-term rise or fall in questionable activity. Trend reports can help provide a "security pulse" for an organization.
- Operational reports allow a detailed review of audit data to help determine the cause of a security incident.

Audit records for HTTP access

The generation of audit records for HTTP access to WebSEAL can consume large quantities of disk space quickly. You can reduce the volume of audit events that are generated by using the following strategies:

- Generate events for unsuccessful HTTP accesses only.
- Selectively disable the generation of events by using attached protected object policies (POPs).

For details about reducing records by generating events for unsuccessful accesses only, see "Native auditing" on page 13 if you are using native Security Access Manager auditing, or see "WebSEAL: Configuration settings" on page 143 if you are using the Common Auditing Service.

For details about using POPs to selectively disable the generation of audit events, see "Disabling resource access events" on page 194. This approach applies to both native Security Access Manager auditing and the Common Auditing Service.

Chapter 2. Overview of the Common Auditing Service

The Common Auditing Service can be used to provide auditing for your environment.

Note: Common Auditing Service is the same feature as the Common Auditing and Reporting Service. The name changed to indicate that exploiting products now provide the reporting functionality. Common Auditing Service provides only the utilities that you can use to manage the tables used to create reports.

The Common Auditing Service consists of the following parts:

Audit server (Common Auditing Service server)
The audit server component is used by all using products. It is also called the "audit service."

Note: Previous releases of Common Auditing Service used the term "event server" instead of "audit server."

Client The Common Auditing Service clients are embedded in the product; the clients are not separately installable components.

Embedded C client
This component is packaged as a set of libraries that is included in Security Access Manager.

Embedded Java client
This component is packaged as a set of JAR files and includes the security event factory and emitter. It is available with Tivoli Federated Identity Manager and Security Access Manager.

Staging utility and XML data store utilities
You can use these utilities to manage operations for the XML audit and report staging databases, such as staging data for reporting and purging inactive tables.

Figure 1 on page 8 illustrates how data flows between the components of Common Auditing Service and Security Access Manager.

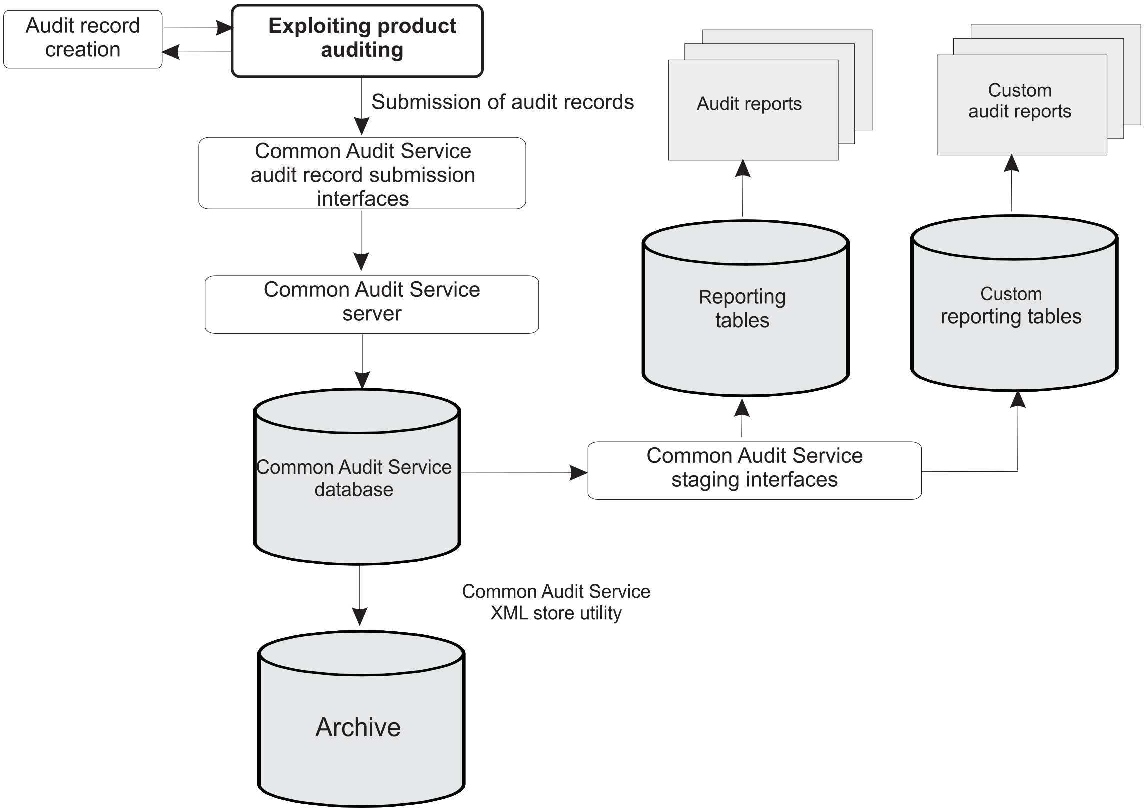
Audit record creation
Exploiting product auditing
Submission of audit records
Common Audit Service audit record submission interfaces
Common Audit Service server
Common Audit Service database
Common Audit Service XML store utility
Archive
Common Audit Service staging interfaces
Reporting tables
Custom reporting tables
Audit reports
Custom audit reports

Figure 1. Structure of Common Auditing Service

Common Auditing Service infrastructure

The Common Auditing Service infrastructure provides the mechanisms to submit, centrally collect, and persistently store and report on audit data.

Common Auditing Service uses the Common Base Event format, which is a standard, XML-based format that defines the structure of a security event. The Common Auditing Service Security Event Factory allows for the generation of events that conform to the Common Base Event security event specifications.

Common Auditing Service enables the storing of security events in an XML data store, which you specify during the configuration of Common Auditing Service.

Common Auditing Service also provides:

- Staging utility to stage the data from the XML data store into report tables. You can generate and create audit reports based on the audit events that are staged into report tables.
- XML store utility to help you manage the XML data store in preparation for archiving, and to clean up restored data that is no longer needed.
- Support for the lifecycle of audit events, including archiving and the restoration of archived event data.

Reporting

Common Auditing Service stores audit data in the XML data store and provides utilities to manage this data; however, Common Auditing Service does not include utilities for creating formatted reports. Security Access Manager uses Tivoli Common Reporting to generate, format, view, and print report data. Tivoli Common Reporting integrates open source reporting interfaces into a common tool that provides a consistent appearance and improves the quality of the report content.

Security Access Manager also provides a set of report definition files, known as a *report package*, for the Tivoli Common Reporting environment. You can use these report templates to generate out-of-box operational reports, including audit history and details, password administration, authorization history and details, and resource access.

Auditing and reporting scenarios

This section provides the following scenarios for the collection and use of audit data:

- Security investigation
- IT control
- Compliance

Security incident investigation scenario

The following scenario shows how audit data can be used to investigate break-in security incidents.

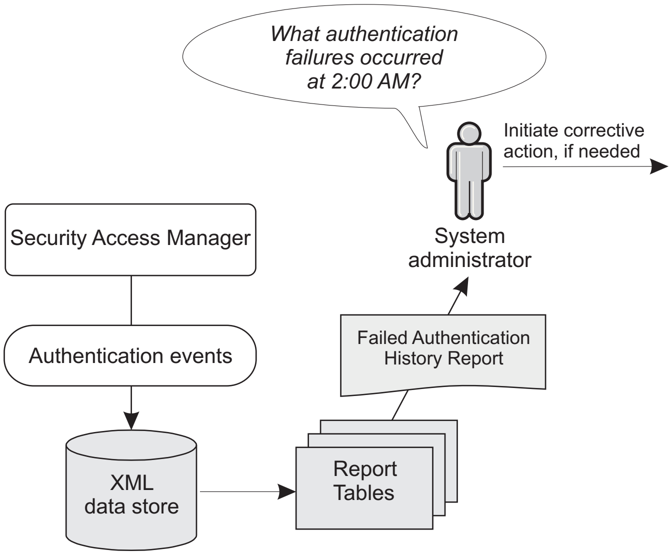
What authentication failures occurred at 2:00 AM?
Initiate corrective action, if needed
System administrator
Security Access Manager
Authentication events
Failed Authentication History Report
XML data store
Report Tables

Figure 2. Security incident investigation scenario

IT control scenario

The following scenario shows how audit data can be used to ensure that only authorized entities are accessing protecting resources.

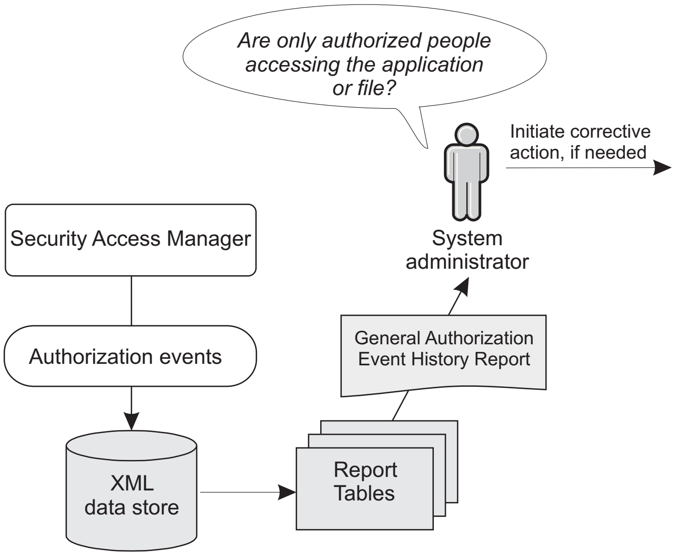
Are only authorized people accessing the application or file?
Initiate corrective action, if needed
System administrator
Security Access Manager
Authorization events
General Authorization Event History Report
XML data store
Report Tables

Figure 3. IT control scenario

Compliance scenario

The following scenario shows how audit data can be used to demonstrate compliance to a security policy.

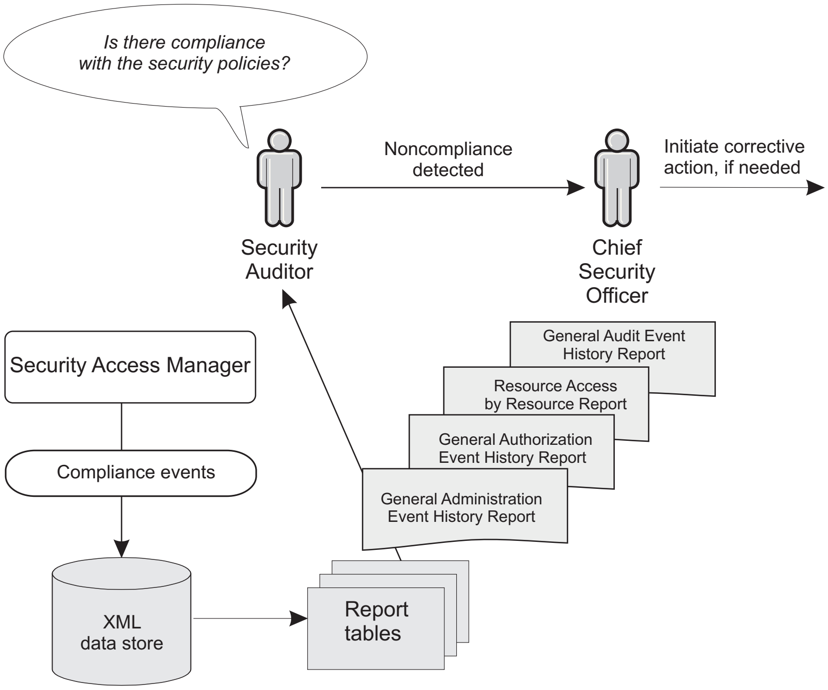
Is there compliance with the security policies?
Noncompliance detected
Initiate corrective action, if needed
Security Auditor
Chief Security Officer
Security Access Manager
General Audit Event History Report
Resource Access by Resource Report
General Authorization Event History Report
Compliance events
General Administration Event History Report
XML data store
Report tables

Figure 4. Compliance scenario

Procedure for collecting audit data

The following tasks describe an overall procedure for collecting audit data to generate a report.

Procedure

1. Identify the installed IBM security software. For example, you might have Security Access Manager and Tivoli Federated Identity Manager in your environment.
2. Identify the type of events to audit. For example, to report Security Access Manager login activity and the effectiveness of initial policy, configure Security Access Manager to send the following Common Auditing Service events:
   - AUDIT_AUTHN
   - AUDIT_MGMT_POLICY

   To report on trust events for Tivoli Federated Identity Manager, you must configure Tivoli Federated Identity Manager to send the IBM_SECURITY_TRUST event.

   See the configuration information of each exploiting product for instructions on setting up the recording of specific types of events.
3. Determine the volume of security events per day.

   The volume of security events generated per day determines the type of reports, frequency of staging, and so on. For example, if you are generating events in millions, archive and prune archived data frequently.

   When you prune frequently, be aware that this limits your ability to run security events details reports. Also, many security events increases the time to stage data and forces you to schedule report generation much later. In addition, you must select start and end time parameters for the reports so that the number of security events returned is approximately 100,000.

   For the purposes of this scenario, we will assume that the number of security events is 100,000 per day.
4. Perform the data management tasks:
   a. Stage the data for generating reports. You must stage the events from the XML data store tables to report tables. You can run the staging utility in incremental mode every day shortly after midnight, for example, at 12:05 AM. For more information, see "Running the staging utility command" on page 66.

   b. Archive the data from the XML data store. Depending on the volume of security events, run the archive process once or twice a week.

      The archival process consists of four phases:
      - Pre-archive
      - Archive by using an archival tool
      - Prune the data from the report tables
      - Post-archive

      Use the XML data store utilities for these processes. For information about these utilities, see "Running the XML data store utilities" on page 67.

   c. Prune the report data from the report tables. Because post-archive removes security events from the XML data store, run the staging utility in prune mode to remove corresponding security events from the report tables. You will use the first timestamp from the pre-archive phase as input. For more information, see "Running the staging utility command" on page 66.
5. Generate reports by using the report-generation tool of the exploiting product, for example, Security Access Manager or Tivoli Federated Identity Manager. The following scenario shows a process to generate a report for a security incident investigation:
   - A security incident investigation is needed to determine who logged in between 2:00 AM and 6:00 AM.

- Use your reporting tool to run an audit security events history. Configure parameters such as start date and time, end date and time, event type, number of events, product name, sort criteria, and so on, to review the events in question, and to display the report in a useful format.
- Run the report for the day in which you are interested. The start time will be 2:00 AM and end time will be 6:00 AM.

Chapter 3. Overview of Security Access Manager event logging

For auditing and other serviceability purposes, Security Access Manager uses a structured hierarchy of events. This hierarchy is built dynamically and allows runtime-associations to be made between event categories and the log agents that record those events.

Figure 5 shows the hierarchy of Security Access Manager events in the event pool.

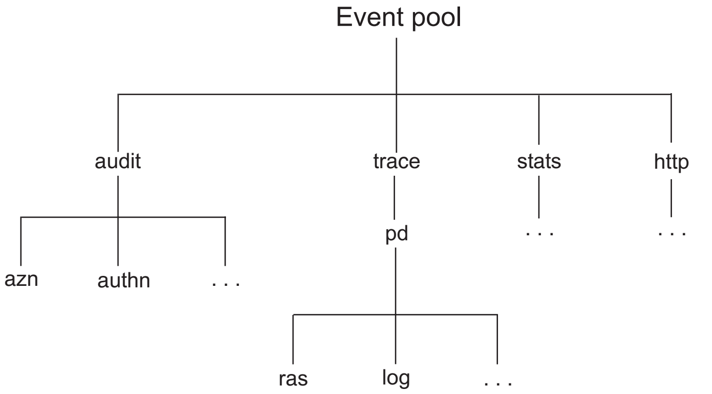

Figure 5. Event pool hierarchy

Natively, Security Access Manager generates and can record the following primary categories of events:

Audit events
For information about audit events, see Chapter 19, "Audit event logging," on page 173.

HTTP request events
For information about HTTP request events, see Chapter 20, "WebSEAL HTTP logging," on page 197.

Statistical events
For information about statistical events, see Chapter 21, "Working with statistics," on page 205.

Trace events
For information about trace events, see *IBM Security Access Manager for Web Troubleshooting Guide*.

Native auditing

Auditing is defined as the logging of audit records. It includes the collection of data about system activities that affect the secure operation of the Security Access Manager server processes. Each Security Access Manager server can capture audit events whenever any security-related auditable activity occurs.

Auditing uses the concepts of a record, an audit event, and an audit trail. Each audited activity is called an *audit event*. The output of a specific server event is called a *record*. An *audit trail* is a collection of multiple records that document the server activity.

When configuring for auditing, think about the source of the events that you want to capture. Audit trail files can capture authorization, authentication, and management events that are generated by the Security Access Manager servers. There are multiple sources for auditing events that you want to gather. You can collect either a combination or all the different types of auditing events at the same time. Table 1 shows some of the event types that can be used for native auditing.

Table 1. Categories and description of native audit events

Event category	Description
audit.authz	Authorization events for WebSEAL servers
audit.azn	Authorization events for base servers
audit.authn	Authentication, credential acquisition authentication, password change, and logout events
audit.authn.successful	Successful authentication credential acquisition authentication, password change, and logout events
audit.authn.unsuccessful	Failed authentication credential acquisition authentication, password change, and logout events
audit.http	HTTP access events
audit.http.successful	Successful HTTP access events
audit.http.unsuccessful	Failed HTTP access events
audit.mgmt	Management events
http	HTTP logging information
http.clf	HTTP request information defined by the request-log-format configuration entry in the `[logging]` stanza. `clf` stands for common log format.
http.ref	HTTP Referrer header information
http.agent	HTTP User Agent head information

Statistics gathering

Security Access Manager servers provide a series of modules that can monitor and collect information about specific server activity. After enabling a module, you can display the statistical information that it gathered since it was enabled. In addition to displaying this information, you can direct these statistics to a log file.

You can work with statistics with the **`server task stats`** command or with stanza entries in the configuration file for the specific server.

When you display statistics, you see a snapshot of the statistics. These statistics provide a view of the recorded activity. If you capture statistics at regular intervals, you can determine trend analyses against the server activities.

For information about enabling and working with the statistics gathering modules, see Chapter 21, “Working with statistics,” on page 205.

Logging process

Figure 6 depicts the relationships among the steps in the logging process. The top part of the figure represents the code of a Security Access Manager server. The code contains probe points where events of specific types can be generated. Generated events are submitted to the server event pool for possible recording through a point of capture (event sink). The event pool defines the events category.

At run time, you can subscribe a log agent at any point in the event pool hierarchy. You can selectively record events that are generated at the probe points for the program. The middle part of the figure depicts subscription.

For example, you can subscribe to a remote client for capturing events. This client forwards the selected events to a remote authorization server.

The lower part of the figure depicts this remote server. Relayed events are placed in the event pool at the remote probe points for the authorization server.

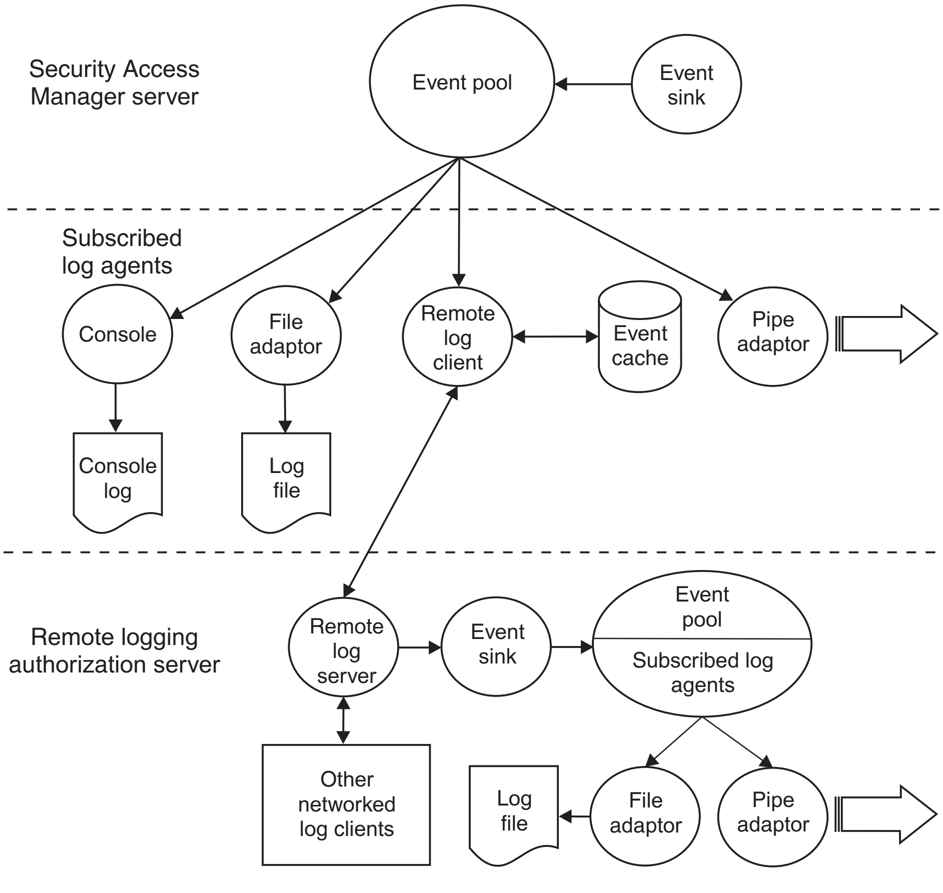

Figure 6. Application-specific probe points

Audit data in UTF-8 format

Security Access Manager produces audit data that uses UTF-8 encoding. When the operating system uses a non-UTF-8 code page, Security Access Manager converts the data to a format that matches the non-UTF-8 code page. In some cases, the conversion can result in data loss. For this reason, run Security Access Manager in an environment that uses UTF-8 encoded code pages.

When the operating system does not use a UTF-8 code page, the conversion to UTF-8 can result in data loss. When data loss occurs, the log file contains a series of question mark (?) characters at the location where the data conversion was problematic.

When running in a non-UTF-8 locale, use the `UTF8FILE` type in the routing file. For more information about the `UTF8FILE` type, see Appendix A, "Routing files," on page 371.

Chapter 4. Globalization

This topic describes the globalization features for Common Auditing and Reporting Service.

This section contains the following topics:
- "Language support overview"
- "Installing language support packages" on page 18
- "Uninstalling language support packages" on page 19
- "Locale environment variables" on page 19
- "Message catalogs" on page 20
- "Text encoding (code set) support" on page 21

Attention: Review the globalization section in the *IBM Security Access Manager for Web Release Notes* or Technotes in the support knowledge database for any language-specific limitations or restrictions.

Language support overview

The Common Auditing and Reporting Service software is translated into the following languages:
- Brazilian Portuguese
- Czech
- Chinese (Simplified)
- Chinese (Traditional)
- French
- German
- Hungarian
- Italian
- Japanese
- Korean
- Polish
- Spanish
- Russian

Note:
1. The installation wizard uses your language of choice, without installing the language pack.
2. The installation wizards and the Windows native installation utility do *not* support the bidirectional languages (for example, Arabic and Hebrew).

The translations for these languages are provided as language support packages on the *IBM Security Access Manager for Web Version 7.0* DVD. To obtain language support for Common Auditing and Reporting Service, you must install the language support package for that product. Each language is a separately installable product installation image.

You must install the language package *after* installing the Common Auditing Service components but *before* configuring them. If you do not install the language support package, the associated product displays all text in English.

If language support is installed and you upgrade the product, you must also install the corresponding language support product, if one exists. If you do not install the language support after upgrading, the associated product might display some fields and messages in English.

Installing language support packages

To install language support packages for Security Access Manager, follow these steps:

Procedure

1. Log on as **root** or as an administrative user.
2. Insert or mount the *IBM Security Access Manager for Web Version 7.0* DVD for the appropriate operating system, and change to the root directory where the DVD is located.
3. Ensure that IBM Java Runtime 1.6, provided with Security Access Manager, is installed for your particular operating system. For instructions, see the *IBM Security Access Manager for Web Installation Guide*.

   The language pack message files, by language, for IBM Java Runtime 1.6 include:

   ```
   Java6.msg.Ja_JP
   Java6.msg.Zh_CN
   Java6.msg.Zh_TW
   Java6.msg.ja_JP
   Java6.msg.ko_KR
   Java6.msg.zh_CN
   Java6.msg.zh_TW
   ```
4. Depending on the component that you want to install, run one or more of the following setup scripts.
   - To install with Launchpad, select the scripts for the required components.

     Note:

     a. Scripts are used for AIX, Linux, and Solaris operating systems; batch files (.bat extension) are used for Windows operating systems.
     b. If you issue a script without specifying the *jre_path*, you must ensure that the Java executable program is part of the PATH statement. Otherwise, issue the script that specifies the *jre_path* as follows:

        ```
        language_package jre_path
        ```

        To install the language package for Common Auditing and Reporting Service, enter the following command:

        ```
        install_cars_lp /usr/bin
        ```

        where `install_cars_lp` installs the language packages for Common Auditing and Reporting Service, and `/usr/bin` is the path to the JRE.
   - To install in console mode, complete the following steps:
   - Ensure that the IBM Java Runtime 1.6 is available in the command execution path (or prefix the command with the JRE directory.)
   - Run the following command:

     ```
     java -jar language_package.jar run -console
     ```

     where *language_package* is the name of the language package to install:

`carslp.jar`
: Installs the language packages for Common Auditing and Reporting Service.

5. Click **Next** to begin installation. The Software License Agreement window is displayed.
6. To accept the license agreement, select the **I accept** check box to accept the terms.
7. Click **Next** to display a dialog that shows a list of the languages.
8. Select the language packages that you want to install.
9. Click **Next** to display a dialog that shows the location and features of the languages that you selected.
10. Click **Next** to accept the languages selected.

    The installation wizard validates that sufficient disk space is available.
11. Click **Next** to install the languages that you selected.
12. After installation for the language pack completes successfully, click **Finish** to close the wizard and restart your system.

Uninstalling language support packages

Uninstall the language packs if you no longer need the language support for your environment.

Procedure

1. Change to one of the following directories:

   AIX®, Linux, and Solaris operating systems:
   ```
   /opt/IBM/Tivoli/CommonAuditService/CARSLP/lp_uninst
   ```

   Windows operating systems:
   ```
   C:\Program Files\IBM\Tivoli\CommonAuditService\CARSLP\lp_uninst
   ```
2. Uninstall the language support packages with the following command:

   AIX, Linux, and Solaris operating systems:
   ```
   jre_path/java -jar cars_lp_uninstall.jar
   ```

   Windows operating systems:
   ```
   jre_path\java -jar cars_lp_uninstall.jar
   ```

   where *jre_path* is the path where the Java executable program is located. If the Java executable program is in the path, you do not have to specify *jre_path*.

Locale environment variables

As with most current operating systems, localized behavior is obtained by specifying the required locale. You set the LANG environment variable to the required locale name as specified by POSIX, X/Open, or other open systems standards.

Note: If you are in a Windows environment, you can alternatively modify the language setting in the **Regional Settings** of the **Control Panel**.

If you specify the LANG environment variable and modify the regional settings, the LANG environment variable overrides this regional setting.

As specified by open systems standards, other environment variables override LANG for some or all locale categories. These overriding variables include the following variables:

- LC_COLLATE
- LC_CTYPE
- LC_MONETARY
- LC_NUMERIC
- LC_TIME
- LC_MESSAGES
- LC_ALL

If any of the previous variables are set, you must remove their setting for the LANG variable to have full effect.

LANG variable on AIX, Linux, or Solaris systems

Most AIX, Linux, and Solaris operating systems use the LANG variable to specify the required locale. Different AIX, Linux, and Solaris operating systems require different locale names to specify the same language. Ensure that you use a value for LANG that is supported by the AIX, Linux, or Solaris operating system that you are using.

To obtain the locale names for your AIX, Linux, or Solaris operating system, enter the following command:

```
locale –a
```

LANG variable on Windows systems

Most Windows operating systems do not use the LANG environment variable. However, you can use LANG to determine the required language. To do so, set the LANG environment variable to the canonical locale name based on the ISO language or territory codes without a code set suffix. For example:

- `fr` is the locale for standard French
- `ja` is the locale for Japanese
- `pt_BR` is the locale for Brazilian Portuguese
- `C` is the locale for English in C locale

Using locale variants

Although Security Access Manager software currently provides only one translated version for each language, you can use a preferred locale variant, and Security Access Manager finds the corresponding language translation. For example, Security Access Manager provides one translation for French, but each of the following locale settings finds the appropriate translation:

- `fr` is the locale name for standard French
- `fr_FR` is the locale name for French in France
- `fr_CA` is the locale name for French in Canada
- `fr_CH` is the locale name for French in Switzerland

Message catalogs

Message catalogs are typically installed in a `/msg` subdirectory and each of these message catalogs is installed under a language-specific subdirectory. For example, the Security Access Manager base components use the following directories:

- On AIX, Linux, and Solaris operating systems:

```
/opt/PolicyDirector/nls/msg/locale
```

- On Windows operating systems:

```
install_dir/nls/msg/locale
```

Other Security Access Manager components use similar directories for their message catalogs.

Security Access Manager recognizes variations in AIX, Linux, or Solaris locale names and is typically able to map the specified value to the appropriate message catalog.

The NLSPATH environment variable is used to find the appropriate message catalog directory, as specified by open systems standards. For example, if the message catalogs are in `/opt/PolicyDirector/nls/msg`, the NLSPATH variable is set to:

```
/opt/PolicyDirector/nls/msg/%L/%N.cat:/opt/PolicyDirector/nls/msg/%L/%N
```

Note: For Windows, use a semicolon (;) instead of a (:) as the separator. For example:

```
C:\Program Files\PolicyDirector\nls\msg\%L\%N.cat;C:\Program
Files\PolicyDirector\nls\msg\%L\%N
```

The `%L` directive is expanded to the message catalog directory that most closely matches the current user language selection. Also, `%N.cat` expands to the required message catalog.

If a message catalog is not found for the required language, the English `C` message catalogs are used.

For example, suppose that you specify the AIX locale for German in Switzerland as follows:

```
LANG=De_CH.IBM-850
```

The `%L` directive is expanded in the following order to locate the specified locale:

1. `de_CH`
2. `de`
3. `C`

Because Security Access Manager does not provide a German in Switzerland language package, `de_CH` is not found. If the Security Access Manager German language package is installed, `de` is used. Otherwise, the default locale `C` is used, causing text to be displayed in English.

Text encoding (code set) support

Different operating systems often encode text in different ways. For example, Windows systems use `SJIS` (code page 932) for Japanese text, but AIX, Linux, or Solaris operating system often use `eucJP`.

In addition, you can provide multiple locales for the same language. By doing so, you can use different code sets for the same language on the same machine. Providing multiple locales for the same language can cause problems in one of the following situations:

- Text is moved from system to system.
- Text is moved between different locale environments.

Security Access Manager addresses these problems by using Unicode and UTF-8 (the multibyte form of Unicode) as the internal canonical representation for text.

Message catalogs are encoded by using UTF-8, and the text is converted to the locale encoding before being presented to the user. In this way, the same French message catalog files can be used to support various Latin 1 code sets, such as:

- ISO8859-1
- Microsoft 1252
- IBM PC 850
- IBM MVS™ 1047

UTF-8 is also used to achieve text interoperability. For example, Common Object Request Broker Architecture (CORBA) strings are transmitted as UTF-8. Doing so enables remote management within a heterogeneous network in which local text encoding can vary. For example, Japanese file names can be manipulated on Japanese PC endpoints from a desktop that executes in the AIX Japanese EUC locale.

Text interoperability across the secure domain is also achieved by storing strings as UTF-8 within the object database. Strings are converted to the local encoding for viewing and manipulation by applications that are executing on different operating system code sets.

Location of code set files

Interoperability across your secure domain depends on code set files, which are used to complete a UTF-8 conversion and other types of encoding-specific text processing.

These files are installed in the following directories:

- On AIX, Linux, and Solaris operating systems:
  `/opt/PolicyDirector/nls/TIS`
- On Windows operating systems:
  install_dir`\nls\TIS`

Part 2. Installing Common Auditing and Reporting Service

Chapter 5. Installing, configuring, and upgrading the Common Auditing Service audit server

This section describes the tasks used to install and configure the Common Auditing Service audit server.

Installing Common Auditing Service

This topic describes how to install the Common Auditing Service features.

The installation of Common Auditing Service involves the following tasks:

- Installing the prerequisite products
- Determining the target directory locations
- Ensuring that the requirements described in the preinstallation checklist are met before you start the installation
- Installing the Common Auditing Service components:

  Audit Service
  This installation component includes the Audit Service server (audit server), configuration utility, report staging utility, and XML data store (XMLSTORE) utility.

  Audit Configuration Console (configuration console)
  The Audit Configuration Console feature includes the files that implement the graphical configuration console.

To upgrade the Common Auditing Service audit server, see "Upgrading the Common Audit Service audit server" on page 57.

Installing prerequisite products

The Common Auditing Service requires the presence of other software products. Some of these products can be optionally installed after the Common Auditing Service is installed; but they *must* be installed before you start to configure the Common Auditing Service.

The Common Auditing Service audit server requires the following software. For details about the supported versions of the prerequisite software, see the *IBM Security Access Manager for Web Release Notes*.

IBM DB2 Server
The DB2 server is required to configure the audit server. The DB2 server does not need to be installed before you install the Common Auditing Service component.

WebSphere Application Server or WebSphere Application Server Network Deployment
WebSphere Application Server must be installed and operational before you start to install either the Common Auditing Service or the Common Auditing Service Configuration Console.

DB2 client (DB2 Client or DB2 Runtime Client)
If the DB2 server is installed on a separate system from the Common Auditing Service audit server, then the DB2 client must be installed on the

same system as the audit server. If required, the DB2 client must be installed before you *configure* the audit server, but not before you *install* the audit server. In a cluster environment, the DB2 client must be installed on each of the managed nodes.

If the DB2 server is DB2 Version 9.7, install the DB2 Client (or Runtime Client) Version 9.7. Run the **db2level** command on the DB2 server computer to determine the version of DB2 server that is running in your environment. See "Installing the DB2 client on Windows systems" or "Installing the DB2 Client on AIX, Linux, or Solaris systems" on page 27 for more instructions.

Installing the DB2 client on Windows systems

This section describes how to install the DB2 client on a Windows platform. This software is required if you want to run the DB2 server on a machine that is different from the WebSphere Application Server node where you are configuring the Audit Service component.

Procedure

1. Download the DB2 client for the appropriate DB2 server and the Windows platform from the following website:

   DB2 9.7 client (DB2 Client or DB2 Runtime Client)
   http://www.ibm.com/software/data/db2/udb/support/downloadv9.html
2. Locate the appropriate level of client in the table and download the setup file with either Download Director or FTP.
3. Follow the directions in the installation wizard to install the client.

What to do next

Note: If the database server is remote to the WebSphere Application Server node where configuration is taking place, at the node from the audit server system, use the following DB2 catalog command to add a TCP/IP node entry to the node directory.

The TCP/IP communications protocol is used to access the remote database node. Cataloging enables DB2 command-line access to the remote database server. In a cluster environment, configuration is completed on a Deployment Manager node in the WebSphere Application Server Network Deployment edition; otherwise, configuration is completed on a stand-alone server node.

```
db2 catalog tcpip node nodename remote hostname server service_name
```

where:

nodename
Specifies a local alias for the node to be cataloged.

hostname
Specifies the host name or the IP address of the node where the target database is located. The host name is the name of the node that is known to the Internet Protocol network. The maximum length of the host name is 255 characters.

service_name
Specifies the service name or the port number of the server database manager instance. The maximum length is 14 characters. This parameter is case-sensitive.

If a service name is specified, the services file on the client is used to map the service name to a port number. A service name is specified in the database manager configuration file of the server. The services file on the server is used to map this service name to a port number. The port number on the client and the server must match.

You must verify that the TCP/IP node is cataloged correctly. Run the following DB2 commands:

```
db2 attach to nodename user username using password
db2 list applications
db2 detach
```

Where:

nodename
Specifies the alias of the instance to which you want to attach.

username
Specifies the authentication identifier.

password
Specifies the password for the user name.

Installing the DB2 Client on AIX, Linux, or Solaris systems

This section describes how to install the DB2 Client on an AIX, Linux, or Solaris platform. This software is required if you want to run the DB2 server on a machine that is different from the WebSphere Application Server node where you are configuring the Audit Service component.

Procedure

1. Download the DB2 client for the appropriate AIX, Linux, or Solaris platform from the following website:

   DB2 9.7 client (DB2 Client or DB2 Runtime Client)
   http://www.ibm.com/software/data/db2/udb/support/downloadv9.html
2. Uncompress and extract the file.
3. Run **db2setup** that is in the admcl directory.
4. Select **Install Products**.
5. Select the radio button for the DB2 client that you are installing.
6. Click **Next** in the Welcome to the DB2 Setup wizard.
7. Click **I accept the terms in the license agreement** if you accept the terms of the license agreement.
8. Click **Next**
9. Select **Typical** installation type.
10. Click **Next**.
11. Select **Create a DB2 instance** in the Setup a DB2 instance window.
12. Select the **New User** radio button and specify a user name and password. You can either accept the defaults or change the settings that are appropriate for your environment. If you already created a user, you might want to also select the **Existing User** radio button and specify the user name.
13. Click **Next**.
14. Click **Finish**.

What to do next

Note: When the database server is remote to the WebSphere Application Server node where configuration is taking place, enter the following command at the node to add a TCP/IP node entry to the node directory:

```
db2 catalog tcpip node nodename remote hostname server service_name
```

The TCP/IP communications protocol is used to access the remote database node. Cataloging enables DB2 command-line access to the remote database server. In a clustered environment, configuration is completed on a Deployment Manager node in the WebSphere Application Server Network Deployment edition; otherwise, configuration is completed on a stand-alone server node. Before cataloging is completed on an AIX, Linux, or Solaris platform, a DB2 client instance must be created in the existing DB2 client installation. This is not necessary on a Windows platform. Source the DB2 client instance owner profile in a command shell or start the DB2 Command Line Interface shell before entering the command:

where:

nodename
: Specifies a local alias for the node to be cataloged.

hostname
: Specifies the host name or the IP address of the node where the target database is located. The host name is the name of the node that is known to the Internet Protocol network. The maximum length of the host name is 255 characters.

service_name
: Specifies the service name or the port number of the server database manager instance. The maximum length is 14 characters. This parameter is case-sensitive.

  If a service name is specified, the services file on the client is used to map the service name to a port number. A service name is specified in the database manager configuration file of the server. The services file on the server is used to map this service name to a port number. The port number on the client and the server must match.

You must verify that the TCP/IP node is cataloged correctly. Run the following DB2 commands:

```
db2 attach to nodename user username using password
db2 list applications
db2 detach
```

Where:

nodename
: Specifies the alias of the instance to which you want to attach.

username
: Specifies the authentication identifier.

password
: Specifies the password for the user name.

Pre-installation checklist for all platforms

This section lists the conditions and required actions before installing Common Auditing Service on a Windows, AIX, Linux, or Solaris operating system. If an item is specific to the type of operating system, the limitation is noted in the line item.

Ensure that you check or complete the following tasks before you start the installation of Common Auditing Service:

- Prepare the values for the installation. See Table 2 on page 33.
- Verify that there is enough space to install the Common Auditing Service audit server. Additionally, the system temp directory must have approximately 20 MB to 50 MB for unpacking the audit server installation JAR file during the installation process. After the installation completes, the temporary directories are removed and the file space is reclaimed.
- **On Linux on System z® systems**, Common Auditing Service requires installation of a 32-bit compatibility library package. If the package is not installed on the system, the Common Auditing Service installation will fail because of the missing dependency. Before installing Common Auditing Service, you must install the following 32-bit compatibility library packages:
  - **Red Hat Enterprise Linux 5 on System z:**
    - `compat-libstdc++-295-2.95.3-81.s390.rpm` or higher version
  - **Red Hat Enterprise Linux 6 on System z:**
    - `compat-libstdc++-295-2.95.3-86.el6.s390.rpm` or higher version
  - **SUSE Linux Enterprise Server 9 on System z**:
    - `compat-32bit-2004.7.1-1.2.s390x.rpm` or higher version
  - **SUSE Linux Enterprise Server 10 on System z:**
    - `compat-32bit-2006.1.25-11.2.s390x.rpm` or higher version
  - **SUSE Linux Enterprise Server 11 on System z:**
    - `compat-32bit-2009.1.19-2.1.s390x.rpm` or higher version

  You can obtain the 32-bit compatibility library packages from the Novell Customer Center for SUSE Linux Enterprise Server, or from the Red Hat Network website for Red Hat Enterprise Linux.
- 'The required level of WebSphere Application Server must be installed.
- You can install Common Auditing Service in a WebSphere Application Server stand-alone profile or in a deployment manager profile. It cannot be installed on a managed node.
- The WebSphere Application Server instance that is associated with the profile into which you are installing Common Auditing Service must be running when you start the installation wizard.
- You can run the installation wizard with WebSphere Application Server global security set on or off.
- If WebSphere Application Server global security is set on, you are prompted during installation for the administrator ID and password. Ensure that you install Common Auditing Service using the same administrator ID that is used to install WebSphere Application Server.
- In a WebSphere cluster environment, the Deployment Manager must be running when you start the installation wizard. Set up the other components of a cluster, such as the managed nodes, HTTP server, and plug-ins after the installation of Common Audit Service audit server, but before configuring the Common Audit Service.
- Common Auditing Service has two separately installable components:

- Audit Service (audit server, configuration utilities, and report setup utilities)
  - Audit Configuration Console (configuration console)
- You can install either or both component; however, the following conditions apply:
  - If you are installing Common Auditing Service for the first time *in the same profile*, select the default installation settings, which install both the Audit Service and configuration console.
  - Each installation of the product:
    - Must use a separate installation path
    - Must be installed against a unique WebSphere Application Server profile

    If the Audit Service and Audit Configuration Console are installed at different times but specify the same installation path, they are installed into the same profile.
  - If you reinstall Common Auditing Service, you can install either or both features, depending on your objective.
  - The Audit Configuration Console can configure an Audit Service that is installed in any profile.
  - You can install Common Auditing Service by using a root or non-root administrator ID.
  - Common Auditing Service can be installed multiple times on the same platform to support multiple WebSphere Application Server profiles on different or common WebSphere Application Server installations:

Interactive installation

This section describes how to start and complete an interactive installation of the Common Auditing Service audit server. The interactive installation gives you the option to use GUI panels to enter your setup information for installation or use console mode on the command line.

Starting the installation wizard

This topic describes the command syntax used to start the Common Auditing Service interactive installation wizard in either graphical or console (command line) mode.

The Common Auditing Service installation package consists of the following files:

install_cars_audit_srv.jar
This required file is a single, platform-independent JAR file.

install_cars_audit_srv_*platform*{.exe}
Set of platform-dependent executable binary files (one per platform). The `exe` extension is applicable only on the Windows platform. The corresponding commands for these files are described in this section.

You must use the Java Runtime Environment (JRE) version 1.6 to run Common Auditing Service. Websphere Application Server 8.0 includes JRE 1.6. Ensure that JRE 1.6 is installed, then set the environment variable to the location of the JRE. In the following instructions, the JRE used by WebSphere Application Server is at the correct level (version 1.6).

Note: Linux on System z users: Linux on System z requires the 32-bit Java Runtime Environment (JRE) version 1.6. The 32-bit rpm image is named `ibm-java-s390-sdk-6.0-10.0.s390.rpm` and is included with the Security Access

Manager DVD. For installation instructions, see the "Linux: Installing IBM Java Runtime" section of the *IBM Security Access Manager for Web Installation Guide*.

In the command window, set the environment variable:

- **Systems other than Linux on System z:** `JAVA_HOME=WAS_HOME/java`
- **Linux on System z:** `JAVA_HOME=/opt/ibm/java-s390-60`

where *WAS_HOME* is the installation directory of the WebSphere Application Server.

To run the installation wizard in interactive GUI or console mode, go to the directory that corresponds to the operating system platform that you are using. The following directory names apply:

For operating systems other than Windows platforms:

- `linux_x86`
- `linux_s390`
- `solaris`
- `usr/sys/inst.images` (AIX)

For Windows platforms:
`windows`

In the appropriate directory, specify one of the following commands:

For AIX
install_cars_audit_srv_aix [**-console**] [**-is:javahome** *java_home*]

For Linux on x86-64
install_cars_srv_linux [**-console**] [**-is:javahome** *java_home*]

For Linux on System z
install_cars_srv_linuxs390 [**-console**] **-is:javahome /opt/ibm/java-s390-60**

For Solaris
install_cars_srv_solaris [**-console**] [**-is:javahome** *java_home*]

For Windows
install_cars_srv_win64.exe [**-console**] [**-is:javahome** *java_home*]

For running the Java installation on any platform:**java -cp install_cars_srv.jar run** [**-console**] [**-options-record** *response_file*]

Parameters

`-console`
Run the program in console mode, specifying options on the command line. If you do not specify **`-console`**, the GUI panel installation starts. For a list of the configuration options to enter, see "Audit server installation options" on page 33.

`-options-record` *`response_file`*
Generate a response file by using the options you choose on each panel and write it to the specified file. After you run this interactive installation, you can then use this response file to run a silent installation as it will contain all of the appropriate parameters and values.

`-is:javahome` *`java_home`*
Specify the home directory of the Java virtual machine that the installation launcher uses.

Sample

An example of using the Windows command to use console mode:

```
install_cars_srv_win64.exe -console
```

An example of using the Solaris command to use GUI panels:

```
install_cars_srv_solaris
```

Interactive installation with the GUI panels

This section describes the interactive installation for Common Auditing Service.

Before you begin

See "Interactive installation" on page 30 for the command you enter to begin the audit server installation. Run the command from the media that contains the audit server installation programs.

Procedure

1. Select the language that you want to use for the installation. The default is English.
2. Read the license agreement.
3. Click **Next** if you agree with the license agreement, or press **Cancel** to exit the program. The Welcome dialog is displayed, indicating that the installation will install the Common Auditing Service component.
4. Click **Next** to continue.
5. Select the path into which you want to install Common Auditing Service. A default directory path is provided and is created if necessary. If a Common Auditing Service feature is already installed on this path then only uninstalled features are available for installation.
6. Click **Next** to continue.
7. Select the features that you want to install. By default the Common Auditing Service Server and Configuration Console are selected. If a selected feature is already installed in the specified installation path, that feature is not presented.
8. Click **Next** to continue. The installation wizard searches for WebSphere Application Server installations that can be used as target locations to install Common Auditing Service.
9. Specify the profile directory of the WebSphere Application Server instance into which you are deploying Common Auditing Service. The profile can be either a deployment manager profile or a stand-alone profile. A default directory is provided and the field cannot be blank. See "Audit server installation options" on page 33 for information about the default directory path.
10. Click **Next** to continue. The installation wizard checks to determine whether the specified installation directory already contains the product files. If product files are detected, you are prompted to specify a different target directory.
11. If WebSphere Application Server global security is set, you are prompted in the next window to enter the WebSphere Application Server administrator ID and password. Click **Next** to continue.
12. In the summary window, ensure that all the information that is shown is correct. If you must make a change, click **Previous** to return to a previous window; otherwise, click **Next** to continue.

13. After several minutes, the final window shows that the installation was successful, or indicates that errors occurred and related information is stored in the `serverInstall.log` file.
14. Restart the target WebSphere Application Server Process (Deployment Manager or stand-alone single server).

Audit server installation options

This section describes the Common Auditing Service installation parameters.

Description

The following table summarizes the default options and values that are used for an interactive installation of Common Auditing Service with the ISMP GUI panels.

Table 2. Interactive installation options and values

Configuration option	**Description**

Directory name	Specifies the Common Auditing Service audit server installation directory.
	The default directory for Windows is:
	`c:\Program Files\IBM\Tivoli\CommonAuditService`
	The default directory for AIX, Linux, and Solaris platforms is:
	`/opt/IBM/Tivoli/CommonAuditService`

Feature selection	Specifies the separately installed features of the Common Auditing Service product. The two features are: • **Common Audit Server**. The Common Audit Server feature installs and deploys the following application packages: **CarsConfigMbean.war** This is the management MBean module that is in the selected WebSphere Application Server node. It manages the configuration of Common Auditing Service on that node. **CarsConfigUtil.jar** Comprises the utility class for the configuration MBean. • **Common Audit Server Configuration Console**. The Common Audit Server Configuration Console feature installs and deploys the following application packages: **CARS7.0.war** Comprises the configuration console module. This file is extracted directly into the *`WAS_HOME`* `/AppServer/systemApps/isclite.ear` directory. **CarsConfigUtil.jar** Comprises the utility class for the configuration MBean. This file is extracted into the *`CARS_HOME`* `/server/cons_lib` directory.

Table 2. Interactive installation options and values (continued)

Configuration option	Description
WebSphere Application Server Profile Directory	Specifies the directory path of the WebSphere Application Server profile where you are deploying Common Auditing Service. If the installation wizard detects a WebSphere Application Server profile, the *WAS_HOME* value from this installation is used to create the default profile directory path as: `WAS_HOME/AppServer/profiles/AppSrv01` If a WebSphere Application Server installation is not detected, you must correct the path or create a profile. If a WebSphere Application Server Network Deployment installation is detected, then the default profile is Dmgr01; otherwise, the default profile is AppSrv01.
WebSphere Application Server Administrator user ID	Specify the administrator user ID for WebSphere Application Server. If WebSphere Application Server global security is not enabled, you are not prompted for this value.
WebSphere Application Server Administrator Password	Specify the password for the administrator user ID for WebSphere Application Server. If WebSphere Application Server global security is not enabled, you are not prompted for this value.

Silent installation

This section describes the silent installation of the Common Auditing Service audit server.

Purpose

The silent installation processes the choices in the response file and returns the command prompt when complete. No messages are displayed during the silent installation.

To create a response file that contains all necessary parameters and values, run the interactive installation by using the **`-options-record`** parameter. See "Starting the installation wizard" on page 30 for more information.

Note: After completing the installation, restart the target WebSphere Application Server process (Deployment Manager or stand-alone single server).

Syntax

To run the installation in silent mode, enter one of the following commands from the root directory of the installation media (either the product download directory or the installation DVD):

For AIX

> **install_cars_audit_srv_aix -silent -options** *response_file* [**-is:javahome** *java_home*]

For Linux on x86-64
install_cars_audit_srv_linux -silent -options *response_file* [**-is:javahome** *java_home*]

For Linux on System z
install_cars_audit_srv_linuxs390 -silent -options *response_file* [**-is:javahome** *java_home*]

For Solaris
install_cars_audit_srv_solaris -silent -options *response_file* [**-is:javahome** *java_home*]

For Windows
install_cars_audit_srv_win64.exe -silent -options *response_file* [**-is:javahome** *java_home*]

For running the Java installation on any platform:

java -cp install_cars_audit_srv.jar run -silent -options *response_file*

Parameters

`-options` *`response_file`*
Specifies the name of the response file to use. For example, `serverInstall.rsp`.

`-is:javahome` *`java_home`*
Specifies the home directory of the Java virtual machine that the installation launcher uses.

Sample

The following is an example of using the Windows command with the `serverInstall.rsp` response file:

```
install_cars_audit_srv_win64.exe -silent -options serverInstall.rsp
```

Enabling language support

This section describes how to enable language support.

The Common Auditing Service is translated into the following languages:

- Chinese (traditional)
- French
- German
- Japanese
- Czech
- Hungarian
- Italian
- Korean
- Polish
- Portuguese (Brazil)
- Russian
- Simplified Chinese
- Spanish

The translations for these languages are provided as language packages on the fix pack installation media. The readme file included with the product fix pack describes how to specify the download directory where the language packs are located.

To obtain language support for the Common Auditing Service audit server, you must install the language support package. If you do not install the language support package, the associated product displays all text in English.

If language support is installed and you upgrade the product, you must also install the corresponding language support product, if one exists. If you do not install the language support after upgrading, the associated product might display some fields and messages in English.

Installing language support packages

This section describes how to install language support packages for the Common Auditing and Reporting Service audit server.

Before you begin

The installation media for the Common Auditing and Reporting Service audit server contains the message catalogs for the various languages into which the audit server is translated.

Procedure

1. Log on as root or as an Administrative user.
2. Access the product DVD or extract the files from the archive file that you downloaded from Passport Advantage.
3. Ensure that IBM Java Runtime provided with Security Access Manager is installed for your particular operating system. For instructions, see the *IBM Security Access Manager for Web Installation Guide*.
4. Run either the interactive installation or console mode installation:
   - For interactive installation, run the following command:
     ```
     java -jar carslp.jar
     ```
   - For console mode installation, run the following command:
     ```
     java -jar carslp.jar -console
     ```
5. Click **Next** to begin the installation. The Software License Agreement window is displayed.
6. To accept the license agreement, select the **I accept** check box to accept the terms.
7. Click **Next**. A dialog shows a list of the languages.
8. Select the language packages that you want to install.
9. Click **Next**. A dialog shows the location and features of the languages that you selected.
10. To accept the languages that are selected, click **Next**.
11. After installation completes, click **Finish** to exit the wizard.

What to do next

If it is necessary to uninstall the language support packages, see “Uninstalling language support packages” on page 19 for instructions.

Customizing the XML store data definition language script

This topic describes how to customize the schema of the XML data store (database) to meet your custom data storage needs.

Before you begin

If you want to customize the schema, do so *before* you configure the XML data store (database) for Common Auditing Service by using the configuration console.

To have a customized XML data store, you must create a database during configuration. So, if you are upgrading from an older version of Common Auditing Service and you are using an existing audit database, the steps described in this topic are not applicable.

Before changing the XML store data definition language script, see the performance tuning information in the *IBM Security Access Manager for Web Performance Tuning Guide*. Typically, you must create larger table spaces for use by the XML store tables, and use more than one container for a table space to take advantage of parallel I/O operations by DB2. Consider customizing the buffer pool size to minimize disk I/O, and customizing the database configuration parameters. Creating custom report tables for the purposes of generating custom reports can be done post configuration.

Procedure

1. Identify the changes to the basic data definition language script that you must make to meet your custom storage needs. The changes include identifying the parameters to modify and suitable values for those parameters to set in the script.
2. Save a copy of the original data definition language script that was included in the Common Auditing Service Server installation. The file path of the installed script is `*CARS_HOME*/server/dbscripts/cr_dbobjects.db2`, where *CARS_HOME* is the installation path of Common Auditing Service.
3. Customize the script by editing the `cr_dbobjects.db2` script with a text editor.

   Ensure that there are no SQL syntax errors when you modify the script. Any syntax errors or inappropriate database configuration settings in the customized script can cause a failure in the configuration of the Common Auditing Service Server. If a configuration error occurs after customizing the script, see the configuration logs to determine whether the modifications made to the basic script caused the configuration to fail. You must run the configuration steps again after correcting any errors in the script.
4. Perform the configuration with the GUI configuration console, as described in "Configuring the audit server."

Configuring the audit server

This section describes how to configure the Common Auditing Service audit server.

Pre-configuration checklist for all platforms

This topic lists the required actions before you use the configuration console to set up the audit server.

Procedure

1. After completing Common Auditing Service installation, restart the target WebSphere Application Server process (Deployment Manager or stand-alone single server).
2. Ensure that the target DB2 server instance is running. If the target DB2 server is in the stopped state, start the DB2 instance before you start to configure the Common Auditing Service audit server.
3. Ensure that the user credentials (user name and password) are part of the DB2 instance group, usually `db2grp1`.
4. The DB2 TCP/IP port number is required during the audit server configuration. To determine the port number, do the following steps:
   - **On Windows:** Open the `services` folder in the `C:\Windows\System32\drivers\etc` directory to determine the port number.
   - **On AIX, Linux, or Solaris:**
     a. Enter: `cd /usr/etc`
     b. Enter: `cat services`
     c. Scroll through the list of services until you find the connection port number next to the same service name (`svcename`) parameter that is in the database manager configuration.

   The *DB2 Information Center* describes configuring TCP/IP communications for a DB2 instance.
5. If the DB2 database is remote, ensure that the DB2 node for the remote database is cataloged. Use the **`db2 catalog`** command as shown in “Installing the DB2 Client on AIX, Linux, or Solaris systems” on page 27.
6. If more than a single instance of DB2 is configured on the host, ensure that the system PATH environment variable contains the path to the files of the instance that you will use for Common Auditing Service, for example, the default instance.

   If you upgrade from a previous version of Common Auditing Service, then after cataloging the remote DB2 node, ensure that the audit database that belongs to the previous version of Common Auditing Service that is present on the remote DB2 node is cataloged to the local DB2 client. After cataloging the remote DB2 server node, run the following command to catalog an existing remote audit database into the local DB2 client:

   ```
   db2 catalog database remote_db_name as remote_db_name
   at node cataloged_tcpip_node_name
   ```
7. On a Windows platform, ensure that the **`db2cmd.exe`** command is specified in the system PATH variable.
8. On AIX, if you plan to install DB2 9.7 or later, verify that AIX SP2 is applied by running the following command: **`oslevel -s`**

Interactive configuration using the GUI panels

This topic describes the interactive configuration for Common Auditing Service using the Integrated Solutions Console (ISC) module plug-in to the WebSphere Application Server administrative console.

Before you begin

See “Pre-configuration checklist for all platforms” on page 37 for the items you must consider before you start to configure the use of Common Auditing Service.

Procedure

1. Open a web browser and set the value of the URL to the administrative console port of the WebSphere Application Server deployment manager or stand-alone server that was specified as the target profile during installation of the Audit Configuration Console (default port value is 9060 or 9043 for a secure console).

   Example: http://webspherеserver.ibm.com.:9060/ibm/console
2. Log in as a WebSphere Application Server administrator.
3. Select **Common Audit Service** > **Audit Service Configuration** from the administrative menu. This selection starts the Common Auditing Service configuration wizard. The options presented on each window of the wizard are described in "Common Audit Service configuration options" on page 40. The Welcome dialog is displayed, indicating that Common Auditing Service must be configured before the application can be used.
4. Click **Next** to continue.
5. In the **Common Audit Service Host** window, enter the host name and SOAP port number of the target WebSphere Application Server process (deployment manager or stand-alone single server) where Common Auditing Service is installed.
6. Click **Next** to continue.
7. In the **WebSphere Security** window, if global security is enabled on the target WebSphere Application Server process, select the Global Security check box and enter the WebSphere Application Server administrator name and password.
8. Click **Next** to continue.
9. In the **WebSphere Target Mapping** window, select the configuration target. The list of clusters and independent servers that are available for deployment are displayed in the drop-down list. You must select an entry; if no items are listed, the target WebSphere Application Server is not configured correctly. In this case, you must create a cluster on the target WebSphere Deployment Manager and restart the configuration.
10. In the **Audit Database** window, enter the following information to configure the database that is used by Common Auditing Service. These options are described in "Common Audit Service configuration options" on page 40.
    - Database Instance Owner ID
    - Database Instance Owner Password
    - Database Instance Profile Path

      In a remote database configuration, on AIX, Linux, and Solaris platforms, specify the path of the profile directory of the DB2 Client instance. On a Windows platform, specify the DB2 installation home location.
    - Audit Database Name
    - (Optional) Remote Database Node Name

      Specify this name only when you want to configure the audit database on a remote DB2 server instance.
    - Remote Audit Database
11. Click **Next** to continue.
12. In the **JDBC Connector** window, enter the following information to configure the JDBC driver that is used to connect to the database.
    - Database Server Host Name
    - Database Server TCP Service Port

- JDBC Driver Path

13. Click **Next** to continue.
14. Review the list of options you selected in the configuration Summary window. If the options are correct, select **Finish** to begin the configuration. If one or more options are incorrect, use **Back** to return to a window and make the appropriate changes. When you finish the configuration steps, services that are enabled to run at startup are started.
15. Review the **Common Auditing Service Status** window to determine the outcome of the configuration. If the configuration was unsuccessful, correct the problems and start the configuration again from the Welcome panel.
16. Click **OK** to return to the Welcome panel.

Common Audit Service configuration options

This topic lists and describes the options that are used to configure Common Auditing Service using the configuration console.

Description

The following table summarizes the default values and options used for an interactive configuration of Common Auditing Service using the GUI windows of the configuration console.

Table 3. Interactive configuration values and options

Configuration option	Description
Host	Specifies the name of the WebSphere Application Server host system on which the Common Audit Service configuration component is running. In a WebSphere Application Server cluster environment, specify the name of the host that is running the deployment manager, for example, `idp.example.com`
SOAP Connector Port	Specifies the WebSphere Application Server port number that is configured for SOAP communication. You can view the port values for a WebSphere Application Server Deployment Manager instance by selecting the following links in the administrative console of the Deployment Manager that is hosting the target cluster: **System Administration** -> **Deployment manager**-> **Administration Services**-> **Ports**-> SOAP_CONNECTOR_ADDRESS To view the value of the SOAP connector port for a stand-alone single server, select following links in the administration console of that stand-alone WebSphere Application Server: **Servers**-> **Application servers**-> **server1**-> **Ports**-> SOAP_CONNECTOR_ADDRESS
WebSphere Administrative User Name	Specifies the name of the WebSphere Application Server administrative user that was specified when administrative security was enabled in the target WebSphere Application Server.
WebSphere Administrative User Password	Specifies the password for the WebSphere Application Server administrative user that was specified when administrative security was enabled in the target WebSphere Application Server.

Table 3. Interactive configuration values and options (continued)

Configuration option	Description
Deployment target	Specifies the WebSphere Application Server deployment process where you want to deploy Common Auditing Service.
Database Instance Owner ID	Specifies the administrator user ID for the database instance where the event databases will be created. For example, enter `db2admin`.
Database Instance Owner Password	Specifies the password for the administrator user ID for the database instance.
Database Instance Profile Path	If the target DB2 server is installed locally, this field specifies the path of the db2profile (executable file) for the DB2 instance where the XML data store will be configured. If the target DB2 server is installed remotely, this field specifies the path of the db2profile (executable file) for the DB2 administration client instance that has cataloged the target remote DB2 server instance where the XML data store will be configured.
Audit Database Name	Specifies the name of the database that is used for the XML data store. The default name is `eventxml`.
Remote Database Server Node Name	Use this field only if the target DB2 server is remote. This field specifies the cataloged node name of the remote DB2 server instance that is hosting the XML data store. Specify the same name that is configured in the local DB2 Administration client.
Database Server Host Name	Specifies the DNS host name of the DB2 server that is hosting the XML data store.
Database Server TCP Service Port	Specifies the TCP/IP port on which target DB2 server instance is listening for connection requests. The default port on Windows systems is `50000`; the default port on AIX, Linux, and Solaris systems is `50001`.
JDBC driver path	Specifies the class path for the JDBC driver JAR files (`db2jcc.jar` and `db2jcc_license_cu.jar`) that are used to connect to the Common Auditing Service database (XML data store). Usually these JAR files are present in *`DB2_INSTALL_ROOT`*`/java` on AIX, Linux, and Solaris platforms, and in *`DB2_INSTALL_ROOT`*`\java` on Windows platforms.
Create staging tables and configuration utility	Specifies whether to create the staging tables and configure the staging utility. These tables and the utility are required to enable the generation of reports from Common Auditing Service event records that are stored in the XML data store.

Setting up the JDBC data sources

You must set up JDBC data sources so that you can run the Security Access Manager reports and access the staged audit event data.

Procedure

1. Configure the JDBC data source. The data source is the DB2 database that contains the staged audit data.

2. Copy the data source drivers to the BIRT report engine data source drivers library folder. The path is `<tcr_install_dir>\lib\birt-runtime-2-2-2\ReportEngine\plugins\org.eclipse.birt.report.data.oda.jdbc_2.2.2.r22x_v20071206\drivers`.

   Note: The DB2 Universal JDBC driver is a Type 4 driver that can connect directly to a DB2 server. No DB2 client software is required on the platform where the driver is installed. The Universal JDBC driver class name is: `com.ibm.db2.jcc.DB2Driver`
3. Download the DB2, version 9, JDBC Type 4 driver from https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?lang=en_US&source=swg-dm-db2jdbcdriver. The driver files are `db2jcc.jar` and `db2jcc_license_cu.jar`.

   Note: `<tcr_install_dir>` is the Tivoli Common Reporting installation directory path.

Configuring JDBC resources in a clustered environment

This section provides post-configuration manual steps for configuring JDBC resources in a clustered environment. Complete these steps *after* you configure Common Auditing Service using the configuration console. In Common Auditing Service, the JDBC resources for accessing the XML data store are defined at the cluster scope and are not overridden by the server-scope JDBC resources that are created on managed nodes. To complete the configuration of the JDBC resources in a clustered environment, complete the following manual steps using the WebSphere Application Server Administrative Console of the containing Deployment Manager.

Determining the type of cluster

This topic describes how to determine whether a WebSphere Application Server cluster is homogeneous or heterogeneous.

About this task

Common Auditing Service can be configured against homogeneous and heterogeneous clusters. A cluster is a homogeneous cluster if all of the following conditions apply:

- All of the nodes in the target cluster (including the Deployment Manager node and all managed nodes) are running the same operating system.
- The DB2 Universal JDBC drivers (`db2jcc.jar` and `db2jcc_license_cu.jar`) are installed in the same location on all the nodes in the target cluster (including the Deployment Manager node and all managed nodes), for example, `C:\Program Files\IBM\SQLLIB\java`.

If the above conditions do not apply to a cluster, then, from a JDBC configuration point of view, the cluster is a heterogeneous cluster. To complete the configuration of the JDBC resources in a clustered environment, complete the manual steps described in "Configuring JDBC resources against a heterogeneous cluster" or "Configuring JDBC resources against a homogeneous cluster" on page 43 using the WebSphere Application Server Administrative Console of the container Deployment Manager.

Configuring JDBC resources against a heterogeneous cluster

This topic describes how to configure JDBC resources against a heterogeneous cluster.

Procedure

1. Log in to the WebSphere Application Server Administrative Console of the Deployment Manager that is hosting the target cluster.
2. Click **Environment** > **WebSphere variables** in the left section of the window.
3. Select **All scopes** in the scope settings.
4. Select the **DB2UNIVERSAL_JDBC_DRIVER_PATH** variable that is defined at the target cluster scope.
5. Click **Delete** to remove the variable from the configuration of the Deployment Manager.
6. In the scope settings selection box, select one of the managed nodes that runs at least one cluster member.
7. Click **New** to add the DB2UNIVERSAL_JDBC_DRIVER_PATH variable at the scope of the selected managed node, if one does not exist.
8. Initialize the DB2UNIVERSAL_JDBC_DRIVER_PATH variable to a value that specifies the fully qualified file path (location) of the DB2 Universal Driver JAR files (`db2jcc.jar` and `db2jcc_license_cu.jar`) on the selected managed node.
9. Click **Apply and Save Changes**. If automatic synchronization is not enabled in the container Deployment Manager, ensure that you synchronize the changes to all managed nodes in the cluster.
10. Repeat steps 6 through 9 for each managed node in the cluster.
11. Restart all node agents from the administrative console of the Deployment Manager and then restart the Deployment Manager itself.
12. Restart the target cluster and the container Deployment Manager process for your changes to take effect.

Configuring JDBC resources against a homogeneous cluster

This topic describes how to configure JDBC resources against a homogeneous cluster.

Procedure

1. Restart all node agents from the administrative console of the Deployment Manager, and then restart the Deployment Manager itself.
2. Restart the target cluster from the administrative console of the Deployment Manager.

Configuring the compress property

This topic provides instructions for configuring the compress property for the XML data store.

About this task

By default, the XML data store stores the events in compressed format. To store events in uncompressed format, edit the `ibmcarsserver.properties` file.

Procedure

1. Edit the `*WAS_HOME*/profile/*profilename*/config/ibmcars/ibmcarsserver.properties` file and set the value of the `xmlstore.compress` property to `false` (`xmlstore.compress=false`).
2. Restart WebSphere Application Server.

3. Verify that events are stored in uncompressed format by running the SQL commands:

   ```
   db2 "connect to eventxml user db2inst1 using password"
   db2 "select record_id where is_compressed = 'N' fetch first 1 rows only"
   ```

   If no record is selected, the audit events are stored only in compressed format and the compress property change has not taken effect.

   Note: Storing data in uncompressed format increases disk usage. This should be done only after consultation with IBM support.

Configuring a Web server for use in a clustered environment

This topic describes the steps that are required before you start to use the Common Auditing Service audit server in a WebSphere Application Server clustered environment. You must use a Web server to communicate with applications that are installed into a cluster.

Configuring a Web server that is installed on a cluster node

This topic describes how to configure a Web server that is installed on the same system as one of the cluster nodes.

Procedure

1. Have available a functioning WebSphere Application Server cluster. See the WebSphere Application Server Information Center for instructions on setting up a clustered environment.
2. Ensure that an HTTP server that supports WebSphere Application Server (such as IBM HTTP Server) and WebSphere Application Server plug-in packages are installed and at the correct level.
3. Connect to the WebSphere Application Server deployment manager administrative console. Enter **Servers** > **Server Types** > **Web servers** > **New**.
4. Use the wizard to create a Web server, if one does not exist.
5. In Step 1, complete the fields as follows:

   Select Node
   Select the node that corresponds to the Web server. The Web server should be running on the same system as the selected node.

   Server name
   Enter the Web server name.

   Type Leave the default as **IBM HTTP Server**.
6. Click **Next**.
7. In Step 2, select the IHS template radio button.
8. Click **Next**
9. In Step 3, complete the default fields:

   Port

   Typically you can leave the default value as 80.

   Web server installation location
   Use the default value or specify the filepath location if you are not using the default.

   Plug-in installation location
   Leave the default value as **all**.
10. Click **Next**.

11. In Step 4, confirm your specified settings in **Summary of actions**.
12. Click **Finish**.
13. Save the changes with **Synchronize changes with Nodes** selected.

Configuring a Web server that is installed on a system outside the cluster

This topic describes how to configure a Web server that is installed on a different system than any of the cluster nodes.

About this task

Perform the following steps to configure a Web server that is installed on a system that is outside of the cluster.

See the *WebSphere Application Server Information Center* for detailed instructions on configuring a Web server and an application on separate systems (remote). See "Configuring the Web server plug-in for SSL" on page 164 for information about securing communication with the Web server that use SSL.

Procedure

1. Have available a functioning WebSphere Application Server cluster. See the WebSphere Application Server Information Center for instructions on setting up a clustered environment.
2. Ensure that an HTTP server that supports WebSphere Application Server (such as IBM HTTP Server) and WebSphere Application Server plug-in packages are installed on the remote host and at the correct level.
3. Ensure that the installed Web server is stopped.
4. Use the plug-in installer from the WebSphere Application Server product image or DVD to create a plug-in generation script as follows:
   a. Launch the installation wizard for the plugin using the following command:
      `*WebSphere_Application_Server_install_image_path*/plugin/install`
   b. Clear the roadmap check box, then click **Next**.
   c. Read and accept the license agreement (if you agree with its terms).
   d. Click **Next**.
   e. If the prerequisite check is passed, click **Next**; otherwise correct the prerequisites and restart the installation.
   f. Select the type of Web server you are configuring.
   g. Click **Next**.
   h. Select **Web server machine (remote)**.
   i. Click **Next**.
   j. Accept the default location for the plug-ins installation root directory.
   k. Click **Next**.
   l. Browse for the configuration file of the Web server.
   m. Click **Next**.
   n. Specify a name for the Web server. WebSphere Application Server uses this name to manage the Web server.
   o. Click **Next**.
   p. Accept the default location for the `plugin-cfg.xml` file that is created on the Web server host.
   q. Click **Next**.

r. Enter the host name or IP address of the system where the plug-in configuration script will run. This is the host machine for the deployment manager node.

s. Click **Next**.

t. Examine the summary information to ensure that the specified settings are correct.

u. Click **Next**.

v. Click **Next** on the pre-installation summary window to start installation.

w. If the post-installation window shows that the installation was successful, click **Next**; otherwise, correct any problems and reinstall the plug-in.

x. Close the installation roadmap and click **Finish** to exit the wizard.

   On AIX, Linux, or Solaris systems, the plug-in script is the file *plug-in_ installation_root*/bin/configure*webserver_name*.sh

   On Windows systems, the plug-in script is the file *plug-in_ installation_root*\bin\configure*webserver_name*.bat

   where *plug-in_ installation_root* is the value specified in step j, and *webserver_name* is the value specified in step n.

y. Restart the Web server.

5. To prevent script failure, you might need to compensate for file encoding differences. If the file encoding between the Web server host and the WebSphere Application Server host is different and the platforms are different (AIX, Linux, or Solaris versus Windows), then you must convert the plug-in configuration script as follows:

   a. On an AIX, Linux, or Solaris platform, run the following command:

   ```
   locale
   ```

   On a Windows platform, run the following command:

   ```
   CHCP
   ```

   Run these commands on both the Web server and the WebSphere Application Server systems. The results provide the values of *web_server_machine_encoding* and the *application_server_machine_encoding* variables, respectively.

   b. **Before** moving from an AIX, Linux, or Solaris platform (where the Web server is located), run the following command:

   ```
   iconv -f web_server_machine_encoding -t application_server_machine_encoding
   configurewebserver_name.bat
   ```

   c. **After** moving from a Windows platform (where the Web server is located), run the following command:

   ```
   iconv -f web_server_machine_encoding -t application_server_machine_encoding
   configurewebserver_name.sh
   ```

6. Configure the Web server plug-in to the WebSphere Application Server. Following is an example of how to do this using AIX, Linux, or Solaris:

   a. Copy the Web server configuration file to the WebSphere Application Server installation directory. If you use **ftp**, ensure that you set binary mode first. Following is an example using the ftp copy (**cp**) command:

   ```
   cp /opt/IBM/HTTPServer/Plugins/bin/configurewebserver_name.sh
     /opt/IBM/WebSphere/AppServer/bin/configurewebserver_namewebserver_name.sh
   ```

   Note: If the Web server host and WebSphere Application Server host support different operating systems (AIX, Linux, or Solaris-based and Windows-based), then the script that is copied will be in the crossplatforms

directory. For example:
`opt/IBM/HTTPServer/Plugins/bin/crossPlatformScripts/`
`configure`*`webserver_name`*`.bat`

b. Change to the WebSphere Application Server install directory. For example:

```
cd /opt/IBM/WebSphere/AppServer/bin
```

c. Run the Web server configuration file:

```
./configurewebserver_name.sh
```

d. Connect to the WebSphere Application Server Deployment Manager administrative console. Select **Servers->Server Types-> Web servers->** ***webserver_name*****-> Remote Web server management**.

e. Enter the information in each field:

Port Specifies the HTTP Server administration server port number (default is 8080).

Use SSL
Select this option if the administration port is secured using SSL.

User ID
Specifies the administration user that was created during the installation of the Web server.

Password
Specifies the password of the administration user.

Save the changes with **Synchronize changes with Nodes** selected.

f. Click **OK**.

g. Enter **Servers** > **Server Types** > **Web servers**.

h. Select the check box of the *webserver_name* server. Click **Generate Plug-in** to update the WebSphere Application Server plug-in.

i. Select the check box of the *webserver_name* server. Click **Propagate Plug-in** to update the WebSphere Application Server plug-in.

Enabling the IBM HTTP Server

This topic describes how to enable the IBM HTTP Server in a WebSphere Application Server Network Deployment clustered environment. This process is required in a clustered environment. The HTTP Server acts as a load balancer to forward events to each of the configured managed nodes.

Propagating the plug-in if the IBM HTTP Server is installed on a WebSphere Application Server node host

Follow these steps to propagate the plug-in if the IBM HTTP Server is installed on a WebSphere Application Server node host:

Procedure

1. Connect to the WebSphere Application Server Administrative Console on the deployment manager system:
   a. Expand **Servers->Server Types**.
   b. Click **Web servers**.
   c. Select the check box for the Web server you are using.
   d. Select **Plug-in properties**.
   e. Click **View** to review the `plugin-cfg.xml` document.
   f. Verify that a `ServerCluster` entry exists with the name of the cluster where the CommonAuditService application was deployed.

2. Return to the main window in the console and click **Web servers**.
3. Select the check box for the Web server you are using.
4. Select **Propagate Plug-in**.

Propagating the plug-in if the IBM HTTP Server is installed on a remote host

Follow these steps if the IBM HTTP Server is installed remotely (outside of the cluster).

Procedure

1. Connect to the WebSphere Application Server Administrative Console on the deployment manager system:
   a. Expand **Servers->Server Types**.
   b. Click **Web servers**.
   c. Select the check box for the Web server you are using.
   d. Click Generate **Plug-in**.
2. When the plug-in is generated, note the path and use FTP or another means to move the file indicated for your Web server system, such as:
   ```
   cp path /opt/IBM/WebSphere/Plugins/config/webserver1
   ```
   For example:
   ```
   cp /opt/IBM/WebSphere/AppServer/profiles/Dmgr01/config/cells/
     machine1Cell01/nodes/machine1.tivlab.austin.ibm.com/servers/
     webserver1/plugin-cfg.xml /opt/IBM/WebSphere/Plugins/config/
     webserver1

   cp /opt/IBM/WebSphere/AppServer/profiles/Dmgr01/config/cells/
     machine2Cell01/nodes/machine2.tivlab.austin.ibm.com/servers/
     webserver1/plugin-cfg.xml /opt/IBM/WebSphere/Plugins/config/
     webserver1
   ```
3. Stop and restart the IBM HTTP Server and the HTTP administrative server.
4. Stop and restart the cluster.

Completing the Common Auditing Service application to Web server mapping

This topic describes how to finish the mapping of the Common Auditing Service application to the Web server.

Mapping the Common Auditing Service server to the virtual host:

This topic describes how to map the Common Auditing Service to the virtual host.

Procedure

1. In the WebSphere Application Server Administrative Console, click **Applications->Application Types-> WebSphere Enterprise Applications-> CommonAuditService-> Virtual hosts**.
2. Select the check box for the **Common Audit Service** Web module.
3. Ensure that default_host is the designated virtual host for the selected module.
4. Click **OK**.
5. Save your changes. If Common Auditing Service is operating in a WebSphere Application Server Network Deployment environment, select **Synchronize changes with Nodes** before saving the changes.

Mapping the Common Auditing Service server to the target servers:

This topic describes how to map the Common Auditing Service server to the target servers.

Procedure

1. In the WebSphere Application Server Administrative Console, click **Applications->Application Types-> WebSphere Enterprise Applications-> CommonAuditService-> Manage Modules**.
2. Verify that the `IBMCARSxmlstoreds-ejb` and `Common_Audit_Service` modules are mapped to the cluster (or server) that was selected during configuration.
3. In the Clusters and Servers window, press and hold the **Ctrl** key while selecting the target cluster (or server) and the target Web server.
4. Select the check box for module `Common_Audit_Service`.
5. Click **Apply**.
6. Ensure that the correct cluster (or server) and Web server were updated against the Web Module.
7. Click **OK**.
8. Save your changes. If Common Auditing Service is operating in a WebSphere Application Server Network Deployment environment, select **Synchronize changes with Nodes** before saving the changes.

Verifying your configuration settings for Common Auditing Service

This topic describes several procedures for verifying the correct configuration of the Common Auditing Service audit server. These procedures use the WebSphere Application Server Administrative Console to inspect the deployment parameters of various application components of Common Auditing Service, and use the IBM DB2 command-line interface to review the database instance that is present after a correct deployment. Additional active steps are also provided that use a Web browser.

Verifying the configuration settings for the Common Auditing Service application

Use the following steps in the WebSphere Application Server Administrative Console to verify that the Common Auditing Service application is configured correctly.

Procedure

- Determine that all modules are present:
  1. Click **Applications->Application Types-> WebSphere Enterprise Applications-> CommonAuditService-> Manage Modules**.
  2. Ensure that the `IBMCARSxmlstoreds-ejb` module exists, is of type EJB Module, and is deployed in the correctly named target (whether stand-alone or cluster).
  3. Ensure that the Common Audit Service module exists, is of type Web Module, and is deployed in the correctly named target (whether stand-alone or cluster).
- Ensure that the Web module is configured:
  1. Click **Applications->Application Types-> WebSphere Enterprise Applications-> CommonAuditService-> Session Management**.
  2. In the Configuration window, verify that the following general properties have values:

Enable cookies
Check this option.

Allow overflow
Check this option.

Maximum in-memory session count
Set to 1000 sessions.

Set timeout
Selected and set for 30 minutes.

3. Click **Applications->Application Types-> WebSphere Enterprise Applications-> CommonAuditService-> Context Root For Web Modules**.
4. Verify the following settings in the table:
   - `Web Module=Common Audit Service`
   - `URI=cars-webservice.war,WEB-INF/web.xml`
   - `ContextRoot=CommonAuditService`
5. Click **Applications->Application Types-> WebSphere Enterprise Applications-> CommonAuditService-> JSP and JSF options**.
6. Verify the following settings in the table:
   - `Web Module=Common Audit Service`
   - `URI=cars-webservice.war,WEB-INF/web.xml`
   - `JSP enabled class reloading=enabled`
   - `JSP reload interval in seconds=10`
7. Click **Applications->Application Types-> WebSphere Enterprise Applications-> CommonAuditService-> Virtual hosts**.
8. Verify the following settings in the table:
   - `Web Module=Common Audit Service`
   - `Virtual host=default_host`

- Ensure that the EJB module is configured:
  1. Click **Applications->Application Types-> WebSphere Enterprise Applications-> CommonAuditService-> EJB JNDI names**.
  2. Verify the following settings in the table:
     - `EJB module=IBMCARSxmlstoreds-ejb`
     - `EJB Bean=XmlStore`
     - `URI=IBMCARSxmlstoreds-ejb.jar,META-INF/ejb-jar.xml,`
     - `Target Resource JNDI Name=ejb/com/ibm/cars/xmlstore/xmlstoreds/XmlStoreLocalHome`
  3. This step and the next step apply only to cluster configurations: Click **Applications->Application Types-> WebSphere Enterprise Applications-> CommonAuditService-> Stateful session bean failover settings**
  4. In the Configuration window, verify that the following general properties have values:

     Enable stateful session bean failover using memory to memory replication
     Check this option.

     Use replication settings from EJB container
     Check this option.
- Ensure that the EJB references are configured:
  1. Click **Applications->Application Types-> WebSphere Enterprise Applications-> CommonAuditService-> EJB references**.

2. Verify the following settings in the table:
   - `Module=Common Audit Service`
   - `URI=cars-webservice.war,WEB-INF/web.xml`
   - `Resource Reference=ejb/XmlStore`
   - `Class=com.ibm.cars.xmlstore.xmlstoreds.XmlStoreLocal`
   - `Target Resource JNDI Name=ejb/com/ibm/cars/xmlstore/xmlstoreds/XmlStoreLocalHome`

Verifying the configuration settings for the Common Auditing Service data source

Use the following steps in the WebSphere Application Server Administrative Console to verify that the data source used by the EJB component is configured correctly and can connect to the audit database.

Procedure

- Verify the JDBC provider:
  1. Click **Resources-> JDBC-> JDBC Providers**.
  2. Ensure that the scope setting is set to **All scopes**.
  3. Verify the following settings in the table:
     - `Name=Event_DB2Xml_JDBC_Provider`
     - `Scope=`*`expected_cluster_or_server_scope`*
     - `Description=DB2 Universal JDBC Driver Provider (XA) for the Common Auditing Service`
  4. Click **Resources-> JDBC-> JDBC Providers-> Event_DB2Xml_JDBC_Provider**
  5. In the Configuration window, verify the following setting:

     `Implementation class name=com.ibm.db2.jcc.DB2XADataSource`
- Verify the data source:
  1. Click **Resources-> JDBC-> Data sources**.
  2. Ensure that the scope is set to the cluster or server where Common Auditing Service is deployed.
  3. Verify the following settings in the table:
     - `Name=eventxml`
     - `JNDI name=jdbc/eventxml`
     - `Scope=`*`scope_selected_in_step_2`*
     - `Provider=Event_DB2Xml_JDBC_Provider`
     - `Description=JDBC Datasource for EVENTXML database`
  4. Check the check box of the entry verified in step 3.
  5. Click **Test connection**. The following message displays for a cluster configuration:

     `The test connection for data source eventxml on server nodeagent at node` *`first_node_in_cluster`* `was successful.`

     The following message displays for a stand-alone server configuration:

     `The test connection for data source eventxml on server` *`server_name`* `at node` *`node_name`* `was successful.`

Verifying the configuration settings for the Common Auditing Service data store

Use the following command-line procedures to verify that the correct data store is set up for use by Common Auditing Service.

Procedure

- Verify the audit database schema:
  - On AIX, Linux, or Solaris systems, from the command line enter:

    ```
    . ~db2_instance_name/.profile
    db2 connect to audit_database_name user db2_admin_name using db2_admin_password
    db2 "select * from cei_t_properties where property_name like 'Schema%'"
    ```

  - On Windows systems, from the command line enter:

    ```
    db2_profile.bat
    ```

  The following three DB2 table entries should be displayed:

  ```
  SchemaMajorVersion     6
  SchemaMinorVersion     0
  SchemaPtfLevel         0
  ```

- Verify that the common base event type is used:
  - On AIX, Linux, or Solaris systems, from the command line enter:

    ```
    . ~db2_instance_name/.profile
    db2 connect to audit_database_name user db2_admin_name using db2_admin_password
    db2 "select * from cei_t_properties where property_name like 'Cbe%'"
    ```

  - On Windows systems, from the command line enter:

    ```
    db2_profile.bat
    ```

  The following three DB2 table entries should be displayed:

  ```
  CbeMajorVersion       1
  CbeMinorVersion       0
  CbePtfLevel           1
  ```

Verifying the configuration settings for the Common Auditing Service web service component

Use the following procedures in the WebSphere Application Server Administrative Console to verify that the web service component named Common Audit Service is running correctly:

Procedure

- Determine that the Web application is running:
  1. Enter **Applications->Application Types-> WebSphere Enterprise Applications**.
  2. Verify that the application with name CommonAuditService is present and that the Application Status is running (indicated by the green right arrow).
- Determine the web service port:

  Enter **Servers->Server Types-> WebSphere application servers->** *server_name*-> **Ports**.

  The default host port is named WC_defaulthost with an installed default value of 9080. The secure host port is WC_defaulthost_secure with installed default value of 9443. Values are assigned automatically during profile creation and can differ from these default values to avoid conflict. Multiple servers in a cluster can each have different port allocations.

  - If no SSL configuration is allocated, then point the browser at the web service to test the web service port:

    URL: http://*host_name.WC_defaulthost_port_number*/CommonAuditService/services/Emitter

The browser window should display:

```
{urn:ibm:cars:10}Emitter

Hi there, this is a Web service!
```

- If an SSL configuration is allocated to the Common Auditing Service server at the cluster scope, or at the node scope, then do the following steps to test the web service port:

  1. Obtain a security certificate:
     a. Enter **Security-> SSL certificate and key management-> SSL configurations->** *ssl_configuration_name*.
     b. Determine the keystore name and default server certificate alias values that you want to use.
     c. Enter **Security-> SSL certificate and key management-> SSL configurations->** *ssl_configuration_name***-> Key stores and certificates->** *keystore_name***-> Personal certificates**.
     d. Select the entry with the alias that matches the default server certificate alias from the Personal certificates table.
     e. Click **Extract certificate** to display the general properties of the certificate.
     f. Enter a directory path and file name for the certificate file name.
     g. Click **OK** to extract the certificate from the keystore. Note that this step extracts the certificate only, it does not extract the private key that belongs to the certificate.
  2. Import the server certificate to a Web browser. The steps for this procedure depend on the browser. The following steps are for the Firefox browser:
     a. If the browser host is different from the current host, copy the certificate file obtained in step 1 to the browser host.
     b. Select **Edit-> Preferences-> Advanced-> Security**.
     c. Click **View Certificates**.
     d. Select the **Web Sites** tab to view the list of site certificates.
     e. Click **Import**.
     f. Navigate to the certificate file obtained in Step a, or from Step 1 under **Obtain a security certificate**.
     g. Click **Open**. The certificate should display in the list of certificates that is displayed.
     h. Click **OK**.
     i. Click **Close** to exit.
  3. Point the browser at the web service to test the web service port:

     URL: https://*host_name.WC_defaulthost_secure_port_number*/CommonAuditService/services/Emitter

     The browser window should display:

     ```
     {urn:ibm:cars:10}Emitter

     Hi there, this is a Web service!
     ```

Deploying the Java stored procedure for an audit details report

This section describes how to deploy the Java stored procedure, which is required for an audit details report. An audit event details report is used to view all attributes of a security event.

Before you begin

An example of an audit details report is the General Audit Event Details report that is included with IBM Security Access Manager report package.

Customized reports can also access the audit details Java stored procedure. See Chapter 13, "Creating custom reports," on page 123 for information about creating custom reports.

Before running the custom audit details report, the Java stored procedure must be deployed on the Common Auditing Service audit server. In a WebSphere Application Server stand-alone server environment, the audit server is installed on the system where WebSphere Application Server is installed. In a cluster environment, the audit server is installed on a WebSphere Application Server Network Deployment edition deployment managed node.

During the installation of the audit server, the **ibmcarsddinst.sh** script (AIX, Linux, and Solaris) and **ibmcarsddinst.bat** script (Windows) are installed in the `*CARS_HOME*/server/bin/` directory to make deployment of the Java stored procedure easier. The installation also installs the `*CARS_HOME*/server/lib/ibmcarsdd.jar` file that contains the Java stored procedure. (*CARS_HOME* is the installation directory of the Common Auditing Service.)

Procedure

1. **Linux only:** See "Setting up to run the Java stored procedures on Linux."
2. **All platforms:** See "Setting the jdk_path parameter" on page 55.
3. **All platforms:** Run the **ibmcarsddinst** script. See "Running ibmcarsddinst to deploy the Java stored procedure" on page 55.
4. **All platforms:** Verify that the deployment of the Java stored procedure was successful. See "Verifying the deployment of the IBMCARS_EVENT_DETAIL Java stored procedure" on page 57.

Setting up to run the Java stored procedures on Linux

This section describes what setup is required to run the Java stored procedure on Linux.

About this task

To run the Java stored procedures or user-defined functions, the Linux runtime linker must be able to access certain Java shared libraries. Also, DB2 must be able to load both these libraries and the Java virtual machine. Because the program that does this loading runs with setuid privileges, it only looks for the dependent libraries in the `/usr/lib64` directory.

Note: You must make the symbolic links on the machine that runs the DB2 server.

Procedure

Run the following commands to create symbolic links in the /usr/lib64 directory:

- **On Linux x86-64:**

  ```
  cd /usr/lib64
  ln -s WAS_HOME/AppServer/java/jre/lib/amd64/libjava.so .
  ln -s WAS_HOME/AppServer/java/jre/lib/amd64/classic/libjvm.so .
  ln -s WAS_HOME/AppServer/java/jre/lib/amd64/libjsig.so .
  ln -s WAS_HOME/AppServer/java/jre/lib/amd64/libzip.so .
  ```

- **On Linux on System z:**

  ```
  cd /usr/lib64
  ln -s WAS_HOME/AppServer/java/jre/lib/s390x/libjava.so .
  ln -s WAS_HOME/AppServer/java/jre/lib/s390x/classic/libjvm.so .
  ln -s WAS_HOME/AppServer/java/jre/lib/s390x/libjsig.so .
  ln -s WAS_HOME/AppServer/java/jre/lib/s390x/libzip.so .
  ```

where *WAS_HOME* is the installation directory for the WebSphere Application Server.

Note: In a network deployment environment where WebSphere Application Server is not installed on the DB2 server host, specify the base directory of Java 1.6 or later, instead of specifying *WAS_HOME*/AppServer.

Setting the jdk_path parameter

This section describes how to set the Software Development Kit (SDK) for Java installation path configuration parameter, jdk_path, on all platforms. The SDK for Java is used for running Java stored procedures and user-defined functions.

Before you begin

The DB2 database manager parameter specifies the directory under which the SDK for Java is installed. The CLASSPATH and other environment variables used by the Java interpreter are computed from the value of this parameter. Because there is no default value for this parameter, you should specify a value when you install the SDK for Java.

Procedure

1. Verify the existing jdk_path by using the following command in a DB2 command-line window:

   ```
   db2 get dbm cfg
   ```

   Look for jdk_path in the configuration file to see the current jdk_path setting.
2. Set the jdk_path parameter by using the following **db2** configuration command:

   ```
   db2 update dbm cfg using JDK_PATH java_installation_path
   ```

   where *java_installation_path* is the location of Java. See the *IBM DB2 Command Reference* for more information.

Running ibmcarsddinst to deploy the Java stored procedure

Use the following information to run the operating system-specific **ibmcarsddinst** script to deploy the Java stored procedures on the server machine. The Common Auditing Service configuration console does not include an option to complete this task.

Syntax

For AIX, Linux, or Solaris
ibmcarsddinst.sh -u *user* **-p** *password* [**-a** *database_alias*] [**-d** *directory*]

For Windows
ibmcarsddinst.bat -u *user* **-p** *password* [**-a** *database_alias*] [**-d** *directory*]

Parameters

-u *user*
Specifies the database user name.

-p *password*
Specifies the password associated with the database user name.

[**-a** *database_alias*]
Specifies the database alias. The default value is EVENTXML.

[**-d** *directory*]
Specifies the location of the JAR file that contains the Java stored procedure. You must specify the full path and not the relative path. The default value is the current directory.

Sample

Following is an example of how to run the file on a Windows system:

```
ibmcarsddinst.bat -u joe -p secret1pw -d CARS_HOME/server/lib
```

where *CARS_HOME* is the installation directory of the Common Audit Service.

Notes

On a Windows system, run the **db2cmd** command to start a DB2 shell. In this shell, run the **ibmcarsddinst.bat** script.

In addition to printing informational messages, the deployments set the ERRORLEVEL variable to 0 for success and non-zero values for failures or warnings.

On AIX, Linux, or Solaris systems, running the **ibmcarsddinst.sh** script returns a status code of 0 for success and non-zero for failures and warnings.

Verifying the deployment of the IBMCARS_DD_REPORT Java stored procedure

This section describes how to verify that the IBMCARS_DD_REPORT Java stored procedure is deployed correctly.

Procedure

1. Enter the following SQL commands from the command line:

   ```
   db2 connect to eventxml user user using password
   db2 "call IBMCARS_DD_REPORT(record_id, format)"
   ```

   where *record_id* is the record identifier of the event whose details are required. If the specified *record_id* exists in the event store, the *record_id* and the associated event details are returned in XML format. If the Java stored procedure is not deployed correctly, then the following error is returned:

   ```
   SQL0440N.  No authorized routine named "IBMCARS_DD_REPORT" of type
   "PROCEDURE" having compatible arguments was found.
   ```
2. End the DB2 session with the following command:

```
db2 terminate
```

Verifying the deployment of the IBMCARS_EVENT_DETAIL Java stored procedure

This section describes how to verify that the IBMCARS_EVENT_DETAIL Java stored procedure is deployed correctly.

Procedure

1. Enter the following SQL commands from the command line:

   ```
   db2 connect to eventxml user user using password
   db2 "call IBMCARS_EVENT_DETAIL(record_id, format)"
   ```

   record_id
   Specifies the record identifier of the event whose details are required.

   format
   Specifies the type of output format:

   - Specify map to display the security event details as name-value pairs.
   - Specify xml to set off special formatting of the data. This is the equivalent of calling the IBMCARS_DD_REPORT Java stored procedure.

   If the specified *record_id* exists in the event store, the *record_id* and the associated event details are returned. If the Java stored procedure is not deployed correctly, the following error is returned:

   ```
   SQL0440N.  No authorized routine named "IBMCARS_EVENT_DETAIL" of type
   "PROCEDURE" having compatible arguments was found.
   ```

2. End the DB2 session with the following command:

   ```
   db2 terminate
   ```

Upgrading the Common Audit Service audit server

This topic describes the procedures for installing and configuring the Common Auditing Service audit server to use an existing earlier version of the audit server database.

Considerations for upgrading the Common Audit Service audit server

Consider the following points when preparing to upgrade from Common Audit Service versions 6.0, 6.0.1, or 6.1.x to version 7.0:

- What is new in Common Audit Service version 7.0:
  - WebSphere Application Server 7.0 and 8.0 support
  - DB2 9.7 FP4 and later support

  Reports and functionality for Common Audit Service remain unchanged from version 6.1. If you do not require WebSphere Application Server 8 or DB2 9.7 FP4 and do not want to upgrade the Common Audit Server version 6.1 server, you can choose to keep your Common Audit Server at version 6.1 and have full compatibility with Common Audit Server clients at the 7.0 level.
- Common Audit Service Version 7.0 requires DB2 9.7 FP4 or later. The DB2 server is *not* upgraded during an upgrade of Common Audit Service. Before configuring the Common Audit Service, you *must* upgrade DB2 to 9.7 FP4 or later.

  If you do not have DB2 version 9.7 FP4 or later before configuring the Common Audit Service, the following message displays in the installation panel:

```
CBAIN0130E Prerequisite detection has found an installation of
either IBM DB2 Server or Client but it is not a correct version.
The versions allowed are  Version 9.7 and higher. You must install
an allowable version of the IBM DB2  product either now or before
attempting to use the selected product feature.
```

The DB2 image can be found on the product DVD in the following directory: `tdsV6.3/db2`.

For more information about installation or upgrade of DB2 9.7 FP4 or later, see the DB2 documentation.

- The XML data store (XMLSTORE database) from an earlier version of the Common Audit Service audit server is the *only* product component that upgrades to version 7.0. The database upgrades if you specify to configure Common Audit Service Version 7.0 to use an existing XMLSTORE database that was used by earlier versions of Common Audit Service.
- Installations of Common Audit Service before version 6.1 allowed only the `root` user to install the product on AIX, Linux, or Solaris-based platforms, and allowed only the `Administrator` user to install the product on Windows platforms. Upgrades from Version 6.0.x *must* be run by a root user on AIX, Linux, or Solaris-based platforms and by an Administrator user on Windows platforms.

Upgrade goals

The main goals of the upgrade procedure are as follows:

- Preserve the data in the existing audit database by retaining the original DB2 database during the upgrade.
- Ensure the integrity of the audit data that is present in the original audit database by backing up the audit data on the file system of the database server.
- Provide a common procedure for upgrading earlier versions of Common Audit Service to version 7.0.
- Prevent accidental removal of the audit database if you choose to uninstall earlier versions of Common Audit Service.
- Allow existing Common Audit Service client applications to switch to the new server in a phased manner by allowing both the old and new Common Audit Service audit server to write to the database.

The previous goals are achieved by using the following features in Common Audit Service Version 7.0:

- Common Audit Service Version 7.0 allows the audit server to be installed on the same host (physical machine) that has an earlier installation. The new version and the earlier versions of the audit server are installed in different locations (file paths).
- Configure the audit server to use an existing database by using the configuration utility of Common Audit Service Version 7.0.
- Common Audit Service Version 7.0 ships duplicate copies of the **`ConfigureRm.bat`** script on Windows platforms, and the **`ConfigureRm.sh`** script on AIX, Linux, or Solaris platforms.

  When these scripts are replaced with copies of corresponding scripts from an earlier installation of the product at `*was60_profile_path*\event\dbscripts\db2xml` on Windows platforms, and at `*was60_profile_path*/event/dbscripts/db2xml` on AIX, Linux, or Solaris platforms, a previous installation of the product can be uninstalled without dropping the associated XMLSTORE database, preventing an accidental removal of the XML data store.

Preparing to upgrade the Common Auditing Service audit server

Perform the following tasks *before* you upgrade to version 7.0 of Common Auditing Service.

Procedure

- Back up the existing Common Auditing Service XML data store (XMLSTORE database).

  The upgrade process alters the XMLSTORE database in a transaction mode; so, the changes made to the existing XMLSTORE database are not committed in the event of an upgrade failure.

  Also, the upgrade process does not drop the existing XMLSTORE database in the event of an upgrade failure. Nevertheless, you should back up the existing XMLSTORE database before operating on it to prevent the unexpected loss or corruption of the event data contained in the existing XMLSTORE database.

  Perform these steps to back up the existing XMLSTORE database:

  1. Disconnect all applications that are connected to DB2 and restart the database server instance by running following commands from the operating system command window:

     ```
     db2stop force
     db2start
     ```

  2. Create a backup directory with the following example command:

     AIX, Linux, or Solaris systems:

     ```
     mkdir /export/eventxml_bkup
     ```

     Windows systems:

     ```
     md \export\eventxml_bkup
     ```

     Note: On Windows platforms only, do not create a backup directory that has one or more blank spaces in the file path. The DB2 database backup command fails to back up databases to locations that have blank space characters in the file path.

  3. Modify the permissions on the backup directory to ensure that the DB2 instance owner user can write to it:

     AIX, Linux, or Solaris systems:

     ```
     chmod a+w /export/eventxml_bkup
     ```

     Windows systems:

     Change permissions on `/export/eventxml_bkup` by using right-click -> **Properties**.

  4. Perform a full backup of the database to the newly created backup directory with this example command:

     AIX, Linux, or Solaris systems:

     ```
     db2 backup database eventxml user db2inst1 using password to
     /export/eventxml_bkup with 2 buffers buffer 512 parallelism 2
     ```

     Windows systems:

     ```
     db2 backup database eventxml user db2inst1 using password to
     \export\eventxml_bkup with 2 buffers buffer 512 parallelism 2
     ```

     Where *db2inst1* is the database instance owner ID, and *password* is the database instance owner password.

- Ensure that all prerequisite software is installed on the system before installing the product. The required software is listed in the *Release Notes*® for the product that is using Common Auditing Service.
- Ensure that all conditions are met that are described in "Pre-installation checklist for all platforms" on page 29.
- If you upgrade from Common Auditing Service Version 6.0.x or version 6.1.x and deploy in a WebSphere Application Server Version 8 cluster, set up the IBM HTTP Server Version 8 (using the WebSphere Application Server Version 8.0) to use a different port than port 80. Specifying a different port allows both the existing cluster and the new cluster to be used simultaneously.

Upgrading the Common Audit Service audit server

This topic describes the procedures for installing and configuring the Common Auditing Service audit server to use an existing earlier version of the audit server database.

Installing Common Auditing Service Version 7.0 when upgrading to use an existing database

Follow the procedure described in "Interactive installation" on page 30 to install Common Auditing Service interactively, or follow the procedure described in "Silent installation" on page 34 to install Common Auditing Service silently in console mode by using a response file.

Note: Ensure that you install Common Auditing Service at a location that is different than the location where the existing earlier version of Common Auditing Service is installed.

After completing the installation, restart the target WebSphere Application Server process (Deployment Manager or stand-alone single server).

Configuring Common Auditing Service Version 7.0 to use an existing audit database

Follow this procedure to configure Common Auditing Service Version 7.0 to use an existing XML data store (XMLSTORE database) that is being used by an older version of Common Auditing Service.

Procedure

1. Log in to the administrative console of the WebSphere Application Server in which you installed the **Common Audit Service Server** feature of Common Auditing Service Version 7.0.
2. Select **Common Audit Service-> Audit Service Configuration** from the left pane of the window to start the configuration wizard for Common Auditing Service.
3. Click **Next** in the Welcome window to continue. The Welcome dialog is displayed, indicating that Common Auditing Service must be configured before the application can be used.
4. Click **Next** to continue.
5. In the **Common Auditing Service Host** window, enter the host name and SOAP port number of the target WebSphere Application Server process (Deployment Manager or stand-alone single server) where Common Auditing Service exists.
6. Click **Next** to continue.

7. If administrative security is set ON in the target WebSphere Application Server process, enter the WebSphere administrator user name and password in the **WebSphere Security** window.
8. In the **WebSphere Target Mapping** window, select the configuration target.
9. In the **Audit Database configuration** window, enter the following information:

   Database Instance Owner ID

   Specify the name of the user who is the instance owner of the DB2 instance where the existing target XML data store (XMLSTORE database) is located.

   Database Instance Owner Password

   Specify the password of the user who is the instance owner of the DB2 instance on which the target lower-version existing XMLSTORE database is located.

   Database Instance Profile Path

   If the target DB2 server is locally installed, specify the file path of the db2profile executable file that is associated with the DB2 instance on which the existing lower-version XMLSTORE database is located.

   If the target DB2 server is remotely installed, specify the file path of the db2profile executable file that is associated with the DB2 Administration client instance that cataloged the target remote DB2 server instance on which the target XMLSTORE database is located.

   On Windows platforms, you might not need to create a DB2 client instance to catalog the remote DB2 server instance. If so, specify the installation root location of the underlying DB2 client in this field.

   Audit Database Name

   Specify the name of the existing lower-versioned target XMLSTORE database to be upgraded to version 7.0.

   Remote Database Node Name

   Specify a value only if the target XMLSTORE database is on a remote DB2 server. If the XMLSTORE database is local, leave this field blank. This field specifies the cataloged node name of the remote DB2 server instance that is hosting the target XMLSTORE database, as it appears in the local DB2 Administration client.
10. In the **Audit Database JDBC Connector** window, enter the following information:

    Database Server Host Name

    Specify the DNS host name of the DB2 server that is hosting the target lower-versioned XMLSTORE database.

    Database Server TCP Service Port

    Specify the TCP/IP port on which target DB2 server instance is listening for connection requests.

    JDBC Driver path

    Specify the path to the location of the system that contains DB2 type-4 JDBC driver JAR files (`db2jcc.jar` and `db2jcc_license_cu.jar`). Usually these library JAR files are present at *DB2_INSTALL_ROOT*/java on AIX, Linux, and Solaris platforms, and at *DB2_INSTALL_ROOT*\java on Windows systems.

11. Click **Next** in the **Summary** window to start the configuration of Common Auditing Service Version 7.0 to use the existing version of the XMLSTORE database.

    After the configuration wizard completes, ensure that the status is SUCCESS for all server components that are displayed in the status window.

    A status of SUCCESS for all server components indicates that you successfully configured Common Auditing Service Version 7.0 to use the existing lower version of the XMLSTORE database. Additionally, the target lower-versioned XMLSTORE database was upgraded to version 7.0.

What to do next

Immediately after finishing the above procedure, follow the post-upgrade steps described in “Post-upgrade steps: remove the old script, configure the clients to use the new port, uninstall the old version of the audit server” to cause clients that use the older version of Common Auditing Service to start sending events to Common Auditing Service Version 7.0, and to ensure that the upgraded existing XMLSTORE database is not accidentally dropped during uninstallation of the older version of Common Auditing Service.

Post-upgrade steps: remove the old script, configure the clients to use the new port, uninstall the old version of the audit server

After successfully completing the procedures that enable the Common Auditing Service Version 7.0 audit server to use the existing, older-version XMLSTORE database, follow the steps in this procedure.

About this task

These steps prevent an unintended loss of data, enable your application clients to begin sending events over the new audit server port, and uninstall the old version of Common Auditing Service.

Procedure

1. Replace the database removal script of the earlier lower version of Common Auditing Service with the identically named script that is shipped with Common Auditing Service Version 7.0.

   Note: If you are in a WebSphere Application Server Network Deployment environment during the upgrade of Common Auditing Service, complete this step only on the Deployment Manager.

   AIX, Linux, or Solaris systems

   Replace the old **`dbConfigureRm.sh`** script that is in the following directory:

   `OLD_CARS_HOME``/server/dbscripts`

   with the new **`dbConfigureRm.sh`** script that is in the following directory:

   `CARS_HOME``/server/etc/upgrade`

   Windows systems

   Replace the old **`dbConfigureRm.bat`** script that is in the following directory:

   `OLD_CARS_HOME``\server\dbscripts`

   with the new **`dbConfigureRm.bat`** script that is in the following directory:

```
CARS_HOME\server\etc\upgrade
```

2. Perform the following steps *before* you start the procedure to uninstall the earlier lower-level version of Common Auditing Service.
   - If the Common Auditing Service audit server was upgraded in a stand-alone server environment, complete the following steps:
     a. Log on to the WebSphere Application Server Administrative Console.
     b. Select **Servers** > **Server Types** > **WebSphere application servers** > **server1** > **ports**.
     c. Identify the application port:
        – **Without SSL communication enabled:** See the **`WC_defaulthost`** entry in the table.
        – **With SSL communication enabled:** See the **`WC_defaulthost_secure`** entry in the table.
     d. Stop and restart all versions (old and new) of the Common Auditing Service audit server.
     e. Reconfigure one or more Common Auditing Service client applications to send audit events to the new application port. The old and new versions of the Common Auditing Service audit server continue writing to the database until all clients are configured to use only the new application port and the new server.
   - If the Common Auditing Service audit server was upgraded in a clustered environment, the clients typically communicate with an HTTP server; therefore, changing the configuration on the client application is not necessary.
     a. Stop all versions of Common Auditing Service audit servers. It is important that you stop the old and new servers.
     b. Stop both the old and new clusters of the WebSphere Application Servers that are being used by the old and new versions of the Common Auditing Service audit server. This action automatically stops the audit servers on both of the clusters.
     c. Stop the IBM HTTP Server Version 8 that is configured for use with the WebSphere Application Server 8.0 cluster, and stop the IBM HTTP Server Version 8.0 that is configured for use with the WebSphere Application Server 8.0 cluster.
     d. Reconfigure the IBM HTTP Server Version 6.x that is configured to be used as a load-balancer for the WebSphere Application Server 6.x cluster to listen on port 80, then restart the same HTTP server. The Common Auditing Service Version 8.0 audit server is now the audit server that stores events in the audit database.
3. Uninstall the *older* version of Common Auditing Service using the uninstallation instructions provided in the *Tivoli Access Manager for e-business 6.x.x Auditing Guide*.

   Attention: Do *not* select **Remove Audit Database** while unconfiguring the Common Auditing Service. Choosing this option removes the database contents.

Results

After you successfully complete this procedure, the upgrade to Common Auditing Service Version 7.0 is finished.

Chapter 6. Running the server utilities

This chapter provides information about the staging utility and the XML data store utilities.

- The staging utility incrementally updates and maintains the staging tables.
- The XML data store utilities help you manage the XML data store.

This chapter also includes details about the `ibmcars.properties` configuration file, which contains the options that you can use for these utilities.

When deploying the Common Auditing and Reporting Service in a clustered environment, run the staging utility and XML data store utilities on the deployment manager.

Preparing to run the server utilities

This topic describes how to set the CLASSPATH environment variable for the staging and data store utilities. These settings are necessary before you run the server utilities.

Procedure

Set the CLASSPATH variable to include the following file paths:

AIX, Linux, or Solaris systems

```
CARS_HOME/server/etc:
CARS_HOME/server/lib/ibmcars.jar:
DB2_HOME/java/db2jcc.jar:
DB2_HOME/java/db2jcc_license_cu.jar:
DB2INSTANCE_OWNER/sqllib/java/db2java.zip:
DB2INSTANCE_OWNER/sqllib/java/db2jcc.jar:
DB2INSTANCE_OWNER/sqllib/function:
DB2INSTANCE_OWNER/sqllib/java/db2jcc_license_cu.jar:
```

where:

CARS_HOME
Specifies the installation directory of the Common Auditing Service server. By default, the location is the `/opt/IBM/Tivoli/CommonAuditService` directory.

DB2_HOME
Specifies the installation directory of the DB2 server.

DB2INSTANCE_OWNER
Specifies the home directory of the DB2 instance owner.

Windows systems

```
CARS_HOME\server\etc;
CARS_HOME\server\lib\ibmcars.jar;
DB2_HOME\java\db2jcc.jar;
DB2_HOME\java\db2jcc_license_cu.jar;
DB2_HOME\java\db2java.zip;
DB2_HOME\function;
```

where:

CARS_HOME
Specifies the installation location of the Common Auditing Service server. The default location is C:\Program Files\IBM\Tivoli\CommonAuditService.

DB2_HOME
Specifies the installation directory of the DB2 server. The default location is C:\Program Files\IBM\SQLLIB.

Using the default installation directories, you could set the CLASSPATH variable by entering the following command on a single line:

```
set CLASSPATH=
c:\progra~1\ibm\Tivoli\CommonAuditService\server\etc;
c:\progra~1\ibm\Tivoli\CommonAuditService\server\lib\ibmcars.jar;
c:\progra~1\ibm\sqllib\java\db2jcc.jar;
c:\progra~1\ibm\sqllib\java\db2jcc_license_cu.jar;
c:\progra~1\ibm\sqllib\java\db2java.zip;
c:\progra~1\ibm\sqllib\function;
%CLASSPATH%;
```

Running the staging utility command

The staging utility provides staging of the data from the XML data store to the staging tables. You can stage data in historical, incremental, or prune mode.

Syntax

Use the following command syntax to run the staging utility.

java com.ibm.cars.staging.Staging -mode historical -starttime *value* **-endtime** *value*

java com.ibm.cars.staging.Staging -mode incremental

java com.ibm.cars.staging.Staging -mode prune -prunetime *value*

Parameters

You can specify the parameters shown in the syntax above and also the following optional parameters on the command line or in the ibmcars.properties file. For a description of each parameter, see "Configuration parameters for the staging utility and XML data store utilities" on page 72.

-configurl *value*

-dbhostname *value*

-dbport *value*

-dbname *value*

-dbusername *value*

-dbpassword *value*

-batchsize *value*

-numworkers *value*

-progress *value*

-help

If you do not set a specific parameter and value in the command, the utility searches for the parameter and value in the ibmcars.properties file. The parameter values that you specify on the command line override any parameter values that are specified in the `ibmcars.properties` file.

Historical mode

When you use historical mode, all events in a specified time range are staged. For this mode, you must specify the start and end time for the staging utility.

The following example shows running historical staging beginning on January 1, 2012 at 10:00 PM through October 6, 2012 at 10:00 PM:

```
java com.ibm.cars.staging.Staging -mode historical
  -starttime "Jan 1, 2012 10:00:00 PM GMT"
  -endtime "Oct 6, 2012 10:00:00 PM GMT"
```

Incremental mode

When you use incremental mode, all new events since the last incremental staging are staged. If incremental staging has never run, all events are staged.

The following example shows running incremental staging:

```
java com.ibm.cars.staging.Staging -mode incremental
```

Prune mode

When you use prune mode, all events older than the specified time are deleted (*pruned*) from the staging tables. For this mode, you must specify the time and date for which all prior events are pruned.

The following example deletes events from the staging tables that are older than October 6, 2011 at 12:00 AM:

```
java com.ibm.cars.staging.Staging -mode prune
  -prunetime "Oct 6, 2011 12:00:00 AM GMT"
```

Note: Run only one staging utility instance at a time; otherwise, the operation (in the case of incremental and historical staging) could fail. If you need more parallelism, increase the number of threads (workers) instead of running another instance of the staging utility.

Return codes

If there is an unrecoverable error during the staging process, the staging utility halts execution. An error can have any number of causes, such as a full database transaction log or full disk space. Correct the situation that caused the error and rerun the staging utility. The return code of the staging utility is 0 on success (the staging utility has completed its work or the **`-help`** parameter was specified), and 1 on error (the staging utility has not completed its work).

Running the XML data store utilities

The XML data store utilities provide tools to manage the XML data store in preparation for archival, and to clean up restored data that is no longer needed. You can run three utilities: pre-archive, post-archive, and clean restore table set.

Notes

- Make a note of the first and last timestamp because you need this information when you want to prune the report tables. When you run the **XMLStoreUtils** program for the first time, you get an exception because there is no data to archive.
- The settings for the XML data store utility parameters are determined in the following order:
  1. Check the XML data store utility settings specified on the command line.
  2. Check the settings in the `ibmcars.properties` file.
  3. Check the default settings in the code.

Syntax

Use the following command syntax for each of the XML data store utilities:

java com.ibm.cars.xmlstoreutils.XmlStoreUtils **-operation prearchive**

java com.ibm.cars.xmlstoreutils.XmlStoreUtils **-operation postarchive** [**-mode force**] [**-copydir** *value*]

java com.ibm.cars.xmlstoreutils.XmlStoreUtils **-operation cleanrestore** [**-mode force**]

Parameters

The following parameters can also be specified on the command line or in the `ibmcars.properties` file. For a description of each parameter, see "Configuration parameters for the staging utility and XML data store utilities" on page 72.

-configurl *value*

-dbhostname *value*

-dbport *value*

-dbname *value*

-dbusername *value*

-dbpassword *value*

-dbbackup *value*

-copydir *value*

-help

If you do not set a specific parameter and value in the command, the utility searches for the parameter and value in the `ibmcars.properties` file. The parameter values that you specify on the command line override any parameter values that are specified in the `ibmcars.properties` file.

Prearchive operation

Use the prearchive operation before archiving data from the XML data store tables. The prearchive operation prints out the data needed for archiving, such as:

- The names of the XML data store tables to archive.
- The first date contained in the tables to be archived. For example: Jan 1, 2012 5:30:00 AM
- The last date contained in the tables to be archived. For example: Jan 2, 2012 3:42:03 PM

Postarchive operation

After you finish archiving XML data store tables, use the postarchive operation to remove the data from the inactive XML data store tables. The postarchive operation prompts for confirmation to purge the data from the XML data store tables. For silent mode operation, specify **–mode force**, which forces the postarchive operation without a confirmation prompt. Postarchive completes the following actions:

- Purges the data from the target XML data store tables.
- Updates the cei_t_properties table with the current active bucket number, in which the value is swapped from 0 to 1, and vice and versa.

The audits that are purged from the XML audit store tables are not available for drilldown reporting. Before running the postarchive operation, use the staging utility prune operation to remove the report table data for audits that range within the begin date and the end date as provided by the prearchive operation. See "Running the staging utility command" on page 66.

Cleanrestore tables operation

Use the cleanrestore operation when the audits in the restore tables are no longer required. The cleanrestore operation prompts for confirmation that the data in the restore tables will be cleaned and will no longer be available. For silent mode operation, specify **–mode force**, which forces the cleaning of the restore tables without a confirmation prompt.

Samples

The following command provides help information for the XML data store utility:

```
java com.ibm.cars.xmlstoreutils.XmlStoreUtils -help
```

The following command completes the prearchive operation:

```
java com.ibm.cars.xmlstoreutils.XmlStoreUtils -operation prearchive
```

The following command completes the postarchive operation and bypasses the prompts. If the database server has archive logging configured, the XML data store utility backs up the data to the `C:\foo` directory. If the database server has circular logging enabled, the XML data store utility ignores the **copydir** parameter and backs up the data to the `C:\foo` directory.

```
java com.ibm.cars.xmlstoreutils.XmlStoreUtils -operation postarchive -mode force
 –copydir C:\\foo
```

The following command completes the cleanrestore tables operation and bypasses the prompts:

```
java com.ibm.cars.xmlstoreutils.XmlStoreUtils -operation cleanrestore -mode force
```

The ibmcars.properties file

The `ibmcars.properties` file contains configuration properties for the staging utility and XML data store utility. Update the value in the *property=value* entry to make a change.

The `ibmcars.properties` file is in *`CARS_HOME`*`\server\etc` on Windows and *`CARS_HOME`*`/server/etc` on AIX, Linux, or Solaris systems, where *CARS_HOME* is the installation directory of Common Auditing Service.

Sample

Following is a sample ibmcars.properties file:

```
#
# Licensed Materials - Property of IBM
# 5748-XX8
# (c) Copyright International Business Machines Corp. 2012
# All Rights Reserved
# US Government Users Restricted Rights - Use, duplicaion or disclosure
# restricted by GSA ADP Schedule Contract with IBM Corp.
#

# This file contains configuration properties for the CARS Staging and
# XML store utilities.
# The format is "property=value" on a single line.
# A line or a portion of a line beginning with "#" is ignored (comment)


######  General configuration properties

#  util.eventBatchSize denotes the number of events that the staging
#  utility should process in a single batch, for both staging and pruning
#  operations. The default is 1000, which should be adequate for most situations.
#  A value too low will increase the number of transactions, potentially
#  reducing performance; a value too high might result in an overflow of the
#  DB2 transaction log.
#  This option can be specified on the command line with "-batchsize"
util.eventBatchSize=100

#  util.db.hostname denotes the database server host name that the utility will
#  use to connect to the database. The default is localhost.
#  This option can be specified on the command line with "-dbhostname".
#util.db.hostname=<hostname>

# util.db.port specifies the port number on which the DB2 database instance
# is listening. # This option can be specified on the command line with "-dbport".
#util.db.port=50000

#  util.db.name denotes the name of the event database. The default
#  value is "eventxml".
#  This option can be specified on the command line with "-dbname".
#util.db.name=eventxml

#  util.db.user denotes the user name that the utility will
#  use to connect to the database. This user needs to be the owner of the
#  database instance where the event database resides.There is no default
#  for this option.
#  This option can be specified on the command line with "-dbusername".
#util.db.user=<username>

#  util.db.passwd denotes the password for the database user name
#  specified under "util.db.user". There is no default for this option.
#  This option can be specified on the command line with "-dbpassword".
#util.db.passwd=<password>

#  util.startTime denotes the start time for the historical staging
#  interval. Acceptable timestamps are valid time specifiers in the current
#  locale; for example "Jan 1, 2012 10:00:00 PM GMT" for US English. If the
#  specified time cannot be parsed, the staging utility will suggest the
#  proper format. A value is required when the execution mode is historical
#  staging; the property is ignored otherwise. There is no default
#  for this property.
#  This option can be specified on the command line with "-starttime".
#util.startTime=Jan 1, 2012 10:00:00 PM GMT

#  util.endTime denotes the end time for the historical staging
#  interval. Acceptable timestamps are valid time specifiers in the current
#  locale; for example "Jan 1, 2007 10:00:00 PM GMT" for US English. If the
#  specified time cannot be parsed, the staging utility will suggest the
#  proper format. A value is required when the execution mode is historical
#  staging; the property is ignored otherwise. There is no default
#  for this property.
#  This option can be specified on the command line with "-endtime".
#util.endTime=Jan 1, 2012 10:00:00 PM GMT

#  util.pruneTime denotes the prune threshold time for event
#  pruning. Events older than this time will be removed from the staging
#  database. Acceptable timestamps are valid time specifiers in the current
#  locale; for example "Jan 1, 2012 10:00:00 PM GMT" for US English. If the
#  specified time cannot be parsed, the staging utility will suggest the
#  proper format. A value is required when the execution mode is pruning;
#  the property is ignored otherwise. There is no default for this
#  property.
#  This option can be specified on the command line with "-prunetime".
#util.pruneTime=Jan 1, 2012 10:00:00 PM GMT

#  util.numworkers denotes the number of threads that the staging
#  utility will use to perform work in parallel. This value must be an integer
#  and it must be at least 1. The default value is 1. For best performance,
#  use a value one greater than the number of CPUs in the machine (e.g., on
#  a machine with four CPUs, specify five workers). A value too low might
#  result in suboptimal use of the available CPUs, while a value too high
#  might result in high context switching overhead.
#  This option can be specified on the command line with "-numworkers".
util.numworkers=1

#  util.progress controls whether, and how often, the staging
#  utility reports progress on the console (standard output). If a value of
#  N greater than 0 is specified, the staging utility will report progress
#  whenever at least N events have been processed since the last progress
#  report. Note that progress reports might be less frequent than every N
#  events; for example, if the event batch size parameter is larger than N,
#  progress will be reported roughly after every batch. If the value of the
#  progress parameter is 0, progress will not be reported (this is the
#  default behavior).
#  This option can be specified on the command line with "-progress".
util.progress=0

#  util.DriverClassName is used by XmlStoreUtils in forming the url string to
#  be used to connect to the database.
util.DriverClassName=com.ibm.db2.jcc.DB2Driver

#  util.DriverType is used by XmlStoreUtils in forming the url string to
#  be used to connect to the database.
util.DriverType=jdbc:db2:

#  util.db.backup will be used by the post archive utility.
#  Consult your database Administrator to determine your database logging
#  and backup configuration settings.
#
#  options - circular, archive
#  circular - database circular logging is enabled - default
#  archive - database archive logging is enabled - the copydir parameter is
#  required using this option
#  util.db.backup=<archive|circular>
util.db.backup=circular

#  util.db.copydir will be used by the post archive utility to decide if
#  the utility needs to back up the data of the inactive table at the
#  specified location before purging the data.
#  This is optional. This can also be given as a command line argument.
#  For example,
#  On Unix set  "util.db.copydir=/opt/test"
#  On Windows set "util.db.copydir=c:\\test"
#util.db.copydir=<path>

#  util.WasHome points to the WebSphere AppServer path.
#  WasHome is used by the XmlStoreUtils to locate CEI scripts
#  for managing buckets.
#  For example,
#  On Unix set "util.WasHome=/opt/IBM/WebSphere/AppServer"
#  On Windows set "util.WasHome=C:\\Program Files\\WebSphere\\AppServer"
#util.WasHome=<path>

####### Tracing and Logging properties

# See the general documentation for configuring CARS JLog for details
# on the properties below
baseGroup.CBAStagUtilTraceLogger.isLogging=false
baseGroup.CBAStagUtilTraceFileHandler.fileName=trace__StagUtil.log
baseGroup.CBAStagUtilMessageFileHandler.fileName=msg__StagUtil.log

baseGroup.CBAStagUtilMessageAllMaskFilter.parent=CBAMessageAllMaskFilter
baseGroup.CBAStagUtilMessageFileHandler.parent=CBAMessageFileHandler
baseGroup.CBAStagUtilTraceFileHandler.parent=CBATraceFileHandler

baseGroup.CBAStagUtilTraceLogger.parent=CBATraceLogger
baseGroup.CBAStagUtilTraceLogger.name=CBAStagUtilTraceLogger
baseGroup.CBAStagUtilTraceLogger.description=Common StagUtil Trace Logger
baseGroup.CBAStagUtilTraceLogger.component=StagUtil
baseGroup.CBAStagUtilTraceLogger.handlerNames=CBAStagUtilTraceFileHandler
baseGroup.CBAStagUtilTraceLogger.filterNames=CBAStagUtilTraceAllMaskFilter
  CBATraceClassFilter

baseGroup.CBAStagUtilMessageLogger.parent=CBAMessageLogger
baseGroup.CBAStagUtilMessageLogger.name=CBAStagUtilMessageLogger
baseGroup.CBAStagUtilMessageLogger.isLogging=true
baseGroup.CBAStagUtilMessageLogger.description="Common StagUtil Message Logger"
baseGroup.CBAStagUtilMessageLogger.component=StagUtil
baseGroup.CBAStagUtilMessageLogger.handlerNames=CBAStagUtilMessageFileHandler
baseGroup.CBAStagUtilMessageLogger.filterNames=CBAStagUtilMessageAllMaskFilter
  CBAMessageClassFilter

baseGroup.CBAStagUtilTraceAllMaskFilter.parent=CBATraceAllMaskFilter
baseGroup.CBAStagUtilTraceAllMaskFilter.mask=9
baseGroup.CBAStagUtilMessageAllMaskFilter.mask=FATAL | ERROR | WARNING |
  NOTICE | NOTICE_VERBOSE

baseGroup.CBAStagUtilTraceClassFilter.description="Common StagUtil Trace
  Class Filter"
baseGroup.CBAStagUtilTraceClassFilter.className=com.ibm.cars.ras.csjlog.
  CSClassFilter

baseGroup.CBAStagUtilMessageClassFilter.description=Common Audit Service
  Class Filter
baseGroup.CBAStagUtilMessageClassFilter.className=com.ibm.cars.ras.csjlog.
  CSClassFilter

baseGroup.CBAStagUtilTraceFileHandler.description=Common StagUtil Trace File
  Handler
```

Configuration parameters for the staging utility and XML data store utilities

For the staging utility and XML data store utilities, you can specify the parameters on the command line or set them in the ibmcars.properties file.

The following list shows each parameter, how you can specify it (in the command line or in the configuration file, or both), and the accepted values.

Configuration file URL
Specifies the location of Common Auditing Service configuration file.
Command (staging and XML data store)
`-configurl` *`value`*
Configuration
Not used.
Value Valid location. The default is *`CARS_HOME`*`/Server/etc/ibmcars.properties, where` *`CARS_HOME`* `is the installation directory of the Common Auditing Service.`

DB backup
Specifies the database logging and backup configuration settings. By default this parameter is set to circular. Consult your database administrator to determine the value for this parameter.
Command (XML data store)
`dbbackup` *`value`*
Configuration
`util.db.backup=`*`value`*
Value `circular` or `archive`

Copy directory
Specifies the path to a directory to be used for the files generated by the load utility.

This parameter is required only if you have enabled forward recovery for the eventxml database (XML data store) with the LOGRETAIN or USEREXIT database configuration settings enabled. By default, the eventxml database does not use forward recovery.

Refer to the DB2 documentation for further details on how to enable the eventxml database for roll forward recovery.
Command (XML data store)
`-copydir` *`value`*
Configuration
`util.db.copydir=`*`value`*
Value Valid directory.
Linux or UNIX
`util.db.copydir=/opt/test`
Windows
`util.db.copydir=c:\test`

Database instance owner ID
Denotes the user name that the utility will use to connect to the database. This user needs to be the owner of the database instance where the XML data store is located.
Command (staging and XML data store)
`-dbusername` *`value`*
Configuration
`util.db.user=`*`value`*
Value Valid user name.

Database host name
Denotes the database server host name where DB2 is running.
Command (staging and XML data store)
`-dbhostname` *`value`*

Configuration
`util.db.hostname=`*`value`*

Value Valid host name or IP address. The default is `localhost`.

Database instance owner password

Denotes the password for the database user name specified under util.db.user or -dbusername.

Command (staging and XML data store)
`-dbpassword` *`value`*

Configuration
`util.db.passwd=`*`value`*

Value Correct password for the specified user.

Database name

Denotes the name of the audit database.

Command (staging and XML data store)
`-dbname` *`value`*

Configuration
`util.db.name=`*`value`*

Value Valid database name. The default is `eventxml`.

Database port number

Specifies the port number on which the DB2 instance is listening. This should be the main connection port configured on the DB2 server.

Command (staging and XML data store)
`-dbport` *`value`*

Configuration
`util.db.port=`*`value`*

Value Integer. The default is `50000`.

Driver class name

Specifies the driver class name and is used by the XML data store utility in forming the URL string to be used to connect to the database.

Command
Not used.

Configuration
`util.DriverClassName=`*`value`*

Value Valid driver class name. For example:

```
util.DriverClassName=com.ibm.db2.jcc.DB2Driver
```

Driver type

Specifies the driver type and is used by the XML data store utility in forming the URL string to be used to connect to the database.

Command
Not used.

Configuration
`util.DriverType=`*`value`*

Value Valid driver type. For example:

```
util.DriverType=jdbc:db2:
```

End time

Specifies the end time when the staging utility is launched in historical mode. Usually used when reporting or archiving data.

Command (staging)
`-endtime` *`value`*

Configuration
`util.endTime=`*`value`*

Value Valid timestamp in the following format:

`*mmm dd*, *yyyy hh*:*mm*:*ss am_or_pm* GMT`

For example:

```
Jan 12, 2007 10:00:00 PM GMT
```

Event batch size
Denotes the number of security events that the staging utility should process in a single batch, for both staging and pruning operations.
Command (staging)
`-batchsize *value*`
Configuration
`util.eventBatchSize=*value*`
Value Positive integer. The default is `100`.

Help Provides usage information for the utilities.
Command (staging and XML data store)
`-help`
Configuration
`Not used.`
Value None.

Logging flag
Specifies if logging is turned on.
Command
Not used.
Configuration
`baseGroup.CBAStagingUtilMessageLogger.isLogging=*value*`
Value Possible values are:
- `true`
- `false`

The default is `true`.

Message file name

Name of the message file.
Command
Not used.
Configuration
`baseGroup.CBAStagUtilMessageFileHandler.fileName=*value*`
Value Valid file name. The default is `msg__StagUtil.log`.

Number of workers
Denotes the number of threads that the staging utility will use to complete work in parallel. The value for best performance is the number of processors of the machine that contains the database, plus one.
Command (staging)
`-numworkers *value*`
Configuration
`util.numworkers=*value*`
Value Positive integer. The default is 1.

Operation type

Determine which type of operation to complete for the XML data store utilities:
Pre-archive
Use before archiving data from the XML data store tables. The prearchive operation prints out the data needed for archiving, such as the names and the dates of the tables to archive.

Post-archive
Use to remove the data from the inactive XML data store tables.

Clean restore table set
Use to clear the security events in the restore table set when they are no longer required.

Command (XML data store)
`-operation` *`value`*

Configuration
Not used.

Value Possible values are:
- `prearchive`
- `postarchive`
- `cleanrestore`

Progress report
Controls whether, and how often, the staging utility reports progress on the console (standard output). If a value of *N* security events greater than 0 is specified, the staging utility will report progress whenever at least *N* security events have been processed since the last progress report. Note that progress reports might be less frequent than every *N* security events.

For example, if the event batch size parameter is larger than *N*, progress will be reported roughly after every batch. If the value of the progress parameter is 0, progress will not be reported (this is the default behavior).

Command (staging)
`-progress` *`value`*

Configuration
`util.progress=`*`value`*

Value An integer greater than or equal to 0.

The default is `0`.

Prune threshold time
Denotes the prune threshold time for event pruning. Security events older than this time will be removed from the staging database.

Command (staging)
`-prunetime` *`value`*

Configuration
`util.pruneTime=`*`value`*

Value Valid timestamp in the current locale. For example in US English:

```
Jan 12, 2012 10:00:00 PM GMT
```

Staging utility execution mode
Specify under what mode the staging utility runs:

Incremental
New security events since the last incremental staging are staged. If incremental staging has never run, all security events are staged.

Historical
All security events in a specified time range are staged.

Prune All security events older than a specified time are pruned.

Command (staging)
`-mode` *`value`*

Configuration
`util.mode=`*`value`*

Value Possible values are:
- `historical`
- `incremental`
- `prune`

The default is `incremental`.

Start time

Specifies the start time when the staging utility is launched in historical mode. Normally used when reporting or archiving data.

Command (staging)

`-starttime` *value*

Configuration

`util.startTime=`*value*

Value Valid timestamp in the following format:

mmm dd, *yyyy hh*:*mm*:*ss am_or_pm* GMT

For example:

`Jan 12, 2012 10:00:00 PM GMT`

Trace file name

Specifies the name of the trace file.

Command

Not used.

Configuration

`baseGroup.CBAStagUtilTraceFileHandler.fileName=`*value*

Value Valid file name. The default is `trace__StagUtil.log`.

Tracing flag

Specifies if tracing is turned on.

Command

Not used.

Configuration

`baseGroup.CBAStagUtilTraceLogger.isLogging=`*value*

Value Possible values are:

- `true`
- `false`

The default is `false`.

Chapter 7. Unconfiguring and uninstalling Common Auditing and Reporting Service

This topic describes how to uninstall the Common Auditing and Reporting Service 7.0 audit server, configuration console, and configuration utilities.

The uninstallation of a Common Auditing and Reporting Service feature involves the following tasks:

- Reviewing the uninstallation checklist
- Uninstalling the selected feature with either the interactive or silent uninstallation

Note: You must run the uninstallation program to uninstall the audit server or the configuration console. Removing the directory where the feature is installed does not completely uninstall the feature.

If an uninstallation of the server fails, you must complete the steps that are described in "Failed uninstallation workarounds" on page 361 to remove the product from your system.

Unconfiguring Common Auditing Service

This topic describes how to unconfigure the Common Auditing and Reporting Service Version 7.0 audit server, configuration console, and configuration utilities that use the Integrated Solutions Console (ISC) module plug-in to the WebSphere Application Server Administrative Console. *NOTE: You must unconfigure Common Auditing and Reporting Service before you uninstall it.*

Procedure

1. Disconnect all applications from the DB2 database used as the XML data store. The following commands show an example of how to restart DB2 and ensure that no applications are connected:

   ```
   db2stop force
   db2start db2
   db2 list applications
   ```

2. Open a web browser and set the value of the URL to the administrative console port of the WebSphere Application Server Deployment Manager or stand-alone server that was specified as the target profile during installation (default port value is 9060 or 9043 for a secure console).

   Example: `http://webspherserver.ibm.com:9060/ibm/console`

3. Log in as a WebSphere Application Server administrator.
4. Go to the CARS unconfiguration wizard, **Common Audit Service-> Audit Service unConfiguration**, to start the unconfiguration wizard.
5. Proceed through the windows as described in the following steps. The options presented in each window are described in "Common Audit Service configuration options" on page 40. The Welcome dialog is displayed, indicating that Common Auditing Service must be unconfigured before the application can be uninstalled.
6. Click **Next** to continue.

7. In the Audit Service Host window, enter the host name and SOAP port number of the target WebSphere Application Server process (Deployment Manager or stand-alone single server) where Common Auditing and Reporting Service will be unconfigured.
8. Click **Next** to continue.
9. In the **WebSphere Security** window, if global security is enabled on the target WebSphere Application Server process, select the **Global Security** check box, then enter the WebSphere Application Server administrator name and password.
10. Click **Next** to continue.
11. In the **WebSphere Target Mapping** window, select the path of the WebSphere Application Server deployment target where Common Auditing Service is deployed. The list of clusters and independent servers that are available for undeployment are displayed in the dropdown list. You must select an entry from the list.
12. Click **Next** to continue.
13. In the **Audit Database** window, the configured values for the database instance owner ID, XML datastore name, and TCP/IP service port are displayed.

    You must specify the database instance owner password. If you want to remove the Audit database, select **Remove Audit Database**.

    By default, the audit database is *not* removed. If the database is removed, all staging tables related to the database are also removed. Note that the path to the JDBC driver and the data source information in WebSphere Application Server that is used to establish a connection to the database is removed, regardless if the database is retained or removed.

    To re-establish the JDBC connection, you must specify the path of the JDBC driver in the Create JDBC Connector window when you reconfigure after a new installation.
14. Click **Next** to continue.
15. Review the list of options that you selected in the Summary window. If the options are correct, select **Finish** to begin the unconfiguration. If one or more options are incorrect, use **Back** to return to a window and make the appropriate changes.
16. Review the Common Audit Service Status window to determine the outcome of the unconfiguration. If the unconfiguration was unsuccessful, correct the problems and start the unconfiguration again from the Welcome window. Click **OK** to return to the Welcome window.

Unconfiguring and uninstalling Common Auditing and Reporting Service

This topic describes how to uninstall the Common Auditing and Reporting Service 7.0 audit server, configuration console, and configuration utilities.

The uninstallation of a Common Auditing and Reporting Service feature involves the following tasks:

- Reviewing the uninstallation checklist
- Uninstalling the selected feature with either the interactive or silent uninstallation

Note: You must run the uninstallation program to uninstall the audit server or the configuration console. Removing the directory where the feature is installed does not completely uninstall the feature.

If an uninstallation of the server fails, you must complete the steps that are described in “Failed uninstallation workarounds” on page 361 to remove the product from your system.

Uninstallation checklist for all platforms

This topic lists the tasks that you must complete before you attempt to uninstall Common Auditing Service.

- Before you start the uninstallation wizard to remove either feature (audit server or configuration console) of Common Auditing Service, determine if you want to keep or remove the database that is used as your XML data store.
- If you want to remove the audit server but maintain the database, run the unconfiguration wizard and only undeploy Common Auditing Service from the WebSphere Application Server profile (select to keep the database intact).
- If you want to completely remove Common Auditing Service from a system, run the unconfiguration wizard and undeploy the audit server from the WebSphere Application Server profile and select to remove the database and staging tables as well.

  Note: If the Common Auditing Service is not fully unconfigured before starting uninstallation, a warning message will be displayed in the uninstallation window that informs you to completely unconfigure the Common Auditing Service components before you uninstall the Common Auditing Service. If you continue with the uninstallation, you will need to manually remove the server components after uninstallation.
- The procedure for manually removing the audit server components after a successful uninstallation is the same procedure for manually removing the audit server components after a failed uninstallation. The manual uninstallation procedures are described in “Failed uninstallation workarounds” on page 361.
- After you have successfully undeployed Common Auditing Service, you can run the uninstallation wizard to remove the product files and registry entries.

Interactive uninstallation

This section describes the interactive uninstallation of the audit server. The interactive uninstallation gives you the option to use GUI windows or use console mode on the command line.

Starting the uninstallation wizard

This topic describes the command syntax used to start the Common Auditing Service interactive uninstallation wizard in either graphical or console (command line) mode.

Before you begin

Ensure that you undeploy Common Auditing Service from the WebSphere Application Server; see “Uninstallation checklist for all platforms” for more information about undeploying Common Auditing Service.

Before running the interactive uninstallation, follow these steps:

1. Change to the directory where the audit server was installed. For example:

- **Windows:** `c:\Program Files\IBM\Tivoli\CommonAuditService`
- **AIX, Linux, or Solaris:** `/opt/IBM/Tivoli/CommonAuditService`

If you did not use the default directory, change to the directory you chose for your audit server installation location.

2. From the audit server installation directory, change to the `_uninst` directory.

Command syntax

To run the uninstallation in interactive mode, enter the following command:

```
java -cp uninstall.jar run [-console] [-options-record response_file]
[-is:javahome java_home]
```

Parameters

-console
Run the program in console mode, specifying options on the command line. If you do not specify **-console**, the GUI panel uninstallation starts.

-options-record *response_file*
Generate a response file with the options you choose on each panel and write it to the specified file. After you run this interactive uninstallation, you can then use this response file to run a silent uninstallation as it will contain all of the appropriate parameters and values.

-is:javahome *java_home*
Specify the home directory of the Java Virtual Machine that the uninstallation launcher uses.

Sample

An example of using the Windows command to uninstall the audit server by using console mode:

```
java -cp uninstall.jar run -console
```

Interactive uninstallation using the GUI windows

Follow this procedure to complete an interactive uninstallation of Common Auditing Service using the GUI windows.

Before you begin

See "Starting the uninstallation wizard" on page 81 for the command you enter to begin the uninstallation wizard.

Procedure

1. Select the language that you want to use for the installation.
2. Click **OK**. The Welcome dialog is displayed.
3. Click **Next** to continue.
4. In the **Features** window, select both features to uninstall the audit server, configuration console, and configuration utilities. Select **Common Audit Service** to uninstall the audit server and configuration utilities only. Select **Common Audit Service Configuration Console** to uninstall only the configuration console.
5. Click **Next** to continue.

6. If WebSphere Application Server global security is set, you are prompted to enter the WebSphere Application Server administrator ID and password in the WebSphere Application Server Security Details window.
7. Click **Next** to continue.
8. In the **Summary** window, check that the location is correct for the selected features for uninstallation. Click **Back** if you need to change a setting.
9. Click **Next** to begin the uninstallation.
10. The final window shows that the uninstallation was successful or indicates error logs to identify any uninstallation problems.

Silent uninstallation

Follow this procedure to complete a silent uninstallation of the audit server. The silent uninstallation processes the choices in the response file and returns the command prompt when complete. No on-screen messages display during the execution of the silent uninstallation.

Before you begin

Before you run the uninstallation program in silent mode, follow these steps:

1. Add the following lines to the response file to remove the XML descriptor files and the directory in which they were installed:
   ```
   -G removeModifiedResponse="yesToAll"
   -G removeExistingResponse="yesToAll"
   ```
2. Change to the directory where the audit server was installed. For example:
   - **Windows:** `c:\Program Files\IBM\Tivoli\CommonAudit`
   - **AIX, Linux, or Solaris:** `/opt/IBM/Tivoli/CommonAudit`

   If you did not use the default directory, change to the directory you chose for your audit server installation location.
3. From the audit server installation directory, change to the `_uninst` directory.

Syntax

To run the silent uninstallation, the following command for your operating system:

For Windows

```
java -cp uninstall.jar run -silent -options response_file
```

For AIX, Linux, or Solaris

```
java -cp uninstall.jar run -silent -options response_file
```

Parameters

`-options` *response_file*

Specifies the name of the response file to use. For example, `serverUninstall.rsp`.

Sample

An example of using the Windows command with a response file named `serverUninstall.rsp` follows:

```
java -cp uninstall.jar run -silent -options serverUninstall.rsp
```

Final step

The following step is required after you run the silent uninstallation:

- Restart WebSphere Application Server after the uninstallation process is complete.

Uninstalling language support packages

Uninstall the language packs if you no longer need the language support for your environment.

Procedure

1. Change to one of the following directories:

   AIX, Linux, and Solaris operating systems:
   ```
   /opt/IBM/Tivoli/CommonAuditService/CARSLP/lp_uninst
   ```

   Windows operating systems:
   ```
   C:\Program Files\IBM\Tivoli\CommonAuditService\CARSLP\lp_uninst
   ```

2. Uninstall the language support packages with the following command:

   AIX, Linux, and Solaris operating systems:
   ```
   jre_path/java -jar cars_lp_uninstall.jar
   ```

   Windows operating systems:
   ```
   jre_path\java -jar cars_lp_uninstall.jar
   ```

   where *jre_path* is the path where the Java executable program is located. If the Java executable program is in the path, you do not have to specify *jre_path*.

Part 3. Operational reports

Chapter 8. Introduction to the auditing reports

Use Tivoli Common Reporting to analyze important information in your environment.

Before you can generate reports, you must install and configure Common Auditing and Reporting Service, and Tivoli Common Reporting.

Security Access Manager provides a predefined set of BIRT reports in a *report package*. These reports enable you to use the features of Tivoli Common Reporting. You must import the report package into the Tivoli Common Reporting environment to run the reports in the package.

The Security Access Manager report package is on the *IBM Security Access Manager for Web Version 7.0* DVD.

To set up your environment to use reports, see Chapter 9, “Setting up the Security Access Manager reporting environment,” on page 89.

Reports in the report package are described in Chapter 11, “Report descriptions,” on page 95.

To learn how to use Tivoli Common Reporting to generate the operational reports, see Chapter 10, “Running the Security Access Manager operational reports,” on page 93.

Chapter 9. Setting up the Security Access Manager reporting environment

This topic describes the following tasks that you must perform to set up Security Access Manager to generate operational reports.

Before you begin

A *report package* is a `.zip` file containing all of the data required for defining one or more reports, including the required designs and resources and the hierarchy of report sets to contain the reports. The Security Access Manager BIRT reports are included in a report package on the *IBM Security Access Manager for Web Version 7.0* product image or DVD for your operating system.

The Security Access Manager BIRT report package is called `ISAM_TCRReportPackage_7.0.0.0.zip`. This package is required to complete this procedure.

The `ISAM_TCRReportPackage_7.0.0.0.zip` package is located under the *`image_path/platform`* directory where *image_path* is where the product image is downloaded, or DVD is mounted, and *platform* is the one of the following:

- `linux_x86`
- `linux_s390`
- `solaris`
- `usr/sys/inst.images` (AIX)
- `windows`

About this task

This procedure lists the high-level steps required to set up the Security Access Manager BIRT reports.

Procedure

1. Run the Common Auditing and Reporting Service staging utility to stage the event data for reporting. See Chapter 6, “Running the server utilities,” on page 65 for information about using the staging utility.
2. Install Tivoli Common Reporting, version 2.1.1.

   For information on installing Tivoli Common Reporting, version 2.1.1, see the *Tivoli Common Reporting Installation and Upgrade Guide* at: http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?topic=%2Fcom.ibm.tivoli.tcr.doc_211%2Fic-home.html.
3. Start the Tivoli Common Reporting Server. See "Starting the Tivoli Common Reporting server" in the *Tivoli Common Reporting Installation and Upgrade Guide* for information about starting Tivoli Common Reporting.
4. Import the Security Access Manager report package into Tivoli Common Reporting. See “Importing the Security Access Manager report package” on page 90 for information about importing a report package.

5. Configure the Tivoli Common Reporting data source to provide the event data for the reports. See “Configuring Tivoli Common Reporting to access the staged audit data” for information about configuring Tivoli Common Reporting to access audit event data.

What to do next

Before running the General Audit Event Details Report, complete the procedures to install and deploy the Java stored procedure on the DB2 server.

The installation and deployment of the Java stored procedure is part of the installation and configuration of the Common Auditing and Reporting Service audit server. You can find a description in Chapter 5, “Installing, configuring, and upgrading the Common Auditing Service audit server,” on page 25.

Importing the Security Access Manager report package

This topic describes how to import the Security Access Manager report package into Tivoli Common Reporting.

About this task

See "Importing BIRT reports" in the *Tivoli Common Reporting User's Guide* for more information about importing report packages into Tivoli Common Reporting. With release 2.1.1, Tivoli Common Reporting uses command-line actions only to import BIRT reports.

Take these steps to import the `ISAM_TCRReportPackage_7.0.0.0.zip` report package.

Procedure

1. Navigate to the following directory:
   - On Windows:
     `C:\ibm\tivoli\tipv2Components\TCRComponent\bin`
   - On AIX, Linux, or Solaris:
     `/opt/IBM/tivoli/tipv2Components/TCRComponent/bin`
2. Run the following command:
   ```
   trcmd.{bat|sh} -import -bulk ISAM_TCRReportPackage_7.0.0.0.zip
   -user adminid -password adminpwd
   ```

   where *adminid* is the IBM Tivoli Common Reporting user ID (For example, `tipadmin`) and *adminpwd* is the password for the IBM Tivoli Common Reporting user ID.

Configuring Tivoli Common Reporting to access the staged audit data

Security Access Manager reports require that you configure JDBC data sources in order to access the staged audit event data.

The data sources that you must configure include:

JDBC provider
You can use scripting to configure the JDBC provider.

JDBC data source
The data source is the DB2 database that contains the staged audit data.

You can optionally use scripting to configure the JDBC data sources. See "Configuring JDBC data sources for direct access" in the *Tivoli Common Reporting User's Guide* for configuration steps and pointers to examples.

The following procedure uses commands to configure the JDBC data sources. This procedure uses one specific report name in a command; however, you need to perform this procedure only once to set the data source for all reports.

Procedure

1. Get a list of reports by doing the following actions:
   a. Navigate to the following directory:
      - On AIX, Linux, or Solaris: `/opt/IBM/tivoli/tipv2Components/TCRComponent/bin`
      - On Windows: `C:\ibm\tivoli\tipv2Components\TCRComponent\bin`

   b. Run the `trcmd -list` command:

      ```
      trcmd.{bat|sh} -list –reports -user adminid -password adminpwd
      ```

      where *adminid* is the IBM Tivoli Common Reporting user ID (for example, `tipadmin`) and the *adminpwd* is the password for the IBM Tivoli Common Reporting user ID.
2. Modify the parameters related to JDBC connection as follows:

   ```
   trcmd.{bat|sh} -user adminid -password adminpwd -modify -dataSources
        -reports -reportName "/content/package[@name='IBM Products']
   /folder[@name='Security Access Manager']/report[@name='Administrator and
   Self-Care Password Change History']"
        -setDataSource odaUser=oda_user
         odaPassword=oda_userpwd
   odaURL=jdbc:db2://hostname:Port/eventxml
   ```

   Where:

   adminid is the IBM Tivoli Common Reporting user ID (For example, `tipadmin`).

   adminpwd is the password for the IBM Tivoli Common Reporting user ID.

   hostname is the name of the server where the CARS server is installed and configured.

   oda_user is the JDBC data source user ID (Owner of database. For example, `db2admin`)

   oda_userpwd is the password associated with the JDBC data source user ID.

   Port is the value of port number on which the CARS server is configured. Default value for the port value is 50000.
3. To get detail about a specific report, run the command such as in the following example:

   ```
   trcmd.{bat|sh} -user adminid -password adminpwd -list
   -report "/content/package
   [@name='IBM Products']/folder[@name='Security Access Manager']/report
   [@name='Administrator and Self-Care Password Change History']"
   ```

   where *adminid* is the IBM Tivoli Common Reporting user ID (for example, `tipadmin`) and the *adminpwd* is the password for the IBM Tivoli Common Reporting user ID.
4. To verify successful configuration, you can do either of the following steps:
   - Run the command to list reports and verify that all the properties related to JDBC connection were updated.

- Navigate to the **Reporting** > **Common Reporting** section on UI and run reports for different type of events.

Updating and customizing your Tivoli Common Reporting environment

Because Tivoli Common Reporting is an open-source, application-development environment, updates and technical notes for the program code are posted periodically on the IBM developerWorks® Web site. In addition, you must create custom reports that require additional script development to use the features of Tivoli Common Reporting.

To access product updates and report customization information for Tivoli Common Reporting, go to the following IBM developerWorks Web site:

http://www.ibm.com/developerworks/spaces/tcr

Chapter 10. Running the Security Access Manager operational reports

You can use Tivoli Common Reporting and Business Intelligence Reporting Tools (BIRT) to run Security Access Manager operational reports.

To generate a formatted report, you must first stage a subset of the Common Audit Service event data from the XML data store into reporting tables.

For information about running the Common Audit Service staging utility (staging utility), see Chapter 6, "Running the server utilities," on page 65.

For information about using Tivoli Common Reporting to run reports, see the Tivoli Common Reporting documentation Web site:

> http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.tivoli.tcr.doc_211/ic-home.html

Chapter 11. Report descriptions

This topic describes each Security Access Manager operational report.

Available reports

The following operational report definitions are included in the Security Access Manager report package:

- Administrator and Self-Care Password Change History
- Audit Event History by User
- Audit Event History for Security Servers
- Authorization Event History by Action
- Failed Authentication Event History
- Failed Authorization Events History
- General Administration Event History
- General Audit Event Details
- General Audit Event History
- General Authorization Event History
- Group Administration Event History
- Locked Account History
- Most Active Accessors Report
- Resource Access by Accessor
- Resource Access by Resource
- Server Availability
- User Administration Event History
- User Password Change History

Administrator and Self-Care Password Change History

This report shows the administrator and self-care password change statistics during a specified time period.

Purpose

Use this report to track compliance and investigate unusual account activity. Statistics display the number of times when an administrator changes account passwords versus when a user changes their own account passwords.

Parameters

You can define the following parameters for the report so that you get only the information you need:

Table 4. Parameters for the Administrator and Self-Care Password Change History report

Parameter name	Description	Example
Start date and time	Shows the starting timestamp for the statistics in the report. The report shows statistics that occur after the start date. Use the calendar icon to select the start date.	Click the year, date, and time in the calendar icon.
End date and time	Shows the ending timestamp for the statistics in the report. The report shows statistics that occur before the end date. Use the calendar icon to select the end date.	Click the year, date, and time in the calendar icon.

General statistics

The following statistics are provided with this report:

- Total number of administrator-initiated and self-care password changes over the time period

Charts

The following table shows each chart that is available when you run the Administrator and Self-Care Password Change History report.

Table 5. Charts displayed for the Administrator and Self-Care Password Change History report

Chart name	What it shows
Password change events	Report listing number and percentage of administrator versus self-care password changes.
Self-care and administrator password events	Pie chart that shows number and percentage of administrator versus self-care password changes.

Related reports

The following report tracks password data:

- "User Password Change History" on page 116

Audit Event History by User

This report shows the total number of events for a specified user during a specified time period. It also presents a list of all events of the specified event type that is sorted by time stamp and grouped by session ID during the time period.

Purpose

The purpose of this report is to investigate the activity of a particular user during a specified time period.

Parameters

You can define the following parameters for the report so that you get only the information you need:

Table 6. Parameters for the Audit Event History by User report

Parameter name	Description	Example
Start date and time	Shows the starting timestamp for the statistics in the report. The report shows statistics that occur after the start date. Use the calendar icon to select the start date.	Click the year, date, and time in the calendar icon.
End date and time	Shows the ending timestamp for the statistics in the report. The report shows statistics that occur before the end date. Use the calendar icon to select the end date.	Click the year, date, and time in the calendar icon.
Event type	Select the event type, or select **All**.	To report on the authentication events for a user, select **AUDIT_AUTHN** from the drop-down list.
Product name	Enter or select the product, or select **All** to report on all products for the user.	To report on events for IBM Security Access Manager for Web, select that name from the drop-down list.
User	Enter the user name.	To report on user name, smithcm, enter `smithcm`.
Domain	Enter the domain name.	To report on a user in domainA, enter `domainA`.

General statistics

The following statistics are provided with this report:

- Total number of events for a specified user

Charts

The following table shows each chart that is available when you run the Audit Event History by User report.

Table 7. Charts displayed for the Audit Event History by User report

Chart name	What it shows
Audit events for user	Shows a report that details all events for a specified user. The report is sorted by session ID and time stamp during the time period.

Related reports

The following reports track other audit event data:

- "Audit Event History for Security Servers" on page 98

- "General Audit Event History" on page 105
- "General Audit Event Details Report" on page 104

Audit Event History for Security Servers

This report shows a list of audit events that occurred during the specified time period.

Purpose

The purpose of this report is to investigate the activity of a particular security server during a specified time period.

Parameters

You can define the following parameters for the report so that you get only the information you need:

Table 8. Parameters for the Audit Event History for Security Servers report

Parameter name	Description	Example
Start date and time	Shows the starting timestamp for the statistics in the report. The report shows statistics that occur after the start date. Use the calendar icon to select the start date.	Click the year, date, and time in the calendar icon.
End date and time	Shows the ending timestamp for the statistics in the report. The report shows statistics that occur before the end date. Use the calendar icon to select the end date.	Click the year, date, and time in the calendar icon.
Product name	Select or enter the product name, or select **All** to report on all products.	To report on events for IBM Security Access Manager for Web, select that name from the drop-down list.
Sort by	Select one of the following as the sort criteria: • **Timestamp** • **Servername** • **Action**	To sort the report by server name, select **Servername** from the drop-down list.

General statistics

The following statistics are provided with this report:

- Total number of security server events

Charts

The following table shows each chart that is available when you run the Audit Event History for Security Servers report.

Table 9. Charts displayed for the Audit Event History for Security Servers report

Chart name	What it shows
Number of security server audit events by action	Report showing total number of events for each action.
Audit event history for security servers	Report listing all security server events that are sorted by specified sort criteria and time stamp during the time period.

Related reports

The following reports track audit events:

- "Audit Event History by User" on page 96
- "General Audit Event History" on page 105
- "General Audit Event Details Report" on page 104

Authorization Event History by Action

This report shows a list of all authorization events that contain the specified action. The report is sorted by resource and then time stamp during the time period specified.

Purpose

The purpose of this report is to analyze authorization event history for each action for incident investigation and assure compliance.

Parameters

You can define the following parameters for the report so that you get only the information you need:

Table 10. Parameters for the Authorization Event History by Action report

Parameter name	Description	Example
Start date and time	Shows the starting timestamp for the statistics in the report. The report shows statistics that occur after the start date. Use the calendar icon to select the start date.	Click the year, date, and time in the calendar icon.
End date and time	Shows the ending timestamp for the statistics in the report. The report shows statistics that occur before the end date. Use the calendar icon to select the end date.	Click the year, date, and time in the calendar icon.
Product name	Enter or select the product name, or select **All**.	To report on events for IBM Security Access Manager for Web, select that name from the drop-down list.
Location	Enter the name of the location, or select **All**.	To report on location PS01NY, enter `PS01NY`.

Table 10. Parameters for the Authorization Event History by Action report (continued)

Parameter name	Description	Example
Location class	Select the location class **Source** or **User**.	To report on the location class of user, select **User** from the drop-down list.
Action	Enter the name of an action, or select **All**.	To report on only the create actions, enter **Create**.

General statistics

The following statistics are provided with this report:

- Total number of authorization events

Charts

The following table shows each chart that is available when you run the Authorization Event History by Action report.

Table 11. Charts displayed for the Authorization Event History by Action report

Chart name	What it shows
Authorization audit events list	Shows a report that lists all authorization events that contain the specified action. The report is sorted by resource and then time stamp during the time period.

Notes

- The report can generate a large amount of data. Be sure to limit the data that the report produces by specifying a shorter time frame or a particular action.

Related reports

The following reports track authorization events:

- "Failed Authorization Events History" on page 101
- "General Authorization Event History" on page 106

Failed Authentication Event History

This report shows a list of all failed authentication events over the time period. The report is sorted by specified sort criteria, then time stamp.

Purpose

The purpose of this report is to investigate security incidents. An administrator can track failed authentication events to investigate security attacks.

Parameters

You can define the following parameters for the report so that you get only the information you need:

Table 12. Parameters for the Failed Authentication Events History report

Parameter name	Description	Example
Start date and time	Shows the starting timestamp for the statistics in the report. The report shows statistics that occur after the start date. Use the calendar icon to select the start date.	Click the year, date, and time in the calendar icon.
End date and time	Shows the ending timestamp for the statistics in the report. The report shows statistics that occur before the end date. Use the calendar icon to select the end date.	Click the year, date, and time in the calendar icon.
Product name	Select or enter the product name, or select **All** to report on all products.	To report on events for IBM Security Access Manager for Web, select that name from the drop-down list.
Sort by:	Select one of the following fields to sort by: • **User** • **Reason** • **Timestamp**	To sort the report by time stamp, select **Timestamp** from the drop-down list.

Charts

The following table shows each chart that is available when you run the Failed Authentication Events History report.

Table 13. Charts displayed for the Failed Authentication Events History report

Chart name	What it shows
Failed authentication events	Report detailing each authentication event that failed in the time period, giving the reason for the failure.

Failed Authorization Events History

This report shows all of the failed authorization events during a specified time period.

Purpose

The purpose of this report is to investigate security incidents. An administrator can track failed authorization events to investigate security attacks.

Parameters

You can define the following parameters for the report so that you get only the information you need:

Table 14. Parameters for the Failed Authorization Events History report

Parameter name	Description	Example
Start date and time	Shows the starting timestamp for the statistics in the report. The report shows statistics that occur after the start date. Use the calendar icon to select the start date.	Click the year, date, and time in the calendar icon.
End date and time	Shows the ending timestamp for the statistics in the report. The report shows statistics that occur before the end date. Use the calendar icon to select the end date.	Click the year, date, and time in the calendar icon.
Location	Enter or select the name of the location, or select **All**.	To report on location PS01NY, select **PS01NY** from the drop-down list.
Location class	Select the location class **Source** or **User**.	To report on the location class of user, select **User** from the drop-down list.
Sort by	Select one of the following fields to sort by: • **User** • **Domain** • **Timestamp**	To sort the report by time stamp, select **Timestamp** from the drop-down list.

Charts

The following table shows each chart that is available when you run the Failed Authorization Events History report.

Table 15. Charts displayed for the Failed Authorization Events History report

Chart name	What it shows
Failed authorization events	Report detailing each authorization event that failed in the time period, giving the reason for the failure.

Related reports

The following reports track authorization events:

- "Authorization Event History by Action" on page 99
- "General Authorization Event History" on page 106

General Administration Event History

This report shows the history of general management actions that are done over a specified time interval.

Purpose

The purpose of the report is to track the actions of a user, in general, for completing administrative events. It shows the amount of management activity reported over time. Following are the various types of management activities:

- Administration of a policy database, separate from the user registry
- Administration of users and data in the user registry
- Administration of application configuration and servers
- Administration of resource objects
- Application-level administration of users

Parameters

You can define the following parameters for the report so that you get only the information you need:

Table 16. Parameters for the General Administration Event History report

Parameter name	Description	Example
Start date and time	Shows the starting timestamp for the statistics in the report. The report shows statistics that occur after the start date. Use the calendar icon to select the start date.	Click the year, date, and time in the calendar icon.
End date and time	Shows the ending timestamp for the statistics in the report. The report shows statistics that occur before the end date. Use the calendar icon to select the end date.	Click the year, date, and time in the calendar icon.
Product name	Enter or select the product name, or select **All**.	To report on events for IBM Security Access Manager for Web, select that name from the drop-down list.
Sort by	The primary sort for this report is based on user ID and domain. The secondary sort criteria can be: • **Timestamp** • **Event Type** • **Policy Resource Name** • **Resource Type**	For each user ID and domain, sort on event type by selecting **Event Type** from the drop-down list.

General statistics

The following statistics are provided with this report:

- Total number of administration events

Charts

The following table shows each chart that is available when you run the General Administration Event History report.

Table 17. Charts displayed for the General Administration Event History report

Chart name	What it shows
General administration event list	Report listing event information for each administrator.

Related reports

The following reports track administration events:

- "Administrator and Self-Care Password Change History" on page 95
- "Group Administration Event History" on page 108
- "User Administration Event History" on page 115

General Audit Event Details Report

This report shows all information about a single audit event.

Purpose

The purpose of this report is to provide specific details about a certain audit event. Typically, you will run this report after you run any of the other operational reports and determine that you want all the details of one event.

Parameters

You can define the following parameters for the report so that you get only the information you need:

Table 18. Parameters for the General Audit Event Details report

Parameter name	Description	Example
Event reference ID	Enter the event reference ID. This ID is a sequence number that is generated by the Common Audit Service.	To get the event details for reference ID 8090001234, enter **8090001234**.

Charts

The following table shows each chart that is available when you run the General Audit Event Details report.

Table 19. Charts displayed for the General Audit Event Details report

Chart name	What it shows
Event Details for Record ID *reference_ID*	A list of the elements and values that are associated with the specified audit event.

Notes

- Before you run the General Audit Event Details Report, verify that you installed the Java stored procedure on the DB2 server. See "Deploying the Java stored procedure for an audit details report" on page 54.

Related reports

The following reports track audit events:

- "Audit Event History by User" on page 96
- "Audit Event History for Security Servers" on page 98
- "General Audit Event History" on page 105

General Audit Event History

This report shows general information about audit events during a specified time.

Purpose

The report shows a list of all events of the specified event type. The events are sorted by a specified sort criterion and time stamp during the time period. The data aids in incident investigation and assuring compliance. It presents general statistics that list the total number of audit events for each audit event type.

Parameters

You can define the following parameters for the report so that you get only the information you need:

Table 20. Parameters for the General Audit Event History report

Parameter name	Description	Example
Start date and time	Shows the starting timestamp for the statistics in the report. The report shows statistics that occur after the start date. Use the calendar icon to select the start date.	Click the year, date, and time in the calendar icon.
End date and time	Shows the ending timestamp for the statistics in the report. The report shows statistics that occur before the end date. Use the calendar icon to select the end date.	Click the year, date, and time in the calendar icon.
Number of audit events	Enter or select the number of audit events to display.	To show a maximum of 200 audit events, enter `200`.
Event type	Enter or select the event type, or select **All**.	To report on only the AUDIT_MGMT_KEY event type, select **AUDIT_MGMT_KEY** from the drop-down list.
Product name	Select or enter the product name, or select **All** to report on all products.	To report on events for IBM Security Access Manager for Web, select that name from the drop-down list.
Sort by	Select one of the following as the sort criteria: • **Timestamp** • **Outcome** • **Event Type** • **User** • **Domain**	To sort the report by user, select **User** from the drop-down list.

General statistics

The following statistics are provided with this report:

- Total number of each audit

Charts

The following table shows each chart that is available when you run the General Audit Event History report.

Table 21. Charts displayed for the General Audit Event History report

Chart name	What it shows
Audit events by event type	Report listing each audit event followed by the total number for each type.

Related reports

The following reports track audit events:

- "Audit Event History by User" on page 96
- "Audit Event History for Security Servers" on page 98
- "General Audit Event Details Report" on page 104

General Authorization Event History

This report shows general information about authorization events during a specified time period.

Purpose

The purpose of this report is to analyze authorization event history by using filter criterion like time period, product name, location, and access decision. Use the data for incident investigation and assuring compliance. The report presents general statistics that list the total number of authorization events, failed authorization events, successful authorization events, and unauthenticated events during a time period. Additionally, it shows a list of all authorization events that are sorted by a specified sort criteria.

Parameters

You can define the following parameters for the report so that you get only the information you need:

Table 22. Parameters for the General Authorization Event History report

Parameter name	Description	Example
Start date and time	Shows the starting timestamp for the statistics in the report. The report shows statistics that occur after the start date. Use the calendar icon to select the start date.	Click the year, date, and time in the calendar icon.
End date and time	Shows the ending timestamp for the statistics in the report. The report shows statistics that occur before the end date. Use the calendar icon to select the end date.	Click the year, date, and time in the calendar icon.

Table 22. Parameters for the General Authorization Event History report (continued)

Parameter name	Description	Example
Product name	Select the product, or select **All** to report on all products for the user.	To report on events for IBM Security Access Manager for Web, select that name from the drop-down list.
Location	Enter the name of the location, or select **All**.	To report on location PS01NY, enter `PS01NY`.
Location class	Select the location class **Source** or **User**.	To report on the location class of user, select **User** from the drop-down list.
Resource name	Enter the resource name, or select **All** to show all resources.	To report on only the events for `/etc/passwd`, enter `/etc/passwd`.
Access decision	Select one of the following access decision choices: • **All** • **Permitted** • **Denied**	To report on only the authorizations that were denied, select **Denied** from the drop-down list.
Authentication type	Select one of the following authentication type choices: • **All** • **Authenticated** • **Unauthenticated**	To report on only the unauthenticated events, select **Unauthenticated**.
Number of events to show	Select or enter the maximum number of events to display on the report.	To display 1,000 events, select **1000** from the drop-down list.
Sort by	Select one of the following sort criteria: • **Timestamp** • **Resource** • **User** • **Domain**	To sort the events by domain, select **Domain** from the drop-down list.

General statistics

The following statistics are provided with this report:

- Authorization events by access decision
- Unique users per resource

Charts

The following table shows each chart that is available when you run the General Authorization Event History report.

Table 23. Charts displayed for the General Authorization Event History report

Chart name	What it shows
Authorization Audit Events List	Report listing all authorization events for a specified criteria.

Related reports

The following reports tracks authorization events:

- "Authorization Event History by Action" on page 99
- "Failed Authorization Events History" on page 101

Group Administration Event History

This report shows all of the group administration events over time.

Purpose

The purpose of this report is to investigate security incidents and to track changes to groups by administrators.

Parameters

You can define the following parameters for the report so that you get only the information you need:

Table 24. Parameters for the Group Administration Event History report

Parameter name	Description	Example
Start date and time	Shows the starting timestamp for the statistics in the report. The report shows statistics that occur after the start date. Use the calendar icon to select the start date.	Click the year, date, and time in the calendar icon.
End date and time	Shows the ending timestamp for the statistics in the report. The report shows statistics that occur before the end date. Use the calendar icon to select the end date.	Click the year, date, and time in the calendar icon.
Sort by	Specify whether you want the report to be sorted by timestamp, administrator, target group, or action. The default sort criteria is timestamp.	To show a report that is sorted by action, select action from the drop-down list.

General statistics

The following statistics are provided with this report:

- Total number of group administration events

Charts

The following table shows each chart that is available when you run the Group Administration Event History report.

Table 25. Charts displayed for the Group Administration Event History report

Chart name	What it shows
Group Administration Event List	Report that lists all group administration audit events. The events are sorted by specified sort criteria and time stamp during the time period

Related reports

The following reports track administration events:

- "Group Administration Event History" on page 108

Locked Account History

This report shows the accounts that were locked for a specified time.

Purpose

The purpose of this report is to aid in tracking compliance and showing a pattern of unsuccessful login attempts.

Parameters

You can define the following parameters for the report so that you get only the information you need:

Table 26. Parameters for the Locked Account History report

Parameter name	Description	Example
Start date and time	Shows the starting timestamp for the statistics in the report. The report shows statistics that occur after the start date. Use the calendar icon to select the start date.	Click the year, date, and time in the calendar icon.
End date and time	Shows the ending timestamp for the statistics in the report. The report shows statistics that occur before the end date. Use the calendar icon to select the end date.	Click the year, date, and time in the calendar icon.
Number of users	Select or enter the maximum number of users to display in the report.	To show the first 10 users that were locked out during the time period, select **10** from the drop-down list.

General statistics

The following statistics are provided with this report:

- Total number of locked out accounts

Charts

The following table shows each chart that is available when you run the Locked Account History report.

Table 27. Charts displayed for the Locked Account History report

Chart name	What it shows
Locked Account List	Shows a list of all locked account events. The events are sorted by lockout reason

Most Active Accessors Report

This report shows a list of users who are the most active in the system.

Purpose

The purpose of this report is to lead administrators to investigate improper use of their resources.

Parameters

You can define the following parameters for the report so that you get only the information you need:

Table 28. Parameters for the Most Active Accessors report

Parameter name	Description	Example
Start date and time	Shows the starting timestamp for the statistics in the report. The report shows statistics that occur after the start date. Use the calendar icon to select the start date.	Click the year, date, and time in the calendar icon.
End date and time	Shows the ending timestamp for the statistics in the report. The report shows statistics that occur before the end date. Use the calendar icon to select the end date.	Click the year, date, and time in the calendar icon.
Location	Select the name of the location, or select **All**.	To report on location PS01NY, enter `PS01NY`.
Location class	Select the location class **Source** or **User**.	To report on the location class of user, select **User** from the drop-down list.
Number of users to show	Select or enter the maximum number of users to show on the report.	To display 50 users, select **50** from the drop-down list.

General statistics

The following statistics are provided with this report:

- Total number of authorization and resource access events

Charts

The following table shows each chart that is available when you run the Most Active Accessors report.

Table 29. Charts displayed for the Most Active Accessors report

Chart name	What it shows
Most Active Accessors	Report showing each user, domain, and the total number of events for the user. The names are listed in order according to number of events (highest to lowest).

Related reports

The following reports track accessor data:

- "Resource Access by Accessor"
- "Resource Access by Resource" on page 112

Resource Access by Accessor

This report shows the top 10 resources in terms of access and authorization events during a time period for each machine name identified.

Purpose

The purpose of this report is to enable an administrator to identify who is accessing what resource on a protected machine over time. This information shows trends in user access patterns and serves as a starting point for incident investigation.

Parameters

You can define the following parameters for the report so that you get only the information you need:

Table 30. Parameters for the Resource Access by Accessor report

Parameter name	Description	Example
Start date and time	Shows the starting timestamp for the statistics in the report. The report shows statistics that occur after the start date. Use the calendar icon to select the start date.	Click the year, date, and time in the calendar icon.
End date and time	Shows the ending timestamp for the statistics in the report. The report shows statistics that occur before the end date. Use the calendar icon to select the end date.	Click the year, date, and time in the calendar icon.
Product name	Enter or select the product name, or select **All**.	To report on events for IBM Security Access Manager for Web, select that name from the drop-down list.

Table 30. Parameters for the Resource Access by Accessor report (continued)

Parameter name	Description	Example
Location	Enter the name of the location, or select **All**.	To report on location PS01NY, enter `PS01NY`.
Accessor name	Enter the user ID of the accessor.	If `jjsmith` is the user ID you want to investigate, type `jjsmith`.

General statistics

The following statistics are provided with this report:

- Total number of accesses to a resource for an accessor
- Total number of a user's accesses on a location

Charts

The following table shows each chart that is available when you run the Resource Access by Resource report.

Table 31. Charts displayed for the Resource Access by Resource report

Chart name	What it shows
Top 10 Resources Accesses per Location by *accessor*	List of accessors with the most activity for a location. A total number of accesses for the user is shown.

Related Reports

The following report tracks resource usage:

- "Resource Access by Resource"

Resource Access by Resource

This report shows the top 10 accessors in terms of access and authorization events during a time period for each machine name identified.

Purpose

The purpose of this report is to show the most heavily accessed resources on a Security software protected machine over time. With this report, you can see who is accessing what resources and it can be a starting point for investigating abnormal resource utilization.

Parameters

You can define the following parameters for the report so that you get only the information you need:

Table 32. Parameters for the Resource Access by Resource report

Parameter name	Description	Example
Start date and time	Shows the starting timestamp for the statistics in the report. The report shows statistics that occur after the start date. Use the calendar icon to select the start date.	Click the year, date, and time in the calendar icon.
End date and time	Shows the ending timestamp for the statistics in the report. The report shows statistics that occur before the end date. Use the calendar icon to select the end date.	Click the year, date, and time in the calendar icon.
Product name	Enter or select the product, or select **All** to report on all products for the user.	To report on events for IBM Security Access Manager for Web, select that name from the drop-down list.
Location	Enter the name of the location, or select **All**.	To report on location PS01NY, enter `PS01NY`.
Resource	Enter the resource name.	If `/etc/passwd` is the resource you want to investigate, type `/etc/passwd`.

General statistics

The following statistics are provided with this report:

- Total number of accesses for each resource
- Total number of accesses for each location

Charts

The following table shows each chart that is available when you run the Resource Access by Resource report.

Table 33. Charts displayed for the Resource Access by Resource report

Chart name	What it shows
Top 10 Accessors per Location for *resource*	List of most heavily accessed resources for a location. A total number of accesses is shown.

Related Reports

The following report track resource usage:

- "Resource Access by Accessor" on page 111

Server Availability

This report shows the availability status of Security software servers on a specific machine.

Purpose

The purpose of this report is to show the availability of a Security software server over time. It gives an obvious indication that a server is functioning. You can determine whether a critical server is protected. You can use this report for verification purposes during times of infrastructure audits.

The events that display represent *heartbeat* events. Heartbeat events are events that the software daemons periodically creates to indicate that it is operating.

The data that displays is the total time period a protected machine reports (hours, days, and months), a count of the heartbeat events for a machine that is running Security software. This data is displayed in table form showing pings per time period for each protected machine. You can display all protected machines in the report or limit the report by entering a single host name as the subject of the report.

Parameters

You can define the following parameters for the report so that you get only the information you need:

Table 34. Parameters for the Server Availability report

Parameter name	Description	Example
Start date and time	Shows the starting timestamp for the statistics in the report. The report shows statistics that occur after the start date. Use the calendar icon to select the start date.	Click the year, date, and time in the calendar icon.
End date and time	Shows the ending timestamp for the statistics in the report. The report shows statistics that occur before the end date. Use the calendar icon to select the end date.	Click the year, date, and time in the calendar icon.
Product name	Select or enter the name of the product.	To report on IBM Security Access Manager for Web only, select it from the drop-down list.
Location	Enter the name of the location, or select **All**.	To report on location PS01NY, enter `PS01NY`.
Time increment	Select the time increment for reporting from the following options: • **Hourly** • **Daily** • **Monthly**	To report on a daily basis, select **Daily** from the drop-down list.

Charts

The following table shows each chart that is available when you run the Server Availability report.

Table 35. Charts displayed for the Server Availability report

Chart name	What it shows
Number of heart beats per time period	Report showing the total number of heartbeats for a specified time period for specified products and locations.

Related reports

The following reports track server data:

- "Audit Event History for Security Servers" on page 98

User Administration Event History

This report shows all of the user administration events over time.

Purpose

The purpose of this report is to investigate security incidents and to track changes to users by administrators.

Parameters

You can define the following parameters for the report so that you get only the information you need:

Table 36. Parameters for the User Administration Event History report

Parameter name	Description	Example
Start date and time	Shows the starting timestamp for the statistics in the report. The report shows statistics that occur after the start date. Use the calendar icon to select the start date.	Click the year, date, and time in the calendar icon.
End date and time	Shows the ending timestamp for the statistics in the report. The report shows statistics that occur before the end date. Use the calendar icon to select the end date.	Click the year, date, and time in the calendar icon.
Sort by	Select the sort criteria from the following list: • **Timestamp** • **Administrator** • **Target User** • **Action** (such as, add, delete, modify, add to group, and delete from group)	To sort the report by the administrator ID, select **Administrator** from the drop-down box.

General statistics

The following statistics are provided with this report:

- Total number of user administration events

Charts

The following table shows each chart that is available when you run the User Administration Event History report.

Table 37. Charts displayed for the User Administration Event History report

Chart name	What it shows
User Administration Event List	List of all user administration events that are sorted by specified sort criteria and time stamp during the time period.

Related Reports

The following report track administration issues:

- "General Administration Event History" on page 102
- "Group Administration Event History" on page 108
- "User Administration Event History" on page 115

User Password Change History

This report shows the user password change statistics during a specified time period.

Purpose

The purpose of this report is to aid in tracking compliance and investigating unusual account activity. From this report, you can gather the following details:

- Percentage of successful password attempts
- Percentage of unsuccessful password attempts
- Reasons for failed password attempts
- User IDs where the password changes occurred
- User domains where the password changes occurred

Parameters

You can define the following parameters for the report so that you get only the information you need:

Table 38. Parameters for the User Password Change History report

Parameter name	Description	Example
Start date and time	Shows the starting timestamp for the statistics in the report. The report shows statistics that occur after the start date. Use the calendar icon to select the start date.	Click the year, date, and time in the calendar icon.

Table 38. Parameters for the User Password Change History report (continued)

Parameter name	Description	Example
End date and time	Shows the ending timestamp for the statistics in the report. The report shows statistics that occur before the end date. Use the calendar icon to select the end date.	Click the year, date, and time in the calendar icon.

General statistics

The following statistics are provided with this report:

- Total number of password change events
- Total number of unsuccessful password change events
- Password change events that are categorized by successful and unsuccessful
- Failed password changed events that are categorized by reason

Charts

The following table shows each chart that is available when you run the User Password Change History report.

Table 39. Charts displayed for the User Password Change History report

Chart name	What it shows
Password change events	Pie chart that shows percentage of successful versus unsuccessful attempts.
Failed password change events	Pie chart that shows percentages of failed password change event reasons.
User password change history	Report detailing each time that a user changed or attempted to change their password

Related reports

The following report tracks password data:

- “Administrator and Self-Care Password Change History” on page 95

Chapter 12. Reporting scenarios

The following detailed report scenarios describes a situation and suggests a report that would give you the information that you need.

Roles

There are four types of roles that are defined for auditing and reporting of security events in these scenarios.

Table 40. Roles for auditing and reporting

Title	Name that are used in scenarios	Goals
Chief Security Officer	Robert	• Provide the best security solution at the least cost to the company. • Make the computer systems secure. • Ensure that all security audits are successfully passed.
System Administrator	James	• Ensure that applications and systems he manages are always available and running smoothly.
Security Auditor	Christine	• Make sure the CEO and Board trust the security and confidentiality of their systems.
Application platform owner	Miguel	• Make sure the CEO and Board trust the security and confidentiality of their systems. • Maintain a stable environment for applications that stay running 24 hours a day.

Incident investigation scenario

Which user is logging in the system between 3 AM and 4 AM on Wednesdays?

Scenario description

This scenario involves an administrator who is concerned about the number of after-hour logins. Following is the flow of an example situation:

1. Someone on Robert's staff notices that there is an abnormal number of after-hour logins between 3 AM and 4 AM.
2. Robert calls Miguel and ask Miguel to investigate who is logging in at that time of night.
3. Miguel uses BIRT to run a report that shows all the users who logged in between 3 AM and 4 AM.
4. Miguel runs the same report after he restores previously archived events and publishes those reports to Tivoli Common Reporting so that someone on Robert's staff can look at them and determine the next steps to take.

Report to use

- General Audit Event History

Parameters to use

Start date and time
01/09/12 12:00:00 AM

End date and time
03/19/12 12:00:00 AM

Number of audit events
200

Event type
AUDIT_AUTHN

Product name
All

Sort by
Timestamp

How to use the report

Scan the report from the beginning. Look for the 3:00 AM to 4:00 AM time frame for each day. Note the users and the events.

Resource access compliance scenario

Do I have the reports that I need to pass the next audit?

Scenario description

This scenario involves running a compliance report on a monthly basis to prepare for future audits. Following is the flow of an example situation:

1. The company is required to keep records of all accesses to a sensitive application. Robert wants to make sure that this data is on hand in case he is audited.
2. Robert runs a report once a month to show all accesses to the specified application. Robert prints that report and files it away for safekeeping.

Report to use

- General Authorization Event History

Parameters to use

Start date and time
02/01/12 12:00:00 AM

End date and time
03/01/12 12:00:00 AM

Product name
IBM Security Access Manager for Web

Location
PS0760

Location class
Source

Resource name
All

Access decision
All

Authenticated type
All

Number of events to show
1000

Sort by
Timestamp

How to use the report

Scan the report to make sure that the correct data was used and file the report for future audits. Create a report for each sensitive application.

Login policy compliance scenario

How effective is the new login policy?

Scenario description

This scenario involves running a report that captures the number of locked-out accounts. Following is the flow of an example situation:

1. Robert wants to see how the new login policy is affecting users. The new login policy states that when a user attempts to log in more than three times with a password that is not valid, that account is locked out.
2. Robert asks someone on his staff to run a nightly report (each night for six months) that shows how many account lockout events occurred each night.
3. In the nightly reports that generated during this six-month period, Robert notices that when the new login policy was enacted, there were many locked out account events. Over time, the number of locked out account events decreased. Robert assumes that the policy is effective and that users are remembering their passwords.

Report to use

- Locked Account History

Parameters to use

Start date and time (for first nightly report)
02/01/12 12:00:00 AM

End date and time (for first nightly report)
02/02/12 12:00:00 AM

Number of users
100

How to use the report

The report displays the list of accounts that were locked out in date sequence.

Server availability scenario

Was the Security Access Manager policy server available yesterday?

Scenario description

This scenario involves running a report to show the availability of a server over time. The following steps show the flow of an example situation:

1. The Security Access Manager policy server was recently installed on a new machine.
2. James wants to be sure that the policy server is up and operating as expected and runs a report to show the activity for this server.
3. James reviews the report to determine if the activity for this server is normal and operating on target.

Report to use

- Server Availability Report

Parameters to use

Begin time
04/01/12 12:00:00 AM

End time
04/02/12 12:00:00 AM

Product name
All

Product name
IBM Security Access Manager for Web

Location
PS0555

Time increment
Hourly

How to use the report

Note the heartbeat count for each hour the policy server was running to see whether the server was operating normally.

Chapter 13. Creating custom reports

You can create custom reports to meet the needs of your organization. Common Auditing and Reporting Service and Tivoli Common Reporting provide instructions and utilities to help you develop custom reports. To develop a custom report for event data that is captured by Common Auditing and Reporting Service, use the following two procedures:

Define new staging tables
Staging tables are used by Common Auditing and Reporting Service to create reports. If the existing staging tables in the Service do not contain the event information that you need, set up your own staging tables. If you need instructions to do so, see "Creating custom report tables using Common Audit Service".

Create a Tivoli Common Reporting report package
To create a report package, use the BIRT report designer and Tivoli Common Reporting to:

- Design the format and parameters of the report
- Create the scripts that are needed to extract the data from the staging tables
- View the report data

See “Creating custom reports using Tivoli Common Reporting” on page 133 for more information about generating custom reports.

Creating custom report tables using Common Audit Service

Use Common Auditing Service to create custom report tables that can be queried by a reporting utility, for example, Tivoli Common Reporting, to design and generate custom reports.

You can generate custom reports from predefined reporting tables and you can create new custom tables. To generate reports that meet the needs of your organization, first identify the event types and specific elements of each event type that provide the wanted information. Next, examine the predefined staging tables to determine whether the captured event types and attributes are sufficient to meet your needs. If you do not need to stage more data, you can use your reporting utility to query the predefined reporting tables to produce the wanted reports.

The Common Auditing Service staging utility (staging utility) must first stage a specified subset of data into reporting tables to generate formatted reports. The staging utility queries a configuration file, `CARSShredder.conf`, to determine exactly what data to stage. If you need to stage more data that is not provided in the predefined staging tables, then you must customize the `CARSShredder.conf` file and create new reporting tables.

The Common Auditing Service provides a procedure for defining the subset of data that is included in the reporting tables. Custom reports can then be created to analyze the subset of data that is staged into the reporting tables.

To develop a custom report, your organization first identifies the event types and the specific elements of each of these event types that provides the required data.

Next, specify the events, event elements, and corresponding staging table names and columns in the `CARSShredder.conf` configuration file. You then run the Common Auditing Service staging utility against the `CARSShredder.conf` configuration file to stage the event data into the reporting tables from the live database tables that contain the captured event data.

To set up the `CARSShredder.conf` configuration file, you must back up and then replace the default version of the file with a new, custom version that you build by using the `CARSShredder.conf.custom.template`. You do not modify existing default tables to create custom tables; you create new, more reporting tables to hold data for custom reports. The staging utility stages custom data into these newly defined tables.

The following sections describe the concepts and procedures necessary to stage data for custom reports:

- "Requirements for creating new reporting tables"
- "Working with the CARSShredder.conf configuration file" on page 125
- "Steps to support custom reports" on page 130
- "Creating an example custom report" on page 131

Requirements for creating new reporting tables

The audit server installation creates a set of default reporting tables, which an exploiting application, such as Security Access Manager, can query to generate operational reports.

The tables are organized into one primary table, cars_t_event, and secondary tables that correlate to different event types (for example, cars_t_authz for IBM_CBA_AUDIT_AUTHZ events).

Information that is common to every event type is stored in cars_t_event, with cars_seq_number as the primary key. Event type-specific attributes are stored in the secondary tables, which are linked to the corresponding events in the main table using cars_seq_number.

To create a custom report that meets the needs of your organization, you might need to store a custom subset of the audit event data. To store a custom subset of data, create new secondary tables to hold this data. Columns in the custom tables correspond to attributes (elements) of the audit events. *Do not alter the primary table, cars_t_event, because staging utility functionality is closely tied to this table*.

The only two requirements for creating custom secondary tables are:

- Each secondary table must have a column named cars_seq_number which is defined as a foreign key referring to cars_seq_number in cars_t_event.
- The cars_seq_number column must include the rule 'ON DELETE CASCADE'.

The second requirement is necessary if you plan to use the staging utility to prune the reporting tables. The staging utility prunes events from the cars_t_event table and use DB2 to delete the corresponding events from the secondary tables.

Before you begin the process of generating custom reports, examine the mappings between the attributes (elements) in an event and the target reporting table columns. The following sections describe these data relationships in detail.

Working with the CARSShredder.conf configuration file

This topic describes how to work with the `CARSShredder.conf` configuration file.

Purpose of the CARSShredder.conf file

The Common Auditing Service staging utility references the `CARSShredder.conf` file to determine what data to move into the reporting tables. `CARSShredder.conf` specifies the mapping between the attributes of an event in the XML data store, which is in Common Base Event XML format, and the reporting table columns. In the `CARSShredder.conf` file, you first specify a list of event types that are recognized by the staging utility. Then for each event type, you specify what attributes must be staged into which reporting table column.

The `CARSShredder.conf` file must have two parts:

Event section descriptors
This part lists all event types that are processed by the staging utility.

Event stanzas
This part has stanzas for each of the declared event types. A stanza defines the mapping of event attributes to reporting table columns.

Location of the CARSShredder.conf file

The file name of the XML shredder configuration file is `*CARS_HOME*/server/etc/CARSShredder.conf`, where *CARS_HOME* is the installation directory of Common Auditing Service.

Format and contents of the CARSShredder.conf file

In the `CARSShredder.conf` file, first specify the version, which enables the staging utility to support keyword mapping. Specify version 2, as follows, to enable the use of the *key_xpath_map_file* mapping file.

```
CONFIGURATION_VERSION=2.0
```

Next, specify the list of event types to stage into reporting tables by the staging utility. You can either list all of the event descriptors together at the top of the file (as shown in the following **Event section descriptors** section) or you can list each event type right before you specify the mapping of the attributes.

Next, for each event type, specify the mapping of the attributes, which is used to determine which attributes are staged into the reporting tables (shown in Table 41 on page 127).

Event section descriptors

event_type, version, section, key_xpath_map_file

event_type
: Specifies the name of the event type.

version
: Specifies the version of the Common Base Event model that is used to represent the event type.

section
: Specifies the identifier of the section that contains the mappings between attributes of the declared event type and the corresponding reporting table and column names. Each event name that is specified in Event Section Descriptors must have a corresponding stanza in the configuration file.

key_xpath_map_file
: Specifies the mapping properties file used to correlate keyword values with XPath locator strings. The staging utility searches for the file in the *CARS_HOME*/server/etc/shredderxpaths directory. A default set of keyword-XPath properties files is installed in the *CARS_HOME*/server/etc/shredderxpaths directory, such as ibm_cba_audit_authn_mapping_xpath.properties.

The following example of an event descriptor section contains all audit event types:

```
;          Event Section Descriptors
IBM_CBA_AUDIT_AUTHZ,             1.0.1, [authz], ibm_cba_audit_authz
IBM_CBA_AUDIT_AUTHN,             1.0.1, [authn], ibm_cba_audit_authn
IBM_CBA_AUDIT_MGMT_POLICY,       1.0.1, [mgmt_policy], ibm_cba_audit_mgmt_policy
IBM_CBA_AUDIT_MGMT_REGISTRY,     1.0.1, [mgmt_registry], ibm_cba_audit_mgmt_registry
IBM_CBA_AUDIT_RUNTIME,           1.0.1, [rtime], ibm_cba_audit_rtime
IBM_CBA_AUDIT_RUNTIME_KEY,       1.0.1, [rtime_key], ibm_cba_audit_runtime_key
IBM_CBA_AUDIT_MGMT_CONFIG,       1.0.1, [mgmt_config], ibm_cba_audit_mgmt_config
IBM_CBA_AUDIT_MGMT_PROVISIONING, 1.0.1, [mgmt_provisioning], ibm_cba_audit_mgmt_provisioning
IBM_CBA_AUDIT_COMPLIANCE,        1.0.1, [compliance], ibm_cba_audit_compliance
IBM_CBA_AUDIT_RESOURCE_ACCESS,   1.0.1, [resource_access], ibm_cba_audit_resource_access
IBM_CBA_AUDIT_MGMT_RESOURCE,     1.0.1, [mgmt_resource], ibm_cba_audit_mgmt_resource
;
;The following event types do not have event specific tables.
;
IBM_CBA_AUDIT_AUTHN_TERMINATE,  1.0.1, [authn_terminate], ibm_cba_audit_authn_terminate
IBM_CBA_AUDIT_AUTHN_MAPPING,     1.0.1, [authn_mapping], ibm_cba_audit_authn_mapping
IBM_CBA_AUDIT_AUTHN_CREDS_MODIFY  1.0.1, [authn_creds_modify], ibm_cba_audit_authn_creds_modify
IBM_CBA_AUDIT_DATA_SYNC,     1.0.1, [data_sync], ibm_cba_audit_data_sync
IBM_CBA_AUDIT_WORKFLOW,      1.0.1, [workflow], ibm_cba_audit_workflow
IBM_CBA_AUDIT_PASSWORD_CHANGE,  1.0.1, [password_change], ibm_cba_audit_password_change
;                                                          ;
;The following event types are generated using the Common Audit
;Service Security Event Factory
;
IBM_SECURITY_AUTHN,             1.0.1, [security_authn], ibm_security_authn
IBM_SECURITY_MGMT_POLICY,       1.0.1, [security_mgmt_policy], ibm_security_mgmt_policy
IBM_SECURITY_AUTHZ,             1.0.1, [security_authz], ibm_security_authz
IBM_SECURITY_RUNTIME,           1.0.1, [security_rtime], ibm_security_rtime
IBM_SECURITY_COMPLIANCE,        1.0.1, [security_compliance], ibm_security_compliance
IBM_SECURITY_MGMT_CONFIG,       1.0.1, [security_mgmt_config], ibm_security_mgmt_config
IBM_SECURITY_MGMT_PROVISIONING, 1.0.1, [security_mgmt_provisioning], ibm_security_mgmt_provisioning
IBM_SECURITY_MGMT_REGISTRY,     1.0.1, [security_mgmt_registry], ibm_mgmt_registry
IBM_SECURITY_MGMT_RESOURCE,     1.0.1, [security_mgmt_resource], ibm_mgmt_resource
IBM_SECURITY_RESOURCE_ACCESS,   1.0.1, [security_resource_access], ibm_resource_access

;The following event types do not have event specific tables.

IBM_SECURITY_AUTHN_CREDS_MODIFY, 1.0.1, [security_authn_creds_modify], ibm_security_authn_creds_modify
IBM_SECURITY_AUTHN_TERMINATE,    1.0.1, [security_authn_terminate], ibm_security_authn_terminate
IBM_SECURITY_ENCRYPTION,         1.0.1, [security_encryption], ibm_security_encryption
IBM_SECURITY_FEDERATION,         1.0.1, [security_federation], ibm_security_federation
IBM_SECURITY_SIGNING,            1.0.1, [security_signing], ibm_security_signing
IBM_SECURITY_TRUST,              1.0.1, [security_trust], ibm_security_trust
IBM_SECURITY_WORKFLOW,           1.0.1, [security_workflow], ibm_security_workflow
IBM_SECURITY_AUTHN_DELEGATION,   1.0.1, [security_authn_delegation], ibm_security_authn_delegation
IBM_SECURITY_AUTHN_MAPPING,      1.0.1, [security_authn_mapping], ibm_security_authn_mapping
IBM_SECURITY_DATA_SYNC,          1.0.1, [security_data_sync], ibm_security_data_sync
IBM_SECURITY_MGMT_AUDIT,         1.0.1, [security_mgmt_audit], ibm_security_mgmt_audit
IBM_SECURITY_MGMT_KEY,           1.0.1, [security_mgmt_key], ibm_security_mgmt_key
IBM_SECURITY_SELFCARE,           1.0.1, [security_selfcare], ibm_security_selfcare
IBM_SECURITY_ATTACK,             1.0.1, [security_attack], ibm_security_attack
```

Table 41. Event stanza format of the XML shredder configuration file

Event stanzas	
table, *column*, [*XPath* | *constant* | *keyword* | *#keyword#* | *#keyword*:[*arrayindex*][*arrayindex*]#]	
table	Specifies the database table name.
column	Specifies the column in the database table.
XPath locator string	Describes in XPath format notation the location of an event attribute within an event. XPath locator strings corresponding to event attributes are provided in a set of default properties files that are installed in the `CARS_HOME/server/etc/shredderxpaths` directory. To stage an attribute of an event, you can locate the attribute in the XPath file for that event, and specify either the corresponding keyword or the XPath locator string.
constant	Specifies a constant value to place in a column for all events. This value can be: • A string, such as 'AUDIT_AUTHZ' or an integer or other constant. • Any valid clause that DB2 allows in an SQL INSERT STATEMENT, such as CURRENT TIMESTAMP, CURRENT DATE.
keyword	Specifies one of the following keywords: • #RECORD_ID • #VERSION • #GLOBAL_ID • #CREATION_TIME_UTC The staging utility recognizes these keywords and stages them directly from the XML data store tables without shredding the XML event.
#keyword#	The staging utility searches for the keyword value in `CARS_HOME/server/etc/shredderxpaths/key_xpath_map_file.properties`. For example, if the staging utility processes the IBM_SECURITY_ENCRYPTION event, it searches for the keyword value in the `ibm_security_encryption.properties` file in the `CARS_HOME/server/etc/shredderxpaths` directory. If the properties file cannot be located or opened, or if the keyword cannot be in the properties file, the staging utility returns an error and stops processing. In an array of attributes, the staging utility stages the first element in the array. If multiple arrays exist, such as an array of `userInfo` elements, the staging utility stages the first array attribute in the first `userInfo`. For example, specifying `#userInfo.attribute.name#` as the keyword is equivalent to specifying the following XPath expression: `CommonBaseEvent/extendedDataElements [@name='userInfoList']/children [@name='userInfo'][1]/ children[@name='attributes'] /children[@name='attribute'] [1]/children[@name='name']/values`

Table 41. Event stanza format of the XML shredder configuration file (continued)

Event stanzas	

#*keyword*:[*arrayindex*]
[*arrayindex*]#
	To stage a specific element of an array, such as `attributes`, use this format. For example, specifying `#userInfo.attribute.name:[3][2]#` is equivalent to specifying the following XPath expression:
	`CommonBaseEvent/extendedDataElements[@name='userInfoList']`
	`/children[@name='userInfo'][3]/`
	`children[@name='attributes']`
	`/children[@name='attribute'][2]/children[@name='name']`
	`/values`
	To stage the value of the name element from the second attribute of the third userInfoList, specify `#userInfo.attribute.name:[3][2]#`.
	In XPath expressions, the first element of an array starts with an index value of 1 instead of 0.
	If the number of array indexes that is specified in the keyword does not match the number of arrays in the XPath locator string in the mapping file, the staging utility returns an error and stops processing.

Table 41. Event stanza format of the XML shredder configuration file (continued)

Event stanzas

Following is an example of a stanza that stages authorization event type data into a cars_t_event and a cars_t_authz table. The cars_t_event table is the primary table and cars_t_authz is the secondary table. The integrity of the references between the primary and secondary table is enforced by defining constraints at the time of table creation.

```
[security-authn]
cars_t_event, event_id,    #GLOBAL_ID
cars_t_event, cars_seq_number, #RECORD_ID
cars_t_event, time_stamp,   #creationTime#
cars_t_event, eventType,    "'AUDIT_AUTHN'"
cars_t_event, src_location,  #sourceComponentId.location#
cars_t_event, app_usr_name,  #userInfo.appUserName#

cars_t_cauthn, cars_seq_number,  #RECORD_ID
```

Following is a description of each line of the previous stanza example:

`[security-authn]`
Identifies the stanza name, which must be declared in the Event Descriptor section of the file. Each stanza name must be unique within the configuration file.

`cars_t_event, event_id, #GLOBAL_ID`
Instructs the staging utility to read the event `globalInstanceId` and populate the `EVENT_ID` column with it.

`cars_t_event, cars_seq_number, #RECORD_ID`
Maps the event `record_id` field to the `cars_seq_number` column.

`cars_t_event, time_stamp, #creationTime#`
Maps the event `creationTime` to the `time_stamp` column. The `#creationTime#` keyword maps to the `CommonBaseEvent/@creationTime` XPath in the specified *`key_xpath_map_file`*`.properties` file.

`cars_t_event, eventType, "'AUDIT_AUTHN'"`
Instructs that for every event of type `IBM_CBA_AUDIT_AUTHN`, store the constant “'AUDIT_AUTHN'” in the `eventType` column.

`cars_t_event, src_location, #sourceComponentId.location#`
Instructs the staging utility to select the value of the location attribute from the event and store it in the src_location column. The `#sourceComponentId.location#` keyword maps to the `CommonBaseEvent/sourceComponentId/@location` XPath locator string in the specified *`key_xpath_map_file`*`.properties` file. The XPath expression resolves to the value of the location attribute in the `sourceComponentId` element, whose root element is `CommonBaseEvent`.

`cars_t_event, app_usr_name, #userInfo.appUserName#`
Instructs the staging utility to select the value of the `userInfo` and `appUserName` attributes, whose parent element is `userInfoList`, and stage it into the column `app_usr_name`. The `#sourceComponentId.location#` keyword maps to the CommonBaseEvent/extendedDataElements[@name='userInfoList'] /children[@name='userInfo']/children[@name='appUserName']/values XPath locator string in the specified *`key_xpath_map_file`*`.properties` file.

`cars_t_cauthn, cars_seq_number, #RECORD_ID`
Identifies `cars_t_cauthn` as the target table. `cars_t_cauthn` is the secondary table for the `IBM_SECURITY_AUTHN` event type. This triplet maps the `record_id` field to `cars_seq_number`. All secondary tables must contain this mapping.

Requirements for using the CARSShredder.conf file

Following are the requirements for using the XML shredder configuration (`CARSShredder.conf`) file:

- The active configuration file name must be named CARSShredder.conf.
- The following restrictions apply to the event stanzas in the file:
  - Each event stanza must contain the following three triplets:

    ```
    cars_t_event, event_id,         #GLOBAL_ID
    cars_t_event, cars_seq_number,  #RECORD_ID
    cars_t_event, time_stamp,       CommonBaseEvent/@creationTime
    ```
  - You must specify the primary table first in each section before you specify one or more secondary tables.
  - You must specify the mappings for the primary table, cars_t_event, first in each stanza, before you specify one or more secondary table mappings.
  - You must map the cars_seq_number column to #RECORD_ID in all custom secondary tables, as in the cars_t_event table.
- Specify strings within double quotation marks. For example, "CURRENT TIMESTAMP".
- Use a semicolon (;) to denote comments.
- Nest keywords that correspond to XPath statements in number signs, for example, #action#. The ending number sign helps to differentiate the keyword from the reserved keywords, such as #GLOBAL_ID.
- You cannot specify a column name of a target table multiple times. The following shredder file is incorrect and results in runtime error:

  ```
  cars_t_event, src_comp,      #sourceComponentId.component#
  cars_t_event, src_comp,      #sourceComponentId.subComponent#
  ```

Using the CARSShredder.conf.custom.template

When you install the Common Auditing Service audit server, a CARSShredder.conf.custom.template file is installed in the *CARS_HOME*/server/template directory. The CARSShredder.conf.custom.template configuration file specifies only the minimal attributes that need to be staged for each of the event types, so that the staging utility can function correctly in incremental, historical, and pruning modes. Use a copy of this file as a starting point, and create more mappings to custom secondary tables to satisfy your custom reporting needs.

Note: The CARSShredder.conf file does not explicitly identify the XML data source table name. The Common Auditing Service staging utility automatically determines the source table. Also, the data type of the target column is not identified. The staging utility, during the initialization phase, determines the column type by inspecting the target tables. It is your responsibility to ensure that the attribute selected from the event or the source table is appropriately matched to the targeted table column. If a type mismatch occurs, the staging utility attempts to convert from the source format to the target format.

Steps to support custom reports

You must complete the following tasks to support custom reports:

Procedure

1. Create a data definition language (DDL) file that creates custom secondary reporting tables in the XML data store database.
2. Run the DDL file to create the reporting tables by running the following commands:
   a. `db2 connect to` *database_name* `user` *db2username* `using` *db2password*
   b. `db2 -tsf` *custom.ddl*

3. Save a copy of the default CARSShredder.conf file that is shipped with Common Auditing Service as CARSShredder.conf.default so that you can restore it if needed. Replace this file with your own version to generate custom reports. The default file is in the *CARS_HOME*/server/etc directory.
4. Create a copy of the CARSShredder.conf.custom.template file in the *CARS_HOME*/server/template directory. Rename the copied file to CARSShredder.conf, and place it in the *CARS_HOME*/server/etc/ directory.
5. Edit CARSShredder.conf to stage any additional event attributes needed for your custom reports.
6. Run the staging utility to stage data into the reporting tables by using the modified CARSShredder.conf file.
7. To test and debug the output:
   a. Run the staging utility in historical mode to test a small amount of data.
   b. Verify that event attributes are correctly staged with SQL language.
   c. Generate a custom report to verify that the wanted data is included.

Creating an example custom report

You might create an example custom report to list resources that were accessed and to identify whether access was permitted or denied. In the denial report, you can view the reason for denial.

Procedure

1. Create a custom.ddl file that is used to create a custom secondary table.

   The custom.ddl file contains the following entries:

```
create table cars_t_cauthz
(
    cars_seq_number      BIGINT,
    res_name_in_app      VARCHAR(1024),
    res_name_in_plcy     VARCHAR(1024),
    res_type             VARCHAR(1024),
    access_dcn           VARCHAR(1024),
    access_dcn_rsn       VARCHAR(1024),
    action               VARCHAR(1024),
    usr_attr_name        VARCHAR(1024),
    usr_attr_value       VARCHAR(1024),
--
    foreign key (cars_seq_number) references cars_t_event
      on delete cascade
) in cars_ts_16k;
```

2. Run the DDL file to create the cars_t_cauthz custom table.
3. Save the default CARSShredder.conf file as the CARSShredder.conf.default file.
4. Copy the *CARS_HOME*/server/template/CARSShredder.conf.custom.template file to the *CARS_HOME*/server/etc/CARSShredder.conf file.
5. Edit the CARSShredder.conf file by updating the [authz] stanza of the file to include the following triplets:

```
cars_t_cauthz, cars_seq_number, #RECORD_ID
cars_t_cauthz, res_name_in_app, #resourceInfo.nameInApp#
cars_t_cauthz, res_name_in_plcy, #resourceInfo.nameInPolicy#
cars_t_cauthz, res_type, #resourceInfo.type
cars_t_cauthz, access_dcn, #accessDecision#
cars_t_cauthz, access_dcn_rsn, #accessDecisionReason#
cars_t_cauthz, action, #action#
cars_t_cauthz, usr_attr_name, CommonBaseEvent/
     extendedDataElements[@name='userInfo']/
     children[@name='attributes']/children[@name='name']/
     values[contains(.,'attrname')]
```

```
cars_t_cauthz, usr_attr_value, CommonBaseEvent/
     extendedDataElements [@name='attributes']/
     children[@name='name']/values[contains(.,'attrname')]/
     ../../children[@name='value']/values
```

In this example, the last two entries show the XPath expression instead of keywords. Although these XPath expressions display in this guide as having multiple lines, they must be a single line in your file.

The XPath expression instructs the staging utility to stage when a matching condition is found. The following expression instructs the staging utility to select an attribute that contains the string "`attrname`" in the element name. The last entry instructs the staging utility to stage an element named "`value`," which corresponds to the element name that contains the string "`attrname`."

```
CommonBaseEvent/ extendedDataElements[@name='userInfo']/
children[@name='attributes']/children[@name='name']/
values[contains(.,'attrname')]
```

6. Run the staging utility in incremental mode. See “Running the staging utility command” on page 66.
7. Use the reporting tool in your environment, for example, Tivoli Common Reporting, to generate a custom report using the data from the custom tables.

Creating a custom security event details report

This topic describes how to create a custom security event details report.

The Common Auditing Service provides Java stored procedures that provide details of a security event. The Java stored procedures read the data directly from the XML data store and uncompress the data, if necessary, before the data is returned for a single specified security event as one XML document.

The Java stored procedure accepts the record ID of the security event as a parameter, and then searches the active, inactive, and restore sets of tables. After you establish a connection with the database, use the following SQL commands to access the security event data with the reporting tool of your choice

To generate a custom drill-down report with name-value formatting, use the SQL command to call the IBMCARS_EVENT_DETAIL Java stored procedure:

```
db2 "call IBMCARS_EVENT_DETAIL(record_id,'format')"
```

record_id
: Specifies the record identifier of the security event whose details are required.

'*format*'
: Specifies the format of the output. The following valid values are not case-sensitive:

  MAP
  : Display the security event details as name-value pairs.

  XML
  : Do not apply special formatting to the data. Specifying "XML" is the equivalent of calling the IBMCARS_DD_REPORT Java stored procedure.

If the specified *record_id* exists in the XML data store, the record ID and the associated security event details in the specified *format* are returned.

To generate a custom drill-down report without security event details in name-value pair formatting, use the SQL command to call the IBMCARS_DD_REPORT Java stored procedure:

```
db2 "call IBMCARS_DD_REPORT(record_id)"
```

where *record_id* is the record identifier of the security event whose details are required. If the specified *record_id* exists in the XML data store, the record ID and the associated security event details are returned in XML format.

Creating operational reports from archived data

This topic describes the general procedure to create operational reports from archived data.

Run the store utility with the prearchive operation to record the starting date and the ending date of the archive data.

Use a third-party database archiving tool to archive the data, and to restore the archived data into the cei_t_xmlre and cei_t_xmlxre data restore tables.

To stage restored data into reporting tables, run the staging utility in historical mode. Ensure that the starting and ending dates that you specify are appropriate for the restored data. Events that are not already staged will be staged to the reporting tables.

As with other staged data, generate reports either on demand or by using scheduling. For more information about:

- Archiving data, see “Archiving audit data” on page 169.
- Restoring archived data, see “Restoring audit data” on page 170.
- Running the staging utility, see “Running the staging utility command” on page 66.

Creating custom reports using Tivoli Common Reporting

After you create custom staging tables, use the Cognos® Report Studio (available in Tivoli Common Reporting) to create a report package for the custom tables. You can then import the new report package into Tivoli Common Reporting and begin managing the new reports.

See "Creating reports and report packages" in the *Tivoli Common Reporting Development and Style Guide* for a complete description of:

- Concepts about the Tivoli Common Reporting report package.
- Steps that are used to create a Tivoli Common Reporting report package.

After you create the report package, import the package into Tivoli Common Reporting. To do so, use the instructions in "Importing the Security Access Manager reporting package" in the *Tivoli Common Reporting Development and Style Guide*.

Note: The name of the new reporting package compressed file must differ from the name of the default Security Access Manager reporting package.

For more information and more resources on using Tivoli Common Reporting to create custom report packages, go to the following IBM developerWorks website:

> http://www.ibm.com/developerworks/spaces/tcr

Part 4. Common Auditing Service auditing

Chapter 14. Audit events

When you use the Common Auditing Service, you must configure the server-specific Common Auditing Service client to record specific audit events.

All audit events can contain the following information:

- The machine name and application that generates the event. The machine name is recorded by using the location element. The application name is recorded by using the component and subcomponent elements.
- The time when the event was generated. The time is recorded by using the timestamp (creationTime), startTime, and endTime elements.
- The type of event. The event type is recorded by using the eventType element.
- The user who triggered the event. The user information is recorded by using the appUserName, domain, location, locationType, and sessionId elements. The sessionId element is a unique identifier that is associated with the session of which the event was part. This unique ID (sometimes called a *session index*) can be used to correlate a particular resource access event with an earlier authentication event.
- The outcome of the operation. The outcome is recorded by using the result and failureReason elements. The outcome distinguishes successful and unsuccessful operations. For example, most attempts to authenticate a user are successful, because the user provides valid authentication data. However, when an authentication is unsuccessful, the origin of the failure might be that someone is attempting to impersonate another user.

In addition to this information, each event type might record more information about the event. For more information about the elements that can be recorded in an audit event, see Part 6, “Audit events,” on page 223.

Type of audit events

The configuration of the server determines which audit events are recorded. Security Access Manager uses the following Common Auditing Service events:

- AUDIT_AUTHN
- AUDIT_AUTHN_CREDS_MODIFY
- AUDIT_AUTHN_TERMINATE
- AUDIT_AUTHZ
- AUDIT_MGMT_CONFIG
- AUDIT_MGMT_POLICY
- AUDIT_MGMT_REGISTRY
- AUDIT_MGMT_RESOURCE
- AUDIT_PASSWORD_CHANGE
- AUDIT_RESOURCE_ACCESS
- AUDIT_RUNTIME

Common Auditing and Reporting Service provides the following events. However, Security Access Manager does not take advantage of them:

- AUDIT_AUTHN_MAPPING
- AUDIT_COMPLIANCE
- AUDIT_DATA_SYNC
- AUDIT_MGMT_PROVISIONING

- AUDIT_RUNTIME_KEY
- AUDIT_WORKFLOW

AUDIT_AUTHN_CREDS_MODIFY events provide information about modifications credentials for a user identity.

AUDIT_AUTHN

AUDIT_AUTHN events provide information about authentication. The record includes the following information:

- The type of authentication used. For example, the record might show `basicAuthRFC2617` for an HTTP basic authentication or show `certificate` for a public key authentication.
- The user who requested authentication.
- Whether the authentication was successful. If the authentication failed, the record shows the reason for the failure. For example, the record might show that the account was locked out because of repeated password authentication failures.

AUDIT_AUTHN_CREDS_MODIFY

AUDIT_AUTHN_CREDS_MODIFY events provide information about modifications of credentials for a user identity.

AUDIT_AUTHN_TERMINATE

AUDIT_AUTHN_TERMINATE events provide information about when a user ends a session. The record includes information about the user whose session is ending and the reason why that session is ending. Sessions can end for many reasons, including a log out by a user, an administrator action to end a session for a user, or a timeout.

Note: When you use the session management server for session storage, configure the session management server to generate the AUDIT_AUTHN_TERMINATE events. Because WebSEAL and the Web server plug-in cannot determine when the session ends, they cannot generate the AUDIT_AUTHN_TERMINATE event.

AUDIT_AUTHZ

AUDIT_AUTHZ events provide information about authorization decisions.

A WebSEAL record includes the following information:

- The user who attempted access. For example, the record might show `joe` as the user.
- The object that the user attempted to access. For example, the record might show `/WebSEAL/www.example.com-default/index.html` as the object.
- The permission that is needed for access. For example, the record might show `r` for read permission.
- The access decision. For example, the record might show `permitted` or `denied` as the access decision. The access decision is not necessarily final. The ultimate decision depends on the application that is making the authorization check.

Consider the situation when user `joe` attempts to access the `index.html` web page. Joe has the permission to access this object, but he might still be denied access. Assume that the policy associated with the object requires clients to use an encrypted transport to view the object. If Joe attempts to access the object by using the HTTP transport protocol, he is denied access although the outcome of the access record shows `permitted`.

Plug-in for Web Servers does not generate AUDIT_AUTHZ events. Application-level authorization results generate AUDIT_RESOURCE_ACCESS events.

AUDIT_MGMT_CONFIG

AUDIT_MGMT_CONFIG events provide information about configuration and other management operations for a server. These events apply to the policy server and policy proxy server only.

AUDIT_MGMT_POLICY

AUDIT_MGMT_POLICY events provide information about security policy management operations, such as the creation of an access control list, protected object policy, or an authorization rule. These events apply to the policy server and policy proxy server only.

AUDIT_MGMT_REGISTRY

AUDIT_MGMT_REGISTRY events provide information about the users and groups in the user registry, such as creating users and groups or modifying properties for users and groups. These events apply to the policy server only.

AUDIT_MGMT_RESOURCE

AUDIT_MGMT_RESOURCE events provide information about resource management operations. These events apply to the policy server and policy proxy server only.

AUDIT_PASSWORD_CHANGE

AUDIT_PASSWORD_CHANGE events provide information about a password change. These events are generated when users change their passwords or when an administrator changes the password.

Plug-in for Web Servers generates AUDIT_PASSWORD_CHANGE events only for user-initiated password changes, not administrator-initiated password changes.

AUDIT_RESOURCE_ACCESS

AUDIT_RESOURCE_ACCESS events provide information about access to a resource, such as a file or HTTP request or response events outside of AUDIT_AUTHZ events.

AUDIT_RUNTIME

AUDIT_RUNTIME events provide information about servers. These events are generated when a server starts or stops. Runtime events can also include heartbeat events that verify whether a server is running, and statistical events.

AUDIT_RUNTIME events are not generated for administrative operations.

Audit events by server

Each Security Access Manager server supports different audit events. The following list shows which Common Auditing Service events can be used with each Security Access Manager server:

Policy and policy proxy server
The policy server and the policy proxy server can generate the following events that can be recorded by the Common Auditing Service:
- AUDIT_AUTHN
- AUDIT_MGMT_CONFIG
- AUDIT_MGMT_POLICY
- AUDIT_MGMT_REGISTRY
- AUDIT_MGMT_RESOURCE
- AUDIT_RUNTIME

Authorization server
The authorization server can generate the following events that can be recorded by the Common Auditing Service:
- AUDIT_AUTHZ
- AUDIT_RUNTIME

WebSEAL
Each WebSEAL instance can generate the following events that can be recorded by the Common Auditing Service:
- AUDIT_AUTHN
- AUDIT_AUTHN_TERMINATE
- AUDIT_PASSWORD_CHANGE
- AUDIT_RESOURCE_ACCESS
- AUDIT_RUNTIME

For definitive information about whether a WebSEAL server allowed access to a resource, use AUDIT_RESOURCE_ACCESS events.

Plug-in for Web Servers
Each Web server plug-in can generate the following events that can be recorded by the Common Auditing Service:
- AUDIT_AUTHN
- AUDIT_AUTHN_CREDS_MODIFY
- AUDIT_AUTHN_TERMINATE
- AUDIT_PASSWORD_CHANGE
- AUDIT_RESOURCE_ACCESS
- AUDIT_RUNTIME

Plug-in for Web Servers does not generate AUDIT_AUTHZ events. The results for application-level authorization generate AUDIT_RESOURCE_ACCESS events.

Plug-in for WebLogic server
The Plug-in for WebLogic server does not generate events that can be recorded by the Common Auditing Service.

Session management server
Each instance of the session management server can generate the following events that can be recorded by the Common Auditing Service:
- AUDIT_AUTHN
- AUDIT_AUTHN_TERMINATE
- AUDIT_AUTHZ
- AUDIT_RUNTIME

Chapter 15. Common Auditing Service for C-based Security Access Manager servers

If you are using a C-based Security Access Manager server, use the Common Auditing Service embedded C client to send audit events to the Common Auditing Service audit server. Event configuration for the C client is controlled with the server-specific auditing configuration files. To start sending events to the audit server, you must generate the initial configuration files and parameter settings by running the **amauditcfg** utility. The required procedure is described in "Configuring to send audit events through the C client."

After you generate the initial configuration file settings by using the **amauditcfg** utility, use the **pdadmin config modify** command to change settings in the configuration files. Do not use an ASCII editor to modify a configuration file. For information about modifying configuration files, see "Using the config modify command for auditing" on page 145.

Configuring to send audit events through the C client

To start the sending of events to the Common Auditing Service audit server through the C client, use the **amauditcfg** utility.

About this task

The **amauditcfg** utility configures a Security Access Manager server to start (or stop) by using Common Auditing Service.

The **amauditcfg** utility generates the server-specific auditing configuration files that are described in "Common Auditing Service configuration files" on page 142. After these files are generated, use the **pdadmin config modify** command to change one or more configuration settings in the files.

For complete information about using the **amauditcfg** utility, including how to enable secure communications with the audit server, see "amauditcfg" on page 415.

Procedure

- The following example shows a simple way to use the **amauditcfg** utility to configure the use of Common Auditing Service:

  ```
  amauditcfg -action config -srv_cfg_file configuration_file -audit_srv_url url
  ```

- The following example shows how to use the **amauditcfg** utility to configure the default WebSEAL server. The WebSEAL server is on an AIX system. Use the configuration to set up Common Auditing Service over a secure (SSL) connection:

  ```
  /opt/PolicyDirector/sbin/amauditcfg -action config \
    -srv_cfg_file /opt/pdweb/etc/webseald-default.conf \
    -enable_ssl yes -audit_key_file /certs/WSclient.kdb -audit_stash_file \
    /certs/WSclient.sth -enable_pwd_auth yes -audit_id root \
    -audit_pwd da21cars -audit_srv_url \
    http://carsserver.example.com:9443/CommonAuditService/services/Emitter
  ```

  After starting this command, the follow output shows a successful completion:

```
Gathering system information.
Parsing the command line.
Validating the information.
Configuring the server for common auditing and reporting services.
Configuration completed successfully.
```

Common Auditing Service configuration files

Running the **amauditcfg** utility generates the following configuration files to control the configuration of the Common Auditing Service C clients.

These configuration files are used for auditing and statistics purposes:

pdaudit.pdmgr.conf
Configure the Common Auditing Service for the Security Access Manager policy server. Do not confuse this configuration file with the `ivmgrd.conf` configuration file.

pdaudit.pdproxymgr.conf
Configure the Common Auditing Service for a Security Access Manager policy proxy server. Do not confuse this configuration file with the `pdmgrproxyd.conf` configuration file.

pdaudit.pdacld.conf
Configure the Common Auditing Service for the Security Access Manager authorization server. Do not confuse this configuration file with the `ivacld.conf` configuration file.

pdaudit.*instance*-webseald-*host*.conf
Configure the Common Auditing Service for a specific instance of a Security Access Manager WebSEAL server. Do not confuse this configuration file with the `webseald-`*`instance`*`.conf` configuration file.

pdaudit.webpi.conf
Configure the Common Auditing Service for a Security Access Manager Plug-in for Web Servers. Do not confuse this configuration file with the `pdwebpi.conf` configuration file.

pdaudit.appsvr.conf
Configure the Common Auditing Service for any Security Access Manager resource managers. Do not confuse this configuration file with the `aznAPI.conf` configuration file.

For more information about these configuration files, see "Configuration file reference" on page 377.

Policy server: Configuration Settings

Running the **amauditcfg** utility sets the following entries in the `pdaudit.pdmgrd.conf` configuration file:

```
[cars-filter]
auditevent=authn
auditevent=mgmt_config
auditevent=mgmt_policy
auditevent=mgmt_registry
auditevent=mgmt_resource
auditevent=runtime
```

Policy proxy server: Configuration Settings

Running the **amauditcfg** utility sets the following entries in the pdaudit.pdproxymgrd.conf configuration file:

```
[cars-filter]
auditevent=authn
auditevent=mgmt_config
auditevent=mgmt_policy
auditevent=mgmt_registry
auditevent=mgmt_resource
auditevent=runtime
```

Authorization server: Configuration Settings

Running the **amauditcfg** utility sets the following entries in the pdaudit.pdacld.conf configuration file:

```
[cars-filter]
auditevent=authz
auditevent=runtime
```

WebSEAL: Configuration settings

Running the **amauditcfg** utility sets the following entries in the pdaudit.webseald-*instance*.conf configuration file:

```
[cars-filter]
auditevent=authn
auditevent=authn_creds_modify
auditevent=authn_terminate
auditevent=authz
auditevent=password_change
#auditevent=runtime
```

Note: The default audit settings that are defined by the **amauditcfg** utility are probably inappropriate for a WebSEAL server configuration. In particular, authz (AUDIT_AUTHZ) events do not provide definitive information about whether access was allowed to a particular resource. For definitive access information, use the resource_access (AUDIT_RESOURCE_ACCESS) event.

For information about audit configurations for WebSEAL, see "Configurations for WebSEAL." For information about adding or modifying event types, see "Using the config modify command for auditing" on page 145.

Configurations for WebSEAL

Configuring the level of auditing for WebSEAL depends on your business needs and policies. The following levels of auditing are samples of levels that you can set in the pdaudit.webseald-*instance*.conf configuration file:

Level 1 auditing
Audit only unsuccessful authentication events. The configuration file must contain the following details:

```
[cars-filter]
auditevent=authn,outcome=unsuccessful
```

Level 2 auditing
Increases the level of auditing from Level 1 to include all authentication events. The configuration file must contain the following details:

```
[cars-filter]
auditevent=authn
```

Level 3 auditing

Increases the level of auditing from Level 2 to include all sessions. The configuration file must contain the following details:

```
[cars-filter]
auditevent=authn
auditevent=authn_terminate
```

Level 4 auditing

Increases the level of auditing from Level 3 to include unsuccessful access events. The configuration file must contain the following details:

```
[cars-filter]
auditevent=authn
auditevent=authn_terminate
auditevent=resource_access,outcome=unsuccessful
```

Note: An HTTP request is considered unsuccessful if any of the following conditions is true:

- An authentication failure occurred.
- An authorization failure occurred.
- The request dispatch from WebSEAL to the junction failed.
- The HTTP response code is not a `2xx` or `3xx` code.

If none of these conditions is true, the request is considered successful.

Level 5 auditing

Increases the level of auditing from Level 4 to include all access events. The configuration file must contain the following details:

```
[cars-filter]
auditevent=authn
auditevent=authn_terminate
auditevent=resource_access
```

This level of auditing generates the most events. Because of the volume of events, this configuration might not be practical. If the WebSEAL server does not receive heavy traffic, this configuration might be practical.

Plug-in for Web Servers: Configuration settings

Running the **amauditcfg** utility sets the following entries in the `pdaudit.pdwebpi.conf` configuration file:

```
[cars-filter]
auditevent=authn
#auditevent=authn_creds_modify
auditevent=authn_terminate
auditevent=authz
auditevent=password_change
#auditevent=resource_access
auditevent=runtime
```

Note: Although the **amauditcfg** utility sets the `authz` event (AUDIT_AUTHZ), Plug-in for Web Servers does not generate this type of event. Results of application-level authorization generate `resource_access` (AUDIT_RESOURCE_ACCESS) events. Make appropriate changes by using the **config modify** command. For information about adding or modifying event types, see “Using the config modify command for auditing” on page 145.

Using the config modify command for auditing

To change a setting in an auditing configuration file, use the **pdadmin config modify** command. Do not change the contents of any configuration file by using an ASCII editor.

Before you use the **pdadmin config modify** command, log in locally with the **pdadmin login –l** command.

For information about these commands, see "config modify" on page 405 and "login" on page 409.

Starting event auditing

To start Common Auditing Service auditing, change the server-specific configuration file.

Procedure

1. Log in locally.
2. Set the doAudit entry to yes in the [cars-client] stanza of the server-specific configuration file.

   Enter the following command:

   ```
   config modify keyvalue set config_file cars-client doAudit yes
   ```

   For example, to enable Common Auditing Service auditing for an AIX policy server, enter the following command:

   ```
   config modify keyvalue set /opt/PolicyDirector/etc/audit/pdaudit.pdmgr.conf \
    cars-client doAudit yes
   ```

Stopping event auditing

To stop Common Auditing Service auditing, change the server-specific configuration file.

Procedure

1. Log in locally.
2. Set the doAudit entry to no in the [cars-client] stanza of the server-specific configuration file.

   Enter the following command:

   ```
   config modify keyvalue set config_file cars-client doAudit no
   ```

   For example, to stop Common Auditing Service auditing for an AIX policy server, enter the following command:

   ```
   config modify keyvalue set /opt/PolicyDirector/etc/audit/pdaudit.pdmgr.conf \
     cars-client doAudit no
   ```

Adding event types to the event filter

After you enable event auditing, you can add events to be forwarded to the Common Auditing Service audit server.

Procedure

1. Log in locally.
2. Append an auditevent entry with the new event type to the [cars-filter] stanza of the server-specific configuration file.

   Enter the following command:

```
config modify keyvalue append config_file cars-filter auditevent type
```

For example, to add runtime event auditing for an AIX policy server, enter the following command:

```
config modify keyvalue append /opt/PolicyDirector/etc/audit/pdaudit.pdmgr.conf \
 cars-filter auditevent runtime
```

For example, to add resource access event auditing to an AIX WebSEAL server, enter the following command:

```
config modify keyvalue \
append /opt/PolicyDirector/etc/audit/pdaudit.default-webseald-aix.ibm.com.conf \
 cars-filter auditevent runtime
```

Removing event types from the event filter

After you enable event auditing, you can stop events from being forwarded to the Common Auditing Service server.

About this task

To remove an event, remove the auditevent entry for that event type from the [cars-filter] stanza of the server-specific configuration file.

Procedure

1. Log in locally.
2. Enter the following command:

   ```
   config modify keyvalue remove config_file cars-filter auditevent type
   ```

   For example, to remove authorization event auditing from an AIX policy server, enter the following command:

   ```
   config modify keyvalue remove /opt/PolicyDirector/etc/audit/pdaudit.pdmgr.conf \
    cars-filter auditevent authz
   ```

   For example, to remove the authorization event auditing from an AIX WebSEAL server, enter the following command:

   ```
   config modify keyvalue \
    remove /opt/PolicyDirector/etc/audit/pdaudit.default-webseald-aix.ibm.com.conf \
    cars-filter auditevent authz
   ```

Enhancements to improve audit event data throughput and minimize data loss

To minimize the loss of audit events, specify the following configuration options for Common Auditing Service:

- Provide a separate file system for cache files.
- Monitor this file system regularly to determine whether more space is needed, or if the Common Auditing Service audit server needs attention.
- Ensure that a reasonable value is used for the tempStorageFullTimeout property. This property specifies the number of seconds that the application waits before discarding events. See “tempStorageFullTimeout” on page 391 for information about setting the tempStorageFullTimeout property.
- Specify the size and the number of cache files that use the maxCacheFiles and maxCacheFileSize properties. Calculate these values by using the following information:
  - Maximum file system space available minus 5-10% of file system space = usable file system space
  - First, multiply the maximum cache file size times the number of event queue threads (numberEQThreads, the default value is 1.) Then, subtract this value from the usable file system space to determine the remaining file system

space. Divide the remaining file system space by the maximum cache file size to determine the maximum number of cache files to create.

Assume that the Common Auditing Service C client application is unable to send events to the Common Auditing Service audit server by using these properties. In this case, the application attempts to write the events to a cache file. Further, the application waits the period that is specified in the tempStorageFullTimeout option and then attempts another update to the cache file. If the application is still unable to update the cache file, the event is discarded. The application records a count of the number of events that are discarded.

Assume that the Common Auditing Service audit server is able to receive events. In this case, the application logs the number of events that were discarded and sends the cached events to the audit server.

Ensure that the following are configured to prevent the file system from filling up:

- The configuration settings for the Common Auditing Service C client disk cache.
- Use of the file system on which it is allocated.

Otherwise, file system errors can lead to a loss of the audit events that are present in the disk cache files. For example, specifying a value for tempStorageFullTimeout without appropriately specifying the number and size of disk cache files can cause the loss of cached audit events.

Note: When you specify values for the options that are described in "Configuration file stanza reference" on page 379, ensure that they are within the documented range. Behavior can become unpredictable if entries are set to values that are not documented or are larger than the ones supported by the architecture.

Chapter 16. Common Auditing Service for session management servers

You can record audit events for the session management server by sending events to the Common Auditing Service audit server.

The session management server is the only Java based server for Security Access Manager. You cannot use native Security Access Manager auditing with the session management server.

To enable Common Auditing Service to record audit events for the session management server, modify either the `textfile` or `webservice` template configuration file so that the embedded Java client of the Common Auditing Service can:

- Talk with the audit server.
- Know which event to record.

See "Creating a configuration file for auditing session management server events."

Creating a configuration file for auditing session management server events

During configuration of the session management server, you are asked to supply the path to an emitter configuration file for auditing. Create the configuration file by modifying a provided template file as described in these instructions.

About this task

Default configuration template files are included with the session management server. The template files contain properties to configure the Common Auditing Service Java client. You can use either of the following two template files:

`textfile_emitter.properties.template`
Logs events to a text file.

`webservice_emitter.properties.template`
Send events to the Common Auditing Service Web Service with SOAP.

Procedure

1. Determine which configuration template file you want to use:
   - `textfile_emitter.properties.template`
   - `webservice_emitter.properties.template`
2. Locate the template file in one of the following operating system-specific locations:

   AIX, Linux, and Solaris operating systems
   `/opt/pdsms/etc`

   Windows
   `C:\Program Files\Tivoli\PDSMS\etc`
3. Copy the template file, then set the template properties to valid values. To view the contents of the template file, see:
   - "Contents of the webservice_emitter.properties.template file" on page 150

- "Contents of the textfile_emitter.properties.template file" on page 154

4. Save the configuration file.

Results

The configuration file is ready. During session management server configuration, when you specify the location of this auditing configuration file, the file is copied to the following directory: *WAS_HOME*/profiles/*profile_name*/installedApps/*cell_name*/app_name.ear/DSess.war/WEB-INF/*server_name*/DSessAudit.properties.

For instructions on configuring the session management server, see "Configuring session management server" in the *IBM Security Access Manager for Web Shared Session Management Administration Guide*.

Note: To send session events to the Common Auditing Service for auditing events, you must enable Security Access Manager integration during session management server configuration.

Contents of the webservice_emitter.properties.template file

The webservice_emitter.properties.template file contains the following content:

```
# Licensed Materials - Property of IBM
# 5748-XX8
# (c) Copyright International Business Machines Corp. 2007
# All Rights Reserved
# US Government Users Restricted Rights - Use, duplicaion or disclosure
# restricted by GSA ADP Schedule Contract with IBM Corp.
#

####################################################################
#             webservice_emitter.properties.template               #
####################################################################
#
# This template file describes how to control a session management
# server (SMS) common audit and reporting service (CARS) web
# service emitter.
#
# During the configuration of session management server auditing
# you will be asked to supply the path to an emitter configuration
# file.
#
# Create a CARS Web Service emitter configuration file by copying
# this template and setting the properties to valid values.
#
####################################################################

#
# OPTIONAL modifyLocation
#
# This parameter controls whether the pathname used to specify the
# CARS web service emitter cache file location is made relative to
# the SMS server instance.
#
# The value of the property named in modifyLocation will be modified
# at runtime to be an absolute path including elements that identify
# the SMS's deployment in WAS. If the generated directory name does
# not already exist it will be created.
#
# The original value is modified by appending the following
# elements:
#  <originalValue>/<cell name>/<node name>/<application name>
```

```
#
# If the original value is a relative pathname, it will be made
# relative to <Websphere profile>/logs/<server name>/smsaudit
#
# If you are running multiple session management server instances
# on the same node you can use this parameter to construct instance
# specific cache file directories.
#
#modifyLocation = com.ibm.cars.events.emitter.ICARSEmitterProperties
.diskCacheDir

#
# OPTIONAL doNotDeliver
#
# The event delivery policy of the CARS server the emitter connects
# to is not discoverable programmatically.
#
# If it is considered desirable to restrict the types of security
# events generated by the SMS, before they are filtered at the CARS
# server, uncomment the doNotDeliver policy and specify the types of
# security events that are of no interest.
#
# The session management server can generate security events of the
# following types:
#     IBM_SECURITY_AUTHN
#     IBM_SECURITY_AUTHN_TERMINATE
#     IBM_SECURITY_AUTHZ
#     IBM_SECURITY_RUNTIME
#
#doNotDeliver = IBM_SECURITY_AUTHN,IBM_SECURITY_AUTHN_TERMINATE,
IBM_SECURITY_AUTHZ,IBM_SECURITY_RUNTIME

#
# REQUIRED com.ibm.cars.events.emitter.ICARSEmitterProperties.emitterName
#
# Specify the name of the CARS emitter implementation.
#
# The name of the CARS Web Service emitter implementation is
# emitterNameCarsWebService.
#
com.ibm.cars.events.emitter.ICARSEmitterProperties.emitterName =
emitterNameCarsWebService

#
# REQUIRED com.ibm.cars.events.emitter.ICARSEmitterProperties.serverURL
#
# The URL to the CARS Web Service.
#
# Example, for secure communication:
#   https://sms.example.ibm.com:9443/CommonAuditService/services/Emitter
# Example, for non-secure communication:
#   http://sms.example.ibm.com:9080/CommonAuditService/services/Emitter
#
com.ibm.cars.events.emitter.ICARSEmitterProperties.serverURL =

#
# OPTIONAL com.ibm.cars.events.emitter.ICARSEmitterProperties.trustStore
#
# The URL to the CARS Web Service.
# Required if the server's endpoint is accessed over ssl (i.e. https)
#
# Example, for secure communication:
#   https://sms.example.ibm.com:9443/CommonAuditService/services/Emitter
#
#com.ibm.cars.events.emitter.ICARSEmitterProperties.trustStore =

#
```

```
# OPTIONAL com.ibm.cars.events.emitter.ICARSEmitterProperties.trustStorePassword
#
# This is the password of the KeyStore file.
# This is not needed when retrieving the root signer's public certificate.
#
# Required if secure communication with the CARS Web Service is used, AND
# client certificate authentication is used.
#
#com.ibm.cars.events.emitter.ICARSEmitterProperties.trustStorePassword =

#
# OPTIONAL com.ibm.cars.events.emitter.ICARSEmitterProperties.
isClientBasicAuthenticationRequired
#
# This is set to true if the server requires client side basic authentication.
#
# Required if basic authentication is used to authenticate the user to
# the CARS Web Service. This should only be used in conjunction with SSL.
#
# Note: The client password will be stored in clear text in memory while
# the process is running. However, it will be base64 encoded while it is
# going over the wire. This should only be used with SSL unless you want
# everyone knowing your password.
#
#com.ibm.cars.events.emitter.ICARSEmitterProperties.isClientBasicAuthenticationRequired =

#
# OPTIONAL com.ibm.cars.events.emitter.ICARSEmitterProperties.clientBasicAuthenticationUserName
#
# This is the user name of the client.
#
# If specified, it will be used (along with clientPassword) to
# authenticate the client to the Web Service.
#
# Required if basic authentication is used to authenticate the user to
# the CARS Web Service.
#
#com.ibm.cars.events.emitter.ICARSEmitterProperties.clientBasicAuthenticationUserName =

#
# OPTIONAL com.ibm.cars.events.emitter.ICARSEmitterProperties.clientBasicAuthenticationPassword
#
# This is the password of the client.
#
# If specified, it will be used (along with clientBasicAuthicationUserName)
# to authenticate the client to the Web Service.
#
# Required if basic authentication is used to authenticate the user to
# the CARS Web Service.
#
#com.ibm.cars.events.emitter.ICARSEmitterProperties.clientBasicAuthenticationPassword =

#
# OPTIONAL com.ibm.cars.events.emitter.ICARSEmitterProperties.transferSize
#
# The number of events that would be bundled and sent to the server on
# a single request. If the transferSize is set to 1, then events will
# not be queued, but will be written directly to the cache before
# returning to the calling application. Otherwise, this number is only
# a suggested number, the Web Service Emitter implementation may
# actually send more or less than this number.
#
# Default is 10
#
#com.ibm.cars.events.emitter.ICARSEmitterProperties.transferSize =

#
```

```
# REQUIRED com.ibm.cars.events.emitter.ICARSEmitterProperties.diskCacheDir
#
# The directory where cache files will exist.
#
# This parameter must point to an existing writeable directory,
# otherwise initialization will fail.
#
# The value of this parameter may be altered if the modifyLocation
# parameter is also used.
com.ibm.cars.events.emitter.ICARSEmitterProperties.diskCacheDir =

#
# REQUIRED com.ibm.cars.events.emitter.ICARSEmitterProperties.diskCacheFileName
#
# The name prefix for cache files.
# The actual name will be <diskCacheFileName>.<time stamp>.nn
#
com.ibm.cars.events.emitter.ICARSEmitterProperties.diskCacheFileName =

#
# OPTIONAL com.ibm.cars.events.emitter.ICARSEmitterProperties.maxCacheFileSize
#
# The maximum size of a single cache file.
#
# Default is 1000000
#
#com.ibm.cars.events.emitter.ICARSEmitterProperties.maxCacheFileSize =

#
# OPTIONAL com.ibm.cars.events.emitter.ICARSEmitterProperties.maxCacheFiles
#
# The total number of cache files that can exist at any one time.
# This number does not include any corrupted files. In addition,
# if at least 3 files cannot be created, initialization will fail.
#
# Default is 1000
#
#com.ibm.cars.events.emitter.ICARSEmitterProperties.maxCacheFiles =

#
# OPTIONAL com.ibm.cars.events.emitter.ICARSEmitterProperties.MaxQueueSize
#
# The maximum number of messages allowed on the queue.
#
# Default is 1000
#
#com.ibm.cars.events.emitter.ICARSEmitterProperties.MaxQueueSize =

#
# OPTIONAL com.ibm.cars.events.emitter.ICARSEmitterProperties.numberEQThreads
#
# The maximum number of Queue Manager threads.
# This number should be at most half of the MAX_CACHE_FILES value.
#
# Default is 4
#
#com.ibm.cars.events.emitter.ICARSEmitterProperties.numberEQThreads =

#
# OPTIONAL com.ibm.cars.events.emitter.ICARSEmitterProperties.numberCMThreads
#
# The number of threads for draining the cache to the event server.
#
# Default is 4
#
#com.ibm.cars.events.emitter.ICARSEmitterProperties.numberCMThreads =
#
# OPTIONAL com.ibm.cars.events.emitter.ICARSEmitterProperties.rebindInterval
#
# Number of seconds to wait between attempts to bind to the event server.
#
# Default is 10
#
#com.ibm.cars.events.emitter.ICARSEmitterProperties.rebindInterval =

#
# OPTIONAL com.ibm.cars.events.emitter.ICARSEmitterProperties.numberRetries
#
# The number of times to retry each network transfer whan an error is received.
#
# Default is 3
#
#com.ibm.cars.events.emitter.ICARSEmitterProperties.numberRetries =

#
# OPTIONAL com.ibm.cars.events.emitter.ICARSEmitterProperties.retryInterval
#
# The number of seconds to wait before retrying when an error is received
# on a network transfer.
#
# Default is 2
#
#com.ibm.cars.events.emitter.ICARSEmitterProperties.retryInterval =

#
# REQUIRED com.ibm.cars.events.emitter.ICARSEmitterProperties.tempStorageFullTimeout
#
# The number of seconds to wait before discarding events when
# temporary storage is full.
#
# A value of 0 means there is no waiting.
# A value of -1 means wait forever.
#
# If transferSize = 1, temporary storage is full when the disk cache
# cannot be written to. Otherwise temporary storage is full when the audit
# event queue is full AND the disk cache cannot be written to.
#
# The disk cache cannot be written to when:
# The maxCacheFiles value has been reached and each cache file has
# reached the maxCacheFileSize (or will exceed it on the next write),
# OR, the cache file cannot be written to due to a system error.
#
com.ibm.cars.events.emitter.ICARSEmitterProperties.tempStorageFullTimeout =

#
# REQUIRED com.ibm.cars.events.emitter.ICARSEmitterProperties.shutDownTimeout
#
# The number of seconds to wait for the audit event queue to drain during
# a process shutdown.
# A value of -1 means to wait forever.
#
com.ibm.cars.events.emitter.ICARSEmitterProperties.shutDownTimeout =
```

Contents of the textfile_emitter.properties.template file

The textfile_emitter.properties.template file contains the following content:

```
# Licensed Materials - Property of IBM
# 5748-XX8
# (c) Copyright International Business Machines Corp. 2007
# All Rights Reserved
# US Government Users Restricted Rights - Use, duplicaion or disclosure
# restricted by GSA ADP Schedule Contract with IBM Corp.
#

####################################################################
#             textfile_emitter.properties.template                 #
####################################################################
#
# This template file describes how to control a session management
# server (SMS) common audit and reporting service (CARS) text file
# emitter.
#
# During the configuration of session management server auditing
# you will be asked to supply the path to an emitter configuration
# file.
#
# Create a text file emitter configuration file by copying this
# template and setting the properties to valid values.
#
####################################################################

#
# OPTIONAL modifyLocation
#
# This parameter controls whether the pathname used to specify the
# text file emitter audit files location is made relative to the SMS
# server instance.
#
# The value of the property named in modifyLocation will be modified
# at runtime to be an absolute path including elements that identify
# the SMS's deployment in WAS. If the generated directory name does
# not already exist it will be created.
#
# The original value is modified by appending the following
# elements:
#  <originalValue>/<cell name>/<node name>/<application name>
#
# If the original value is a relative pathname, it will be made
# relative to <Websphere profile>/logs/<server name>/smsaudit
#
# If you are running multiple session management server instances
# on the same node you can use this parameter to construct instance
# specific logging directories.
#
#modifyLocation = com.ibm.cars.events.emitter.ICARSEmitterProperties
.auditFileLocation

#
# OPTIONAL doNotDeliver
#
# The event delivery policy of the text file emitter (if any) is not
# discoverable programmatically.
#
# If it is considered desirable to restrict the types of security
# events generated by the SMS, before they are filtered at the CARS
# server, uncomment the doNotDeliver policy and specify the types of
# security events that are of no interest.
#
# The session management server can generate security events of the
# following types:
#     IBM_SECURITY_AUTHN
#     IBM_SECURITY_AUTHN_TERMINATE
#     IBM_SECURITY_AUTHZ
#     IBM_SECURITY_RUNTIME
#
#doNotDeliver = IBM_SECURITY_AUTHN,IBM_SECURITY_AUTHN_TERMINATE,
IBM_SECURITY_AUTHZ,IBM_SECURITY_RUNTIME
```

```
#
# REQUIRED com.ibm.cars.events.emitter.ICARSEmitterProperties.emitterName
#
# Specify the name of the CARS emitter implementation.
#
# The name of the text file emitter implementation is
# emitterNameCarsTextFile.
#
com.ibm.cars.events.emitter.ICARSEmitterProperties.emitterName =
emitterNameCarsTextFile

#
# The text file emitter writes to a rotating set of log files named
# according to a java.util.logging.FileHandler.pattern constructed
# as follows:
#
#     <auditFileLocation>/<auditFilePrefix>_audit%g.log
#
# where the directory separator will be appended to the auditFileLocation
# if necessary, and "%g" is the java FileHandler's generation number used
# to distinguish rotated logs.
#

# REQUIRED com.ibm.cars.events.emitter.ICARSEmitterProperties.
auditFileLocation
#
# Specify the directory where audit files will be created.
#
# This parameter must point to an existing writeable directory,
# otherwise initialization will fail.
#
# The value of this parameter may be altered if the modifyLocation
# parameter is used.
com.ibm.cars.events.emitter.ICARSEmitterProperties.auditFileLocation =

#
# OPTIONAL com.ibm.cars.events.emitter.ICARSEmitterProperties.auditFilePrefix
#
# Specify a prefix to use for the audit file name.
#
com.ibm.cars.events.emitter.ICARSEmitterProperties.auditFilePrefix =

#
# REQUIRED com.ibm.cars.events.emitter.ICARSEmitterProperties.auditFileSize
#
# Specify the maximum number of bytes to write to any one audit file.
#
com.ibm.cars.events.emitter.ICARSEmitterProperties.auditFileSize = 2000000

#
# REQUIRED com.ibm.cars.events.emitter.ICARSEmitterProperties.maxAuditFiles
#
# Specify the number of audit files to use. Count must be at least 1.
#
com.ibm.cars.events.emitter.ICARSEmitterProperties.maxAuditFiles = 5
```

Chapter 17. Securing data flow in the operating environment

This topic describes how to secure the exchange of data among components in the Common Auditing Service audit server and Web Service clients. If you are using the C client to generate events in a WebSphere Application Server single server environment, see "Securing C client events."

Securing C client events

Enabling security for event logging from the Common Auditing Service C client involves configuring the server and the client. Server configuration involves:

- Enabling WebSphere Application Server global security.
- Using the WebSphere Application Server Secure Sockets Layer (SSL) functionality.
- Mapping Common Auditing Service Web service roles to users or groups.

The C client configuration involves specifying the keyring (`.kdb`) and stash (`.sth`) files for SSL and user name and password for client authentication.

Configuring the server

Configuring the server to use the client involves setting up:

- WebSphere Application Server security
- Secure Sockets Layer (SSL)
- Security roles for users and groups

WebSphere Application Server security

The following steps are required to set up WebSphere Application Server security:

Procedure

1. Log on to the WebSphere Application Server administrative console.
2. Configure the user registry as described in one of the following procedures for the different user registries:
   - **Local operating system registry:** See “Configuring the operating system registry” on page 158.
   - **LDAP registry:** See “Configuring the LDAP registry” on page 158.
   - **Custom registry:** See “Configuring a custom registry” on page 160.
   - **Federated registry:** See “Configuring a federated registry” on page 160.
3. Enable the administrative and application security option for the configured user registry:
   a. Click **Security** > **Global security** to display the "Global security" panel.
   b. Select **Enable administrative security**.
   c. Select **Enable application security**.
   d. In **User account repository** > **Available realm definitions**, select the configured user registry (for example, `Standalone LDAP registry`).
   e. In **User account repository**, click **Set as current**. This selection validates properties that are configured for the selected realm.

f. Click **Apply** and then save the changes. If you are in a WebSphere Application Server Network Deployment environment, be sure to select **Synchronize changes with Nodes** before you save the changes.

4. Manually add the security policy for the DB2 JDBC driver. See Configuring security policy for the JDBC provider.
5. Enable the Java 2 security option:
   a. Click **Security** > **Global security** > **Use Java 2 security to restrict application access to local resources**.
   b. Select **Warn if applications are granted custom permissions**.
   c. Select **Restrict access to resource authentication data**.
6. Click **Apply** and save the changes.

Configuring the operating system registry:

Configure the local operating system registry settings.

About this task

This task covers the steps to configure a local operating system registry. However, use an LDAP registry (or a federated repository that includes an LDAP registry) for the following scenarios:

- You want to maintain consistency of the registry between nodes in a clustered environment.
- The Network Deployment cell (all of the nodes) is not on a single server.
- WebSphere Application Server is running on AIX, Linux, or Solaris as a non-root user.

See "Configuring the LDAP registry."

Procedure

1. Log on to the WebSphere Application Server administrative console.
2. Click **Security** > **Global security** to display the "Global security" panel.
3. In **User account repository** > **Available realm definitions**, select **Local operating system** from the drop-down list.
4. Click **Configure**.
5. Specify the **Primary administrative user name property**, which is a user with administrative privileges who is defined in the local operating system.
6. Select one of the following options:
   - Click **Automatically generated server identity**.
   - Click **Server identity that is stored in the repository** and specify the following properties:
     - Server user ID (for example, `root`). This value must be a valid user ID in the local operating system registry.
     - Server user password (for example, `abc26xyz`).
7. Click **OK** and then save the changes.

Configuring the LDAP registry:

Use Lightweight Directory Access Protocol (LDAP) user registries when users and groups are in an external LDAP directory. In clustered environments, LDAP registries are used to maintain consistency of the registry between nodes in the cluster.

About this task

Use an LDAP registry (or a federated repository that includes an LDAP registry) for the following scenarios:

- You want to maintain consistency of the registry between nodes in a clustered environment.
- The Network Deployment cell (all of the nodes) is not on a single server.
- WebSphere Application Server is running on AIX, Linux, or Solaris as a non-root user.

Procedure

1. Log on to the WebSphere Application Server administrative console.
2. Click **Security** > **Global security** to display the "Global security" panel.
3. In **User account repository** > **Available realm definitions**, select **Standalone LDAP registry** from the drop-down list.
4. Click **Configure**.
5. Specify the **Primary administrative user name** property, which is a name of a user in your LDAP registry who has administrative privileges.
6. In the **LDAP** area, specify the following properties:
   - **Type of LDAP server**: For example, `IBM Tivoli Directory Server`.
   - **Host**: For example: `server1.example.ibm.com`.
   - **Port**: For example, `389`.
   - **Base distinguished name**: For example, `ou=tivoli,o=ibm,c=us`.
   - **Search timeout**: For example, `120`.
   - **Reuse connection**: Enabled by default. This property prevents the LDAP connection from reestablishing on each LDAP access.
   - **Ignore case for authorization**: Enable this property if your LDAP server requires it.
7. In the **Security** area, specify the following properties:
   - In the **Server user identity** area, choose one of the following options:
     - Select **Automatically generated server identity**
     - Select **Server identity that is stored in the repository** and specify the following properties:
       - Server user ID (for example, `root`), which is the operating system user ID that the application server is using for security purposes.
       - Server user password (for example, `abc26xyz`).
   - **Bind distinguished name** (for example, `cn=root,ou=tivoli,o=ibm,c=us`)
   - **Bind password** (for example, `abc26xyz`)
   - Select **SSL enabled** to enable SSL communication between the LDAP server and WebSphere Application Server. Click **Centrally managed** to defer the selection of the SSL configuration to the server-wide endpoint management scheme. Otherwise, click **Use specific SSL alias** and select a configuration scheme from the drop-down list.

8. Click **OK** and save the changes. If you are in a WebSphere Application Server Network Deployment environment, select **Synchronize changes with Nodes** before you save the changes.
9. Select **Test connection** to check the validity of the specified information. Otherwise, the validity of the information is not confirmed until this registry is selected as the current repository.

What to do next

To enable security in a WebSphere Application Server Network Deployment environment by using an LDAP user registry, you must configure LTPA as the active authentication protocol to authenticate the users. See “Configuring the LTPA authentication mechanism” on page 162 for instructions.

Configuring a custom registry:

A custom registry is any registry that implements the `com.ibm.websphere.security.UserRegistry` interface. From the WebSphere Application Server administrative console, configure the custom registry settings.

Procedure

1. Log on to the WebSphere Application Server administrative console.
2. Click **Security** > **Global security** to display the "Global security" panel.
3. In **User account repository** > **Available realm definitions**, select **Standalone custom registry** from the drop-down list.
4. Click **Configure**.
5. Specify the **Primary administrative user name** property, which is a name of a user in your custom registry who has administrative privileges.
6. Select one of the following options:
   - Click **Automatically generated server identity**.
   - Click **Server identity that is stored in the repository** and specify the following properties:
     - Server user ID (for example, `root`), which is the operating system user ID that the application server is using for security purposes.
     - Server user password (for example, `abc26xyz`).
7. Specify the **Custom registry class name** property (for example, `com.ibm.websphere.security`)
8. Enable the **Ignore case for authorization property** if your custom class requires it.
9. Click **OK** and save your changes. If you are in a WebSphere Application Server Network Deployment environment, select **Synchronize changes with Nodes** before you save the changes.

Configuring a federated registry:

A federated registry allows consolidation of multiple repositories into a single virtual registry.

About this task

From the WebSphere Application Server administrative console, configure the federated registry settings:

Procedure

1. Log on to the WebSphere Application Server administrative console.
2. Click **Security** > **Global security** to display the "Global security" panel.
3. In **User account repository** > **Available realm definitions**, select **Federated repositories** from the drop-down list.
4. Click **Configure**.
5. Specify the **Realm name**.
6. Specify the **Primary administrative user name property**, which is the name of a user in one of the federated registries who has administrative privileges.
7. Select one of the following options:
   - Click **Automatically generated server identity**.
   - Click **Server identity that is stored in the repository** and specify the following properties:
     - Server user ID (for example, `root`). This value must be a valid user ID in the local operating system registry.
     - Server user password (for example, `abc26xyz`).
8. Optional: Enable the **Ignore case for authorization** property if case sensitivity is not important.
9. Optional: Enable the **Allow operations if some of the repositories are down** property if you want operations to continue when a repository is down.
10. Use the **Repositories in the realm** table to manage the repositories that you want federated. Optionally, you can add the built-in file repository and any external LDAP registries.
11. Click **OK** and save your changes. If you are in a WebSphere Application Server Network Deployment environment, select **Synchronize changes with Nodes** before you save the changes.

Configuring the security policy for the JDBC provider:

A JDBC provider JAR file and directory path are configured to access the DB2 Audit Database (XML data store). If Java 2 security is enabled, this JAR file must be granted access permissions before it is configured into the WebSphere Application Server by using the configuration console.

About this task

To grant access permissions to the JAR file, add the appropriate grant policy to the `app.policy` file that is in the target node. For a cluster configuration, the target node is the Deployment Manager node; for a stand-alone server, the target node is the server parent node.

Procedure

1. Start the **`wsadmin`** tool of the target profile for which you want to configure a Common Audit Service server instance
2. Extract the policy file to a temporary location with the following command:

   ```
   wsadmin>set obj [$AdminConfig extract
    cells/cell_name/node/node_name/app.policy
   temp_file_path/app.policy]
   ```

   If you are using Deployment Manager, set the *node_name* value for a CellManager node.

   Ensure that *temp_file_path* exists before running the command.

Use either a single forward slash (/) or a double back slash (\\) while you specify the path on Windows, do not use a single back slash (\) to specify the path.

3. In a separate shell, run the Policy Tool to edit the extracted `app.policy` file. See "Editing the app.policy file using the Policy Tool" for detailed instructions on using the Policy Tool. The following list summarizes the changes that you must make.
   a. Create a policy with codeBase "`file:`*`JDBC_driver_path`*`/db2jcc.jar`".
   b. Add the permission "`AllPermission`".
   c. Save the policy file.
4. Check the policy file back into the WebSphere Application Server node with the following command:

```
wsadmin>$AdminConfig checkin cells/cell_name/node/node_name/app.policy
temp_file_path/app.policy &obj
```

Editing the app.policy file using the Policy Tool:

The Policy Tool is a utility to enable editing of Java policy files, such as `app.policy`. Update the policy for the JDBC provider.

Procedure

1. Start the Policy Tool with the following command:
   - On AIX, Linux, and Solaris platforms: *`was_install_root`*`/java/jre/bin/policytool`
   - On Windows platforms: *`was_install_root`*`\java\jre\bin\policytool.exe`

   The tool looks for the `java.policy` file in the home directory. If it does not exist, an error message is displayed.
2. To dismiss the error, click **OK**.
3. Click **File-> Open**.
4. Navigate the directory tree in the **Open** window to the temporary file *`temp_file`*`/app.policy`. Select the file and click **Open**. The existing code base entries are listed in the window.
5. Create a code base entry by clicking **Add Policy Entry**.
6. In the **Policy Entry** window, in the code base column, add the string `file:`*`JDBC_driver_path`*`/db2jcc.jar`, where *JDBC_driver_path* represents the path to your JDBC driver. Use a forward slash (/) to specify *JDBC_driver_path*.
7. Click **Add Permission** to add the permission for the JDBC driver.
8. In the permissions window, select the **AllPermission** entry in the drop-down list.
9. Click **OK**.
10. In the **Policy Entry** window, the string `permission java.security.AllPermission` is displayed beneath the Permission buttons. Click **Done**.
11. Click **File-> Save** to save the updated file.
12. Click **File-> Exit** to exit the tool.

Configuring the LTPA authentication mechanism:

From the WebSphere Application Server administrative console, configure Lightweight Third-Party Authentication (LTPA) token authentication.

Procedure

1. Click **Security** > **Global security** > **LTPA**.
2. In the **Key Generation** area, **Key set group** option, complete the following steps:
   a. Select the **NodeLTPAKeySetGroup** key set group.
   b. Click **Generate keys**.
3. Under **Authentication expiration**, specify the following properties:
   - Authentication cache timeout
   - Timeout value for forwarded credentials between servers (for example, 120)
4. Click **OK** and save the changes. If you are in a WebSphere Application Server Network Deployment environment, select **Synchronize changes with Nodes** before you save the changes.
5. Click **Global security**.
6. Click **OK** and save the changes. If you are in a WebSphere Application Server Network Deployment environment, be sure to select **Synchronize changes with Nodes** before you save the changes.

Restarting the cluster:

From the WebSphere Application Server administrative console, restart the cluster to enable the global security method.

Procedure

1. Expand **Servers**.
2. Click **Clusters** and select the target cluster.
3. Click **Stop**.
4. Stop and restart the deployment manager system with the WebSphere Application Server security enabled method.
5. Stop and restart the agent on the managed nodes with the WebSphere Application Server security enabled method.
6. Expand **Servers**.
7. Click **Clusters** and select the target cluster.
8. Click **Start**.

What to do next

From this point on, use the WebSphere Application Server security enabled method for stopping and starting the Deployment Manager and managed nodes.

Configuring SSL

This topic describes how to configure SSL for securing Web service client communications.

Following are three ways you can configure SSL:

- Configure WebSphere Application Server for SSL.
- Configure SSL communication between the IBM HTTP Server plug-in and the WebSphere Application Server.
- Configure the IBM HTTP Server for SSL (required in a clustered environment).

Configuring WebSphere Application Server for SSL:

Configure WebSphere Application Server Secure for Secure Sockets Layer (SSL) authentication.

Procedure

1. Create an SSL configuration entry:
   a. Click **Security** > **SSL certificate and key management**.
   b. Click **SSL Configuration** from the Related Items list.
   c. Click **New** to create an SSL configuration specifically for Common Auditing Service.
   d. Specify **Name** as `CARSSSLConfiguration`.
   e. Specify **Trust store name** (for example, `CellDefaultKeyStore`).
   f. Specify **Keystore name** (for example, `CellDefaultKeyStore`).
   g. Click **Get certificate aliases**.
   h. Specify **Default server certificate alias** (for example, as default).
   i. Specify **Default client certificate alias** (for example, as default).
   j. Click **OK** and save the changes. If you are in a WebSphere Application Server Network Deployment environment, select **Synchronize changes with Nodes** before you save the changes.
2. Configure SSL between the WebSphere Application Server and the Web service client. To do this configuration, assign an SSL configuration to a WebSphere Application Server configuration scope that enables the port for encryption and decryption of inbound data.
   a. Click **Security** > **SSL certificate and key management** > **Manage endpoint security configurations**.
   b. In the inbound local topology tree, click the cluster or server name into which Common Auditing Service is being deployed.
   c. Under **Specific SSL configuration for this endpoint**, enable **Override inherited values**.
   d. Select **CARSSSLConfiguration** from within the SSL configuration field.
   e. Click **Update certificate alias list**.
   f. Specify the certificate alias in the keystore from the drop-down list (for example, default).
   g. Click **OK** and save the changes.
   h. Click **Security** > **SSL certificate and key management**.
   i. Select to dynamically update the run time when SSL configuration changes occur.
   j. Click **Apply** and save the changes. If you are in a WebSphere Application Server Network Deployment environment, select **Synchronize changes with Nodes** before you save the changes.

Configuring the Web server plug-in for SSL:

This topic describes how to set SSL security for communication between the Web server and a WebSphere Application Server Web server plug-in.

About this task

The Web server plug-in can be enabled to securely communicate with the corresponding Web server, which might be critical because the Web server is usually remote to at least some of the nodes in the cluster. See “Configuring a Web server that is installed on a system outside the cluster" on page 45 for information about configuring the Web server plug-in.

Take these steps to implement security by using the SSL protocol for communication between the Web server and the Web server plug-in that is configured in a WebSphere Application Server cluster.

Procedure

1. From the WebSphere Application Server Administrative Console, click **Servers->Web servers**.
2. Select the Web server name.
3. Click **Plug-in properties**.
4. Under **Repository copy of Web server plug-in files**, specify the keystore file name (or accept the default name) in the following field:

   Plug-in keystore file name
   Stores the cryptographic keys for the plug-in. The default value is `Plugin-key.kdb`.
5. Under the Web server copy of Web server plug-in files, specify the keystore filepath in the following field:

   Plug-in keystore directory and file name
   The filepath is *`WAS_HOME`*`/Plugins/config/`*`webserver_name`*`/Plugin-key.kdb`.
6. Click **Manage keys and certificates** to access configuration options for your keys and certificates. By default, you can change the password that you use to protect the keystore.
7. Click **Apply** to save the password changes.
8. Under **Additional Properties**, you can also select the following options:

   Signer certificates
   Use this option to add new certificates, delete certificates, extract certificates, and to retrieve certificates from a port.

   Personal certificates
   Use this option to create a new self-signed certificate, delete a certificate, or to import and export a personal certificate.

   Personal certificate requests
   Use this option to manage personal certificate requests.

   Custom properties
   Use this option to define custom properties for the keystore.
9. Click **Personal certificates** and confirm that at least one personal certificate is in the keystore.
10. Click **Servers-> Web servers->** *webserver_name***-> Plug-in properties-> CMSKeyStore->Copy to Web server key store directory** to copy the keystore and to stash the files to a managed Web server.

Configuring the IBM HTTP Server for SSL:

This topic describes how to configure the IBM HTTP Server for SSL. SSL is required in a WebSphere Application Server clustered environment.

Before you begin

The Common Auditing Service web service client can start the Common Auditing Service either directly by talking to the WebSphere Application Server embedded HTTP server, or indirectly by first going through a web server. The web server can be the IBM HTTP Server or another third-party web server. The web server must be enabled for SSL for secure communication with the client. SEe the appropriate web server documentation for details on how to enable SSL.

Procedure

1. Use the IBM HTTP Server **IKEYMAN** utility to create a CMS key database file and insert the personal certificate of the server.

   For example, to create a CMS key database file, open the `CARSServerKey.jks` file in iKeyman and then save it as a CMS file. Copy the `CARSServerKey.kdb` and `CARSServerKey.sth` files to a directory on the HTTP server (for example, `/data/certs`).
2. Modify the `httpd.conf` file.

   For the IBM HTTP Server to support HTTPS, you must enable SSL on the IBM HTTP Server. You can modify the configuration file of IBM HTTP Server, which is *IHS_HOME*`/conf/httpd.conf`. *IHS_HOME* is the home directory of your IBM HTTP Server. Open the *IHS_HOME*`/conf/httpd.conf` file and add the following lines to the bottom of the file. This example uses port 443.

   ```
   LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
   <IfModule mod_ibm_ssl.c>
     Listen 443
     <VirtualHost *:443>
      SSLEnable
      SSLClientAuth none
      SSLServerCert certname
   </VirtualHost>
   </IfModule>
   SSLDisable
   Keyfile /data/certs/CARSServerKey.kdb
   ```

   Note: The SSLServerCert *certname* is the label of the server certificate in the key database file. If the default certificate in the keyfile is used, it is not needed. Change the host name and the path for the key file accordingly.

   You can also use the administrative console to enable SSL.
3. Restart the IBM HTTP Server.
4. Add the port number to the virtual host.

   To enable the application server to communicate with the IBM HTTP Server using, for example, port 443, add the host alias on the default_host. In the administrative console:
   a. Click **Environment** > **Virtual Hosts** > **default_host**.
   b. Under **Additional properties**, click **Host Aliases** > **New**.
   c. Enter the following information in the fields:
      - Type `*` for **Host Name**.
      - Type `443` for **Port**.
   d. Click **Apply** and **Save**. When you click **Save**, the information is written to the security.xml file and the Web server plug-in. For example, `/opt/IBM/WebSphere/Plugins/config/webserver1_hostname/plugin-cfg.xml` is automatically updated.
5. Enable security on your installed Web server.

   a. Click **Servers** > **Web servers** > *your_web_server* > **Global directives**.
   b. Under **Global Directives** specify the following information:
      - Select **Security enabled**.
      - Enter `CARSWebStore` in **Key store certificate alias**.
      - Enter `*:443` in **Listen ports**.
   c. Click **Apply** and **Save** to enable port 443 for listening on the Web server.
6. Stop and restart the IBM HTTP Server and IBM HTTP Administrative Server.
7. Stop and restart WebSphere Application Server. In a clustered environment, stop and restart the cluster.

Mapping Common Audit Service security roles

Common Auditing Service defines three roles that are called EventSource, eventAdministrator, and eventCreator.

About this task

The EventSource role defines the source of events to be submitted to Common Auditing Service. The eventAdministrator role provides access to the EventAccess and Emitter interfaces. The eventCreator role restricts access to event submission. For more information about configuring the use of security roles, see the WebSphere Process Server Information Center.

Map roles to a user or group with the WebSphere Application Server Administration Console.

Procedure

1. Click **Applications** > **Enterprise Applications** > **CommonAuditService** > **Security role to users/group mappings**.
2. Select the **EventSource** role.
3. Click one of the following options:
   - **Look up users** to map users
   - **Look up groups** to map groups
4. Click **Search** to display the available users or groups list.
5. Select the users or groups from the list and click **>>** to move them to the selected list. For example, if the user registry is Local Operating System, then select `root`.
6. Click **OK** to add the selected users or groups to the **Mapped users** or **Mapped groups** list.
7. Repeat steps 2 through to 6 to add the eventAdministrator and eventCreator roles. Add other users and groups as needed.
8. Click **OK** and then save the changes. If you are in a WebSphere Application Server Network Deployment environment, select **Synchronize changes with Nodes** before you save the changes.

What to do next

The All-Authenticated and Everyone meta-groups override any mapped users or groups. Ensure that you clear these meta-groups when you map a specific user or group.

Securing the XML data store

The XML data store contains audit information and, therefore, must be secured by using the security mechanisms that are provided by DB2 so that only authenticated users with correct privileges access the data.

Chapter 18. Archiving and restoring audit data

This topic describes how to archive and restore your Common Auditing and Reporting Service audit data.

Over time, as security events continue to be stored into the XML data store, available disk space will be consumed. Periodically, older security event data must be purged from the XML data store to free disk space to hold more events. The frequency by which older event data must be purged will depend on a combination of available disk space, event submission rates, and also requirements for how long data must be readily available for interactive reporting.

Archiving audit data

In most environments, security event data must be archived to mass storage media before being purged from the database. To archive security event data from the XML data store, use third-party database archive tools with the Common Auditing Service XML data store utilities.

Use the following XML data store utilities immediately before archiving (prearchive) and after successful archiving (postarchive):

- The XML data store utilities prearchive operation identifies the names of the XML audit and overflow tables within the XML data store database that are inactive and ready to archive. The prearchive operation also identifies the date range of security events in the table for future reference.
- The XML data store utilities postarchive operation purges data from the inactive XML event and overflow tables in the XML data store database, and then activates those tables so that new events can be stored in them.

Note: The first time that you run the prearchive operation after you install the audit server, no data will be in the inactive XML event tables. Run the postarchive operation to prepare data to be archived for the first time.

Here is an example of XML data store utility prearchive operation output:

```
[/opt/IBM/Tivoli/CommonAuditService/server/etc] java com.ibm.cars.xmlstoreutils.XmlStoreUtils
  -operation prearchive -dbpassword tivoli
  -configurl FILE:////opt/IBM/Tivoli/CommonAuditService/server/etc/ibmcars.properties

CBAXU0207I The name of the XML event store table to archive is cei_t_xml00

 The name of the XML event store overflow table to archive is cei_t_xmlx00

 The first date contained in the table set to be archived is Jul 27, 2007 12:04:45 AM GMT

 The last date contained in the table set to be archived is Jul 27, 2007 12:07:05 AM GMT

CBAXU0220I XmlStoreUtility has exited successfully.
```

Figure 7. Sample prearchive output

Detailed information about how to use the XML data store utilities is in “Running the XML data store utilities” on page 67.

Restoring audit data

If archived audit event data is needed for investigation or reporting purposes, it can be restored into the XML event restore and overflow restore tables in the XML data store (database).

The XML event restore table is always named cei_t_xmlre. The overflow restore table is always named cei_t_xmlxre. The staging utility stages events from these restore tables into the report tables that are used for reporting, just as it stages events from the original XML event and overflow tables.

After the restored data is no longer needed, use the XML data store utilities cleanrestore operation to purge the XML event restore and overflow restore tables.

Here is the schema of the XML event tables and the XML event restore tables:

```
CREATE TABLE (XML event table name)
(
  record_id         BIGINT       not NULL,
  version           VARCHAR(16)  not NULL,
  creation_time_utc BIGINT       not NULL,
  global_id         VARCHAR(64)  not NULL,
  extension_name    VARCHAR(192) ,
  is_compressed     CHAR(1)      not NULL with DEFAULT 'N',
  has_overflowed    CHAR(1)      not NULL with DEFAULT 'N',
  xml_data          VARCHAR(7793) for bit data
) in cei_ts_8k;

ALTER TABLE (XML event table name)
  ADD CONSTRAINT xml_record_pk00 PRIMARY KEY
  (record_id);
```

Here is the schema of the overflow and overflow restore tables:

```
CREATE TABLE (overflow table name)
(
  record_id         BIGINT      not NULL,
  xml_data          BLOB(1G)    not NULL
) in cei_ts_base4K_path;

ALTER TABLE (overflow table name)
  ADD CONSTRAINT xml_record_fk00 FOREIGN KEY
  (record_id) REFERENCES cei_t_xml00
    ON DELETE CASCADE;
```

Detailed information about how to use the XML data store utility is in “Running the XML data store utilities” on page 67.

Part 5. Native Security Access Manager auditing

Chapter 19. Audit event logging

To enable logging, define entries in the configuration file.

Procedure

1. Specify the type of audit event.
2. Specify the location of the audit log.
   Note: On Windows operating systems, newly created files are given "Full Control" permissions or inherit permissions from the parent directory. To protect audit files from possible tampering, manually modify the permission settings to "Read & Execute" on newly created files and on any parent directory.
3. Specify the maximum file size.
4. Specify the file flush interval.

Log agents

With event logging, the concept of a *log agent* includes capturing events that are redirected to destinations other than the local file system. Event logging uses the following types of log agents, each agent represents an audit trail:
- "Sending events to the console" on page 176
- "Configuring file log agents" on page 177
- "Configuring pipe log agents" on page 183
- "Configuring remote log agents" on page 185

Configuring audit events

Independent of the logging agent, configure which audit events to capture by using the `logcfg` entry.

When using the Security Access Manager approach, define the `logcfg` entry in any or all the following locations:

- The `[ivmgrd]` stanza of the policy server `ivmgrd.conf` configuration file
- The `[ivacld]` stanza of the authorization server `ivacld.conf` configuration file
- The `[aznapi-configuration]` stanza of a WebSEAL server `webseald.`*`instance`*`.conf` configuration file
- The `[aznapi-configuration]` stanza of the Plug-in for Web Servers `pdwebpi.conf` configuration file
- The `[aznapi-configuration]` stanza of the resource manager `aznAPI.conf` configuration file

Defining logcfg entries

When you define the `logcfg` entry in a configuration file, use the following general format (on a single line) to specify audit event logging:

```
logcfg = category:{stdout|stderr|file|pipe|remote}
[[parameter[=value]],
[parameter[=value]]],
...,
[parameter[=value]]]
```

To enable the recording of audit events, associate an event category with a log agent (file, pipe, or remote) or associate an event category with a console destination (stdout or stderr).

When you define the parameters for any logcfg entry, be aware of the following conditions:

- Parameters can be specified in any sequence
- Parameter names are not case-sensitive
- Parameter names can be shortened to any unambiguous name
- Parameters differ by log agent
- Parameters are optional

Events for a category are inclusive of all subcomponents in the hierarchy. That is, a foo.bar.fred event is captured when the foo.bar category is defined.

You can attach multiple log agents to the same category. For example, the following configuration:

- Captures authorization audit events (category audit.azn) and uses a file agent to copy these events to the audit.azn file.
- Uses a pipe agent to relay these same events to the analyse.exe program.

```
[ivacld]
logcfg = audit.azn:file path=/var/PolicyDirector/log/audit.azn
logcfg = audit.azn:pipe path=/bin/analyse.exe
```

Parameters for the logcfg entry

The available parameters for the logcfg stanza entry differ by log agent.

Table 42 shows which parameters are available for the EventPool category and the following log agents:

- File log agent
- Pipe log agent
- Remote agent
- Remote syslog agent

Table 42 does not show the console log agent. The console log agent does not support parameters. For more information, see “Sending events to the console” on page 176.

Table 42. Available parameters for the logcfg stanza entry

Parameter	EventPool category	File log agent	Pipe log agent	Remote log agent	Remote syslog agent
buffer_size		Yes		Yes	
compress				Yes	
dn				Yes	
error_retry				Yes	Yes
flush_interval	Yes	Yes	Yes	Yes	Yes
hi_water	Yes	Yes	Yes	Yes	Yes
log_id		Yes			Yes
max_event_len					Yes
mode		Yes			

Table 42. Available parameters for the logcfg stanza entry (continued)

Parameter	EventPool category	File log agent	Pipe log agent	Remote log agent	Remote syslog agent
path		Yes	Yes	Yes	Yes
port				Yes	Yes
queue_size	Yes	Yes	Yes	Yes	Yes
rebind_retry				Yes	Yes
rollover_size		Yes			
server				Yes	Yes
ssl_keyfile					Yes
ssl_label					Yes
ssl_stashfile					Yes

Configuring the event pool

Events are passed to subscribed log agents asynchronously from the application-level requests that construct the events. All events enter the common propagation queue before they are forwarded to the subscribed log agents.

The propagation queue is configurable. To configure the propagation queue, define the `logcfg` stanza entry with `EventPool` as the category name and specifies the configuration parameters without specifying a log agent.

Manage the propagation queue to support the configuration of log agents. For example, limit the amount of memory that is used to queue events for a remote log agent. To limit the amount of memory that is used, constrain the propagation queue with the **queue_size** parameter:

```
[aznapi-configuration]
logcfg = EventPool queue_size=number,hi_water=number,
     flush_interval=number_seconds
logcfg = category:remote buffer_size=number,path=pathname,
     server=hostname,queue_size=number
```

You can define the following parameters for pipe log agents:

flush_interval

Configure the **flush_interval** parameter to limit the amount of time that events can remain in the propagation queue. Specify the time in seconds. Assume that the size of the queue does not reach the high water mark within the specified interval. In this case, events in the queue are forwarded to the log agents.

The default value is `10` seconds. Specifying a value of `0` is equivalent to setting the value to `600` seconds.

hi_water

Configure the **hi_water** parameter to indicate the threshold where events in the propagation queue are forwarded to the log agents. Assume that the size of the queue does not reach this high water mark within the defined flush interval. In this case, events in the queue are forwarded to the log agents.

The default value is calculated as two-thirds of the configured queue size. If the queue size is 0 (unlimited), the high water mark is set to 100 events. If the high water mark is 1 event, each event in the queue is forwarded immediately to the log agents.

Setting a low value for the high water mark can have an adverse effect on performance. For more information, see the *IBM Security Access Manager for Web Performance Tuning Guide*.

queue_size

Because each event in the propagation queue consumes memory, configure the **queue_size** parameter to define the maximum number of events that the propagation queue can hold. If the maximum size is reached, the event-producing thread is blocked until space is available in the queue. Blocking corresponds to throttling back the performance of the event-producing thread to a rate that can be consumed by the logging threads.

The default value is 0. Specifying a value of 0 indicates that no size limit is enforced on the propagation queue. The propagation queue can grow to an unmanageable size when:

- You use the default value, and
- The logging threads cannot process events as they enter the propagation queue.

Sending events to the console

Logging to the console is the easiest event logging option to configure. Associate an output destination of standard out or standard error with the category of events in the event pool to capture:

```
[ivmgrd]
logcfg = category:{stdout|stderr}
```

Logging to the console does not use any queuing. The events are written to the console as they are received from the propagation queue. Depending on the queue settings, events might be delayed in the propagation queue.

If you are using console output and running a server in the foreground for debugging purposes, you might want to set the propagation queue settings accordingly. For example, set the **hi_water** parameter to a low value.

Sending events to standard error

You might configure event logging to standard error.

Procedure

1. Edit the appropriate server configuration file. Each server provides its own stanza entry values in its configuration file.
2. Locate the stanza that contains the `logcfg` entries.
3. Define the `logcfg` entry and specify the event category to log and the destination of standard error.

   ```
   logcfg = category:stderr
   ```
4. Save and exit the configuration file.

Example

For example, to capture all audit events to standard error, define the following entry in the configuration file:

```
[ivmgrd]
logcfg = audit:stderr
```

Sending events to standard output

You might capture event logging to standard output.

Procedure

1. Edit the appropriate server configuration file. Each server provides its own stanza entry values in its configuration file.
2. Locate the stanza that contains the `logcfg` entries.
3. Define the `logcfg` entry and specify the event category to log and the destination of standard output.
   ```
   logcfg = category:stdout
   ```
4. Save and exit the configuration file.

Results

To capture all audit events to standard output, define the following entry in the configuration file:

```
[ivmgrd]
logcfg = audit:stdout
```

Configuring file log agents

To record events in a file, specify a log file configuration as follows:

```
[ivacld]
logcfg = category:file path=file_pathname, flush_interval=num_seconds,
     rollover_size=number,log_id=logid,queue_size=number,
     hi_water=number,buffer_size=number,mode={text|binary}
```

Parameter names can be shortened to any unambiguous name. For example, the **hi_water** parameter can be shortened to `hi`.

A file is opened only one time. The file opens according to the options in the first configuration entry that is processed when:

- Multiple configuration entries exist.
- You want to selectively capture events to the same file.
- You want to capture events at different points of the event pool hierarchy.

After a file was opened, further file configurations can use the following shorthand notation to record events to the same file:

```
[ivacld]
logcfg = category:file log_id=logid
```

Writing to a file can be a slow operation relative to the tasks that are generating events. Therefore, events are posted to a file log agent through a second level of queuing. This second level of event queuing is configured like the central event propagation queue, but has different default values.

Parameters for file log agents

You can define the following parameters for file log agents:

buffer_size

Reduce memory fragmentation and improve the performance of writing to a file by:

- Not queuing many small events individually to the file log agent.
- Buffering events into blocks of a nominated size before queuing for writing.

The **buffer_size** parameter specifies the maximum size message that the program attempts to construct by combining smaller events into a large buffer.

Buffers consist of only an integral number of events; events are not split across buffers. If any individual event exceeds that maximum configured size, the large event is recorded in a buffer of its own, exceeding the configured value. The default buffer size for logging to a file is `0` bytes. This value prevents buffering and each event is handled individually.

If a value is specified for the **buffer_size** parameter, events are packed into buffers of that size before queuing to the file log agent.

For example, around 10 events are packed into each buffer that is written to the file when:

- The value for the **buffer_size** parameter is set to `2` KB.
- Events are assumed to be about 256 bytes.

This process reduces the number of disk input/outputs (I/Os) that are made while logging to 10 percent of the equivalent non-buffering case.

A default queue size of 200 also consumes around 10 times the memory of a default configuration that did no buffering if:

- The buffer size was 2 KB.
- The event size was around 200 bytes.

This size is because the maximum queue size value has not been changed. However, the size of events being queued has increased tenfold.

flush_interval

The **flush_interval** parameter is a multiuse parameter.

Ensure that stream buffers are flushed to disk regularly. Configure the frequency with which the server asynchronously forces a flush of the file stream to disk. To configure this frequency, use the **flush_interval** parameter. The value that is defined for this parameter is `0`, `< 0`, or the flush interval in seconds.

Specifying a value of `0` results in the flushing of the buffer every 600 seconds.

Specifying a value of `< 0` results in the absolute value that is used as the asynchronous flush frequency. However, a stream flush is also forced synchronously after each record is written.

Events are consolidated into large buffers that is based on the value of the **buffer_size** parameter. However, the **flush_interval** parameter also might affect the size of buffer written. When a flush is scheduled, an in-memory, partially filled buffer is also queued for writing before it completes the buffer fill.

The event queue is triggered for processing at the flush interval rate. The trigger enables processing of events that were waiting for longer than the scheduled flush time. Such processing applies to a scenario when the queue does not reach the high water mark between scheduled flushes.

hi_water

Processing of the event queue is scheduled regularly at the configured flush interval. It also is triggered asynchronously by the queue size that reaches a high water mark on the event queue.

The default value is two-thirds of the maximum configured queue size. If the maximum queue size is zero, the high water mark is set to a default of `100`.

The transaction rates and the values of these options determine the maximum amount of memory that is consumed by enabling event logging to file.

If the event queue high water mark is set to 1, every event queued is relayed to the log agent as soon as possible. This setting is not optimal. Use it if you want to ensure that events get to disk as fast as possible. Doing so adversely impacts overall performance.

log_id

An open log file is associated with a short name identifier to facilitate the recording of events from different categories to the same file.

Use the **log_id** parameter to set the log file identifier (ID) explicitly; otherwise, it is given a default value. If the **path** parameter is specified, the default value is the configured path name. If the **path** parameter is not specified, the log ID defaults to the domain component of the event category being captured. For example:

```
logcfg = audit.azn:file
```

implies

```
log_id=audit
```

To capture events to a common file, set the log file ID to a suitable value in a fully optioned file configuration. Then, use the shorthand configuration variant to capture events from additional categories as shown:

```
[aznapi-configuration]
logcfg = audit.azn:file path=/opt/PolicyDirector/log/audit.log,
     rollover_size=-1,flush_interval=20,log_id=audit,
     ...
logcfg = audit.authn:file log_id=audit
```

Because of the default rules, this configuration is also equivalent to the following specification:

```
[aznapi-configuration]
logcfg = audit.azn:file path=/opt/PolicyDirector/log/audit.log,
     rollover_size=-1,
     ...
logcfg = audit.authn:file
```

If you construct a configuration where the log ID value does not match any open log file, no events are captured. For example, the following configuration does not record any events because the configuration line that initializes the log file was commented out:

```
[ivacld]
#logcfg = audit.azn:file path=/tmp/azn.log,log_id=azn,...
logcfg = audit.authn:file log_id=azn
```

mode

Configure the **mode** parameter to open a file in either text or binary mode. For example:

```
[aznapi-configuration]
logcfg = audit.azn:file
...
mode={text|binary},
...
```

Text mode is deprecated on AIX, Linux, and Solaris operating systems. Binary mode on a Windows operating system writes the log file in an AIX, Linux, or Solaris-compatible format.

path

The path specifies the name and location of a log file. There is no default value, because the value of the **log_id** parameter takes precedence. An example for the WebSEAL audit trail file on AIX, Linux, and Solaris operating systems is as follows:

```
[aznapi-configuration]
logcfg = category:file path=/var/pdweb/log/audit.log
```

The directory portion of this path must exist. The log file is created if it does not exist.

queue_size

There is a delay between events being placed on the queue and the file log agent removing them. The **queue_size** parameter specifies the maximum size to which the queue is allowed to grow.

Consider that a new event is ready to be placed on the queue. Then, if the queue reaches the maximum size, the requesting thread is blocked until space is available in the queue. This process causes the performance of the event propagation thread to slow down to that of the file logging thread. Limiting the queue size for the log agent must be configured with setting the queue size for the central event propagation queue. Unless the event propagation defined by the **queue_size** parameter is constrained appropriately, memory usage can still grow without bounds.

```
[aznapi-configuration]
logcfg = audit.azn:file
...
queue_size=number_events,
...
```

The default value is 0. Specifying a value of 0 indicates that no limit is enforced on the growth of the unprocessed event queue. Correspondingly, the event propagation thread is not constrained by the speed of the logging thread. The unrecorded event queue can grow to an unmanageable size if:

- You are using the default.
- Events are being generated faster than they can be recorded to file.

rollover_size

Configure the **rollover_size** parameter to specify the maximum size to which a log file can grow. The default value is 2000000 bytes.

When the size of a log file reaches the specified rollover threshold, the existing file is backed up. The back-up happens to a file of the same name with the current date and time stamp appended. A new log file is then started.

The possible rollover size values are interpreted as follows:

- If the **rollover_size** value is less than zero, a new log file is created:
  - With each invocation of the process, and
  - Every 24 hours since that instance.
- If the **rollover_size** value is equal to zero, the log file grows until it reaches 2 GB and then rolls over. If a log file exists at startup, new data is appended to it.
- If the **rollover_size** value is greater than zero, the log file grows until it reaches the lesser of the following values and then rolls over:
  - The specified value
  - 2 GB

  If a log file exists at startup, new data is appended to it.

Sending events to a log file

You might configure Security Access Manager to send event records to a log file.

Before you begin

Before you begin this task, review the information in "Configuring file log agents" on page 177.

Procedure

1. Edit the appropriate server configuration file. Each server provides its own stanza entry values in its configuration file.
2. Locate the stanza that contains the `logcfg` entries.
3. Specify that the category is to send event records to a log file by using the following format:

   ```
   category:file
   ```

   For example, a category might be to audit authorization events (`audit.azn`):

   ```
   logcfg=audit.azn:file
   ```
4. Specify the path to the log file:

   ```
   path=fully_qualified_path
   ```

   The default directories are:

   AIX, Linux, and Solaris operating systems

   `/opt/PolicyDirector/log`

   Windows operating systems

   `C:\Program Files\Tivoli\Policy Director\log\`

   The default file name depends on the type of logging being completed, such as `audit.log`
5. Specify the identifier for the log file:

   ```
   log_id=logid
   ```

   Use the **log_id** parameter to set the log file identifier (ID) explicitly; otherwise, it is given a default value. If the **path** parameter is specified, the default value is the configured path name. If the **path** parameter is not specified, the log ID defaults to the domain component of the event category that is being captured. For example, `logcfg=audit.azn:file` implies `log_id=audit`.

6. Specify the maximum size of the log file:

   ```
   rollover_size= value
   ```

   By default is `rollover_size=2000000`.

   The rollover size values are interpreted as:

   - If less than zero, a new log file is created with each invocation of the process and every 24 hours from that instance.
   - If equal to zero, no rollover is completed, and the log file grows indefinitely. If a log file exists, new data is appended to it.
   - If greater than zero, a rollover is completed when a log file reaches the configured threshold value. If a log file exists at startup, new data is appended to it.
7. Specify the maximum size of the buffer:

   ```
   buffer_size={0|number_kb}
   ```

   By default, the buffer size for logging to a file is `0` bytes, This buffer size prevents buffering so that each event is handled individually. If a value other than `0` is specified, events are packed into buffers of that size before queuing to the file log agent.

   Buffers consist of only an integral number of events; events are not split across buffers. If any individual event exceeds that maximum configured size, the large event is recorded in a buffer of its own, exceeding the configured value.
8. Specify the maximum number of events to queue in memory:

   ```
   queue_size={0|number_events}
   ```

   By default, the queue size is `0`. A zero queue size means that no limit is enforced on the growth of the unprocessed event queue. The requesting thread is blocked until space is available in the queue when:

   - The **queue_size** is defined as any valid value except `0`.
   - The number of events in the queue reaches the defined queue size.
   - A new event is ready to be placed on the queue.
9. Specify the event queue high water mark:

   ```
   hi_water={0|1|number}
   ```

   By default, the event queue high water mark value is two-thirds of the maximum configured queue size.

   If the maximum queue size is `0`, the high water mark is set to a default of `100`. The transaction rates and the values of these options determine the maximum amount of memory that is consumed by enabling event logging to file.

   If the event queue high water mark is set to `1`, every event queued is relayed to the log agent as soon as possible. This setting is not optimal.
10. Specify the frequency for flushing log file buffers:

    ```
    flush_interval={0|number_seconds}
    ```
11. Specify the file mode:

    ```
    mode={text|binary}
    ```

    Binary mode on a Windows operating system writes the log file in an AIX, Linux, or Solaris-compatible format.

    Text mode is deprecated on AIX, Linux, and Solaris operating systems.
12. Save and exit the configuration file.

Example

For example, to configure a file log agent to capture authorization events, the following sample shows the `logcfg` entry:

```
[aznapi-configuration]
logcfg=audit.azn:file path=/opt/PolicyDirector/log/audit.log,
flush_interval=20,rollover_size=2000000,log_id=audit,queue_size=200,
hi_water=100,buffer_size=2,mode=text
```

Tuning the buffer size with the queue size and the event queue high water mark can improve performance.

Configuring pipe log agents

Configure the **pipe** parameter to write output to the standard input of another program. For example:

```
[aznapi-configuration]
logcfg = category:pipe path=program_pathname,
  queue_size=number, hi_water=number, flush_interval=number_seconds
```

Parameter names can be shortened to any unambiguous name. For example, the **hi_water** parameter can be shortened to `hi`.

The named program must exist and must be an executable program. The administrator is responsible for ensuring the security of the program that is to be run.

Each occurrence of a pipe agent in the configuration file starts a new copy of the pipe program. Unlike logging to file, piped events are not multiplexed from different capture points to a single copy of the program.

Parameters for pipe log agents

You can define the following parameters for pipe log agents:

flush_interval

Configure the pipe log agent in the same way that you configure file log agents. The **flush_interval** parameter has similar meaning for both log agents.

hi_water

Configure the pipe log agent in the same way that you configure file log agents. The **hi_water** parameter has similar meaning for both log agents.

path

Configure the **path** parameter to specify the location of the program to receive the log output as standard input. For example:

```
[aznapi-configuration]
logcfg = category:pipe path=/opt/risk_analyser/bin/my_log_watcher
```

There is no default value.

queue_size

Configure the pipe log agent in the same way that you configure file log agents. The **queue_size** parameter has similar meaning for both log agents.

Sending events to another program

You might configure Security Access Manager to send event records to another program.

Before you begin

Before you begin this task, review the information in "Configuring pipe log agents" on page 183.

Procedure

1. Edit the appropriate server configuration file. Each server provides its own stanza entry values in its configuration file.
2. Locate the stanza that contains the `logcfg` entries.
3. Specify the category using this format: *`category`*`:pipe`

   For example, a category might be to audit authorization events (`audit`):

   ```
   logcfg = audit:pipe
   ```
4. Specify that you want to pipe (`:pipe`) event records to another program: For example:

   ```
   logcfg = audit:pipe
   ```
5. Specify the path to the location of the program to receive the log output on standard input:

   ```
   path=fully_qualified_path
   ```

   There is no default value.
6. Specify the maximum number of events to queue in memory:

   ```
   queue_size={0|number_events}
   ```

   By default, the queue size is `0`. A zero queue size means that no limit is enforced on the growth of the unprocessed event queue. A requesting thread is blocked until space is available in the queue if:

   - The *number_events* value is greater than zero.
   - The queue size reaches the maximum *number_events* value.
   - A new event is ready to be placed on the queue.
7. Specify the event queue high water mark:

   ```
   hi_water={0|1|number}
   ```

   By default, the event queue high water mark value is two-thirds of the maximum configured queue size.

   If the maximum queue size is `0`, the high water mark is set to a default of `100`. The transaction rates and the values of these options determine the maximum amount of memory that is consumed by enabling event logging to file.

   If the event queue high water mark is set to `1`, every event queued is relayed to the log agent as soon as possible. This setting is not optimal.
8. Specify the frequency for flushing log file buffers:

   ```
   flush_interval={0|<0|number_seconds}
   ```

   A flush interval of `0` is not allowed. Specifying a value of zero results in the value `600` seconds being used.

   If you specify a negative value, the absolute value is used as the asynchronous flush frequency. However, a stream flush is also forced synchronously after every record is written.

   Ensure that stream buffers are flushed to disk regularly. Use the **flush_interval** parameter to control how often the server asynchronously flushes the file stream.
9. Save and exit the configuration file.

Results

This example pipes event records to the my_log_watcher file:

```
[aznapi-configuration]
logcfg = audit:pipe
path=/opt/risk_analyser/bin/my_log_watcher,queue_size=0,hi_water=100,
flush_interval=300
```

Configuring remote log agents

Configure the remote log agent to send events to a remote authorization server for recording. For example:

```
[aznapi-configuration]
logcfg = category:remote buffer_size=size,
     compress={yes|no},error_retry=timeout,path=name,
     flush_interval=number_seconds,rebind_retry=timeout,
     server=hostname,port=number,dn=identity,
     queue_size=number,hi_water=number
```

Parameter names can be shortened to any unambiguous name. For example, the **hi_water** parameter can be shortened to hi.

Requests to log an event remotely are accepted on a best effort basis only. If the remote authorization server is not available, captured events are cached locally and relayed at a later date, if and when the server becomes available.

Only one remote logging connection is established to a remote authorization server. Consider the case where multiple configuration entries are made to:

- Selectively capture events,
- Capture events at different points of the event pool hierarchy, and
- To the same remote server.

Then, the remote connection is established according to the options of the first remote configuration entry processed. Multiple remote connections can be configured to log to different remote authorization servers.

Events received at the remote authorization server are placed in the event pool of that server. The events are placed in a different location from where they were originally captured on the client system. All events entering a host through the remote logging service are placed in a category constructed in the following manner:

```
remote.client-category-domain.hostname.program
```

Note: The short name version of the host name is shown in some of the examples, however, the fully qualified host name is often required. To obtain system configuration information, you can use the **gethostbyname** command. For events that are filtered by program name, that is, using **pdmgrd**, specify the fully qualified host name.

In the following example, all audit events logged remotely from **pdmgrd** program on host amazon appear on the remote log server under pool remote.audit.amazon.mydomain.com.pdmgrd. Appearing under one pool allows for the remote server to selectively record events in various destinations using standard configurations. All audit events from host amazon can be recorded centrally on host timelord by configurations such as the following examples.

To relay events remotely on host amazon, you might use this example:

```
[aznapi-configuration]
logcfg = audit:remote buffer=2000,compress=y,error=2,
path=/opt/PolicyDirector/log/remote.cache,rebind=600,server=timelord,port=7136
```

On host `timelord` to record events to file, you might use:

```
[aznapi-configuration]
logcfg = remote.audit:file path=consolidated_audit.log
logcfg = remote.audit.amazon.mydomain.com.pdmgrd:file path=amazon_pdmgrd_audit.log
```

Parameters for remote log agents

You can define the following parameters for remote log agents:

buffer_size

To reduce network traffic, events are buffered into blocks of the nominated size before relaying to the remote server. The **buffer_size** parameter specifies the maximum size message that the local program attempts to construct by combining smaller events into a large buffer. Buffers consist only of an integral number of events; events are not split across buffers. If any individual event exceeds that maximum configured size, the large event is sent in a buffer of its own, exceeding the configured value.

The default value is `1024` bytes.

compress

Security Access Manager events are principally text messages. To reduce network traffic, use the **compress** parameter to compress buffers before transmission and expand on reception.

The default value is `no`.

dn

To establish mutual authentication of the remote server, a distinguished name (DN) must be configured. The DN can be checked against the name returned in the remote server's certificate.

The default value is a null string. Explicitly specifying an empty string or using the default value enables the logging client to request a remote server connection with any server that is listening.

Specifying a value for the **dn** parameter limits successful connection to a specific server, such as:

```
dn="cn=ivacld/timelord.testnet.tivoli.com,o=policy director,c=us"
```

A distinguished name must be specified as a string that is enclosed by double quotation marks.

error

If a send to a remote service fails, the system tries again. Before trying again, the system waits for the error retry timeout in seconds. If the attempt to try again fails:

- The link is recorded.
- The given event and future events are saved.

Events are saved in the local event cache file until the remote service is available again.

The default value is `2` seconds.

flush_interval

Events can sit in memory for a long time if:

- Events are being consolidated into large buffers.
- There is less logging activity.

Further, events can sit in memory before being:

- Forwarded to the remote server.
- Written to the cache file.

The **flush_interval** parameter limits the time a process waits to fill a consolidation buffer.

The default value is `20` seconds. A flush interval of `0` is not allowed. Specifying a value of `0` results in the buffer being flushed every 600 seconds.

hi_water

The **hi_water** parameter for a remote logging connection is like the one specified for logging to a file.

path

Configure the **path** parameter to specify the location of a cache file on the local host. The cache file name defaults to `./`*`server`*`.cache`, where *server* is the name of the remote server being logged to.

If the running process cannot establish communication with the remote server, or the link fails during operation, event recording switches to storing events in the specified file. The switch lasts until the server becomes available again. When the server is available, events are drained from the disk cache and relayed to the remote server.

For example, suppose that the path value for **pdmgrd** on AIX, Linux, and Solaris operating systems is as follows:

```
path=/var/PolicyDirector/log/pdmgrd_remote.cache
```

The directory portion of this path must exist. The log file is created if it does not exist. The size of this file is not bound, and it does not have any rollover capability. If a remote server is not accessible for sufficient time, you could run out of disk space.

port

Configure the **port** parameter to specify the port that the remote authorization server listens on for remote logging requests.

The default value is port `7136`.

queue_size

The **queue_size** parameter for a remote logging connection is like the one specified for logging to a file.

rebind_retry

If the remote authorization server is unavailable, the log agent attempts to rebind to this server at this frequency in number of seconds.

```
rebind_retry=number_seconds
```

The default rebind retry timeout value is `300` seconds.

server

The remote logging services are offered by the authorization service. The **server** parameter nominates the hosts to which the authorization server process is bound for event recording.

```
server=hostname
```

Sending events to a remote authorization server

You might configure Security Access Manager to send event records to a remote authorization server.

Before you begin

Before you begin this task, review the information in "Configuring remote log agents" on page 185.

Procedure

1. Edit the appropriate server configuration file. Each server provides its own stanza entry values in its configuration file.
2. Locate the stanza that contains the `logcfg` entries.
3. Specify that the category is to send event records to a remote server using the format *category*`:remote`.

   For example, a category might be to audit authorization events (`audit`) :

   ```
   logcfg=audit:remote
   ```
4. Specify the maximum buffer size. This buffer size is the maximum size message that the local program attempts to construct by combining smaller events into a large buffer:

   ```
   buffer_size={0|number_bytes}
   ```

   If a *number_bytes* value is specified, events are packed into buffers of that size before being relayed to the remote server. By default, the buffer size before relaying to the remote server is `1024` bytes.

   Buffers consist of only an integral number of events; events are not split across buffers. If any individual event exceeds that maximum configured size, the large event is recorded in a buffer of its own, exceeding the configured value.
5. Specify the frequency for flushing log file buffers:

   ```
   flush_interval={0|number_seconds}
   ```

   The **flush_interval** parameter limits the time a process waits to fill a consolidation buffer.

   By default, the flush interval value is `20` seconds. A flush interval of `0` is not allowed. Specifying a value of `0` results in the buffer being flushed every 600 seconds.
6. Specify the maximum number of events to queue:

   ```
   queue_size={0|number_events}
   ```

   By default, the queue size is `0`. A zero queue size means that no limit is enforced on the growth of the unprocessed event queue. The requesting thread is blocked until space is available in the queue if:

   - The maximum value for *number_events* is specified.
   - The maximum value for *number_events* is reached.
   - A new event is ready to be placed on the queue.
7. Specify the event queue high water mark:

   ```
   hi_water={0|1|number}
   ```

By default, the event queue high water mark value is a *number* that represents two-thirds of the maximum configured queue size.

If the maximum queue size is `0`, the high water mark is set to a default of `100`. The transaction rates and the values of these options determine the maximum amount of memory that is consumed by enabling event logging to file.

If the event queue high water mark is set to `1`, every event queued is relayed to the log agent as soon as possible. This setting is not optimal.

8. Specify whether you want to compress buffers before transmission and expand on reception:

   ```
   compress={yes|no}
   ```

   By default, the compress value is `no` to disable.
9. Specify the time to wait whenever a send to a remote service fails and an error occurs:

   ```
   error=seconds
   ```

   By default, the error retry timeout is `2` seconds.
10. Specify the cache file location:

    ```
    path=fully_qualified_path
    ```

    The file name is *server_name*`_remote.cache`. For example: `pdmgrd_remote.cache`

    The default directories are:

    AIX, Linux, and Solaris operating systems
    `/opt/PolicyDirector/log`

    Windows operating systems
    `C:\Program Files\Tivoli\Policy Director\log\`

    The default file name depends on the type of logging being performed, such as `audit.log`
11. Specify the time between attempts to rebind (sign on):

    ```
    rebind_retry=number_seconds
    ```

    By default, the rebind retry timeout value is `300` seconds.
12. Specify the host name of the remote authorization server:

    ```
    server=hostname
    ```
13. Specify the remote server port number:

    ```
    port=pdacld_port
    ```

    By default, the port number value is `7136`.
14. Specify the remote server distinguished name to establish mutual authentication of the remote server:

    ```
    dn="distinguished_name"
    ```

    The default value for the **dn** parameter is a null string. Explicitly specifying an empty string or using the default value enables the logging client to request a remote server connection with any server listening.

    The **dn** parameter value limits a successful connection to a specific server, for example:

    ```
    dn="cn=ivacld/timelord.tivoli.com,o=policy director,c=us"
    ```

    A distinguished name must be specified as a string enclosed by double quotation marks.
15. Save and exit the configuration file.

Example

This example sends event records to the remote `timelord` server:

```
[aznapi-configuration]
logcfg = audit:remote buffer=2000,compress=y,error=2
path=/opt/PolicyDirector/log/remote.cache,rebind=600,server=timelord,port=7136
dn="cn=ivacld/timelord.tivoli.com,o=policy director,c=us"
```

Configuring remote syslog agents

Use the `logcfg` entry to configure the remote syslog agent to send events to a remote syslog server for recording.

For example:

```
[aznapi-configuration]
logcfg = category:rsyslog,error_retry=timeout,log_id=id,
     path=name,flush_interval=number_seconds,max_event_len=length,
     rebind_retry=timeout,server=hostname,port=number,
     ssl_keyfile=key_file,ssl_label=label,ssl_stashfile=stash_file,
     queue_size=number,hi_water=number
```

The agent accepts requests to log an event remotely on a best effort basis only. If the remote syslog server is not available, the agent buffers events in a local cache file. When the server becomes available again, the agent sends the events to the server.

Caching does not occur if you configure the agent to use clear text communication with the syslog server. Clear text communication occurs over the User Datagram Protocol (UDP), which does not guarantee message delivery. In this configuration, the network layer does not notify the agent if the server does not receive the event. This means that events can be lost if the remote syslog server becomes unavailable.

Note: If you do not want to use clear text communication, you can configure SSL. For SSL communication, the agent uses the TLS Cipher Suite to encrypt the data.

Parameters for remote syslog agents

You can define the following parameters for remote syslog agents:

error_retry

If a message sent to a remote syslog service fails, the system tries again. Before trying again, the system waits for the **`error_retry`** timeout in seconds. If the next attempt fails, the agent saves the current event and future events in the local cache file until the remote service is available again.

The default value is `2` seconds.

flush_interval

Events can sit in memory for a long time if there is only a small amount of logging activity.

The **`flush_interval`** parameter limits the time a process waits to fill a consolidation buffer.

The default value is `20` seconds. You cannot use a flush interval of `0` seconds. If you specify a value of `0`, the agent flushes the buffer every 600 seconds.

hi_water

Processing of the event queue is scheduled regularly at the configured flush interval. It is also triggered asynchronously when the queue size reaches a high water mark on the event queue.

Use the **`hi_water`** parameter to define this high water mark. The default value is two-thirds of the maximum configured queue size. If the maximum queue size is zero, the high water mark is set to a default of `100`.

The transaction rates and the values of these options determine the maximum amount of memory that the agent uses for logging events to file.

If the event queue high water mark is set to `1`, WebSEAL relays every queued event to the log agent as soon as possible. This setting is not optimal. A setting of `1` ensures that events get to disk as fast as possible, but this configuration adversely impacts overall performance.

log_id

The **`log_id`** parameter defines the name of the application that the syslog agent includes in the messages sent to the remote syslog server. This field is mandatory.

max_event_len

The **`max_event_len`** parameter specifies the maximum length of an event that the syslog agent transmits to the remote syslog server.

If the event text is longer than the configured length, the agent truncates the message to the maximum event length. If the maximum event length is zero, the agent does not truncate the event text.

If you are using clear text communication to transmit the event, set the **`max_event_len`** parameter to a value less than the maximum transmission unit (MTU). That is, use a value less than the MTU for the network path to the server to avoid fragmentation of the event.

path

Configure the **`path`** parameter to specify the location of a cache file on the local host. The cache file name defaults to `./`*`log_id`*`.cache`, where *log_id* is the value of the **`log_id`** parameter.

Event recording switches to storing events in the specified file if any of the following scenarios occur:

- The running process cannot establish communication with the remote server.
- The link fails during operation.

The switch lasts until the server becomes available again. When the server is available, the agent removes the events from the disk cache and relays them to the remote server.

For example, suppose that the path value for **`pdmgrd`** on AIX, Linux, and Solaris operating systems is as follows:

```
path=/var/PolicyDirector/log/pdmgrd_rsyslog.cache
```

The directory portion of this path must exist. If the log file does not exist, the agent creates the file. The size of this file is not bound, and it does not have any rollover capability. If a syslog server is not accessible for a sufficient time, you might run out of disk space.

port

Configure the **port** parameter to specify the port that the remote syslog server listens on for remote logging requests.

The default port value is 514 for clear text communication and 6514 for SSL communication.

queue_size

There is a delay between placing events on the queue and their removal by the file log agent. The **queue_size** parameter specifies the maximum size of the queue. Consider that a new event is ready to be placed on the queue. If the queue reaches the maximum size, the requesting thread is blocked until space is available in the queue.

This process causes the performance of the event propagation thread to slow down to the speed of the file logging thread.

You must use the **queue_size** parameter to limit the central event propagation queue size. If not, memory usage by the log agent can grow without bounds.

```
[aznapi-configuration]
logcfg = audit.azn:rsyslog
...
queue_size=number_events,
...
```

The default value is 0. Specifying a value of 0 indicates that there is no limit to the growth of the unprocessed event queue. In this case, the speed of the logging thread does not constrain the event propagation thread. The unrecorded event queue can grow to an unmanageable size if:

- You are using the default value.
- Events are being generated faster than they can be recorded to file.

rebind_retry

If the remote syslog server is unavailable, the log agent attempts to rebind to this server at this frequency in number of seconds.

rebind_retry=*number_seconds*

The default **rebind_retry** timeout value is 300 seconds.

server

The remote logging services are offered by the remote syslog server. The **server** parameter nominates the host to which the agent is bound for event recording.

server=*hostname*

ssl_keyfile

The name of the GSKit key database file that contains the CA certificate. The logging agent uses the CA certificate to establish a secure connection with the remote syslog server over SSL.

If you do not configure this value, the logging agent uses clear text that is not encrypted to communicate with the remote syslog server.

ssl_label

The name of the certificate that the logging agent presents to the remote syslog server to establish a secure connection.

If you do not configure this field, the agent uses the default certificate from the key database.

ssl_stashfile

The name of the GSKit stash file that contains the password for the ssl-keyfile database. This field is mandatory if you specify a value for the **ssl-keyfile** field.

Sending events to a remote syslog server

You can configure Security Access Manager to send event records to a remote syslog server.

Before you begin

Before you begin this task, review the information in "Configuring remote syslog agents" on page 190.

Procedure

1. Edit the appropriate server configuration file. Each server provides its own stanza entry values in its configuration file.
2. Locate the stanza that contains the `logcfg` entries.
3. Specify that the category is to send event records to a remote server by using the format *`category`*`:rsyslog`.

   For example, a category that audits authorization events (`audit`):

   ```
   logcfg=audit:rsyslog
   ```
4. Specify the frequency for flushing log file buffers:

   ```
   flush_interval={0|number_seconds}
   ```

   The **flush_interval** parameter limits the time a process waits to fill a consolidation buffer.

   By default, the flush interval value is `20` seconds. You cannot use a flush interval of `0` seconds. If you specify a value of `0`, the agent flushes the buffer every 600 seconds.
5. Specify the maximum number of events to queue:

   ```
   queue_size={0|number_events}
   ```

   By default, the queue size is `0`. A zero queue size means that the agent does not limit the growth of the unprocessed event queue. The requesting thread is blocked until space is available in the queue if:

   - The maximum value for *number_events* is specified.
   - The maximum value for *number_events* is reached.
   - A new event is ready to be placed on the queue.
6. Specify the event queue high water mark:

   ```
   hi_water={0|1|number}
   ```

   By default, the event queue high water mark value is a *number* that represents two-thirds of the maximum configured queue size.

   If the maximum queue size is `0`, the high water mark is set to a default of `100`. The transaction rates and the values of these options determine the maximum amount of memory that the agent uses for logging events to file.

   If the event queue high water mark is set to `1`, WebSEAL relays every queued event to the log agent as soon as possible. This setting is not optimal.
7. Specify the time to wait whenever a send to a remote service fails and an error occurs:

   ```
   error_retry=seconds
   ```

   By default, the **`error_retry`** timeout is `2` seconds.

8. Specify the cache file location:

   path=*fully_qualified_path*

   The default cache file name is ./*log_id*.cache. For example: rsyslog.cache

   Note: The directory portion of this path must exist. If the log file does not exist, the agent creates the file.
9. Specify the time between attempts to rebind (sign on):

   rebind_retry=*number_seconds*

   By default, the **rebind_retry** timeout value is 300 seconds.
10. Specify the host name of the remote syslog server:

    server=*hostname*
11. Specify the remote server port number:

    port=*rsyslog_port*

    The default port number is 514 for clear text communication and 6514 for SSL communication.
12. Specify the application name that the syslog agent includes in the messages sent to the remote server:

    log_id=*name*
13. Specify the maximum length of an event that the agent transmits to the remote syslog server. If the event text is longer than this configured value, the agent truncates the message to the maximum event length. If the maximum event length is 0, the agent does not truncate the event text.

    max_event_len=*length*

    Note: If you are using clear text communication to transmit the event, set the **max_event_len** parameter to a value less than the maximum transmission unit (MTU). Use a value less than the MTU for the network path to the server to avoid fragmentation of the event.
14. Optional: If you require SSL communication with the remote server, you must specify the SSL keyfile:

    ssl_keyfile=*key_file*
15. Optional: If you are using SSL communication, you can use **ssl_label** to specify the certificate name:

    Note: If you do not configure a value for this field, the agent uses the default certificate from the key database.

    ssl_label=*my_label*
16. Optional: If you require SSL communication with the remote server, you must specify the SSL stash file:

    ssl_stashfile=*stash_file*

Example

This example sends event records to the remote timelord server:

```
[aznapi-configuration]
logcfg = audit:rsyslog error_retry=2,path=/opt/PolicyDirector/log/rsyslog.cache,
rebind_retry=600,server=timelord,port=514,log_id=webseal-instance
```

Disabling resource access events

You can use protected object policies (POPs) to selectively disable auditing of access to particular resources.

Procedure

- Disable generating audit records.

  If a POP with the audithttp extended attribute set to no is attached to a resource, access to that resource does not generate an HTTP access audit record. For example, if access to the /images subdirectory is not of sufficient interest to merit an audit record, you can disable audit records by using the following commands:

  ```
  pdadmin sec_master> pop create nohttpaudit
  pdadmin sec_master> pop modify nohttpaudit set attribute audithttp no
  pdadmin sec_master> pop attached /WebSEAL/server/images nohttpaudit
  ```

  After you attach the nohttpaudit POP to the /images subdirectory, access to files under this directory no longer generates an audit event.

- Enable generating audit records.

  If you have a specific resource that must be audited, you can enable auditing of that resource. To enable auditing, attach a second POP *without* the audithttp attribute. For example, the special.jpg file in the /images subdirectory must be audited. You can enable audit records for the file with the following commands:

  ```
  pdadmin sec_master> pop create restorehttpaudit
  pdadmin sec_master> pop attached /WebSEAL/server/images/special.jpg \
    restorehttpaudit
  ```

Process flow for logcfg logging

The following example process flow assumes the [aznapi-configuration] stanza of a WebSEAL configuration file.

Use the syntax of the logcfg entry to specify a log file. The log file is opened at WebSEAL initialization. If no log file is opened during initialization, regardless of other configuration settings, no events are logged. Unless a log file is specified, all event data is lost.

```
[aznapi-configuration]
logcfg = http.agent:file path=/var/pdweb/log/abc.log,log_id=agent
```

You can use the log_id identifier to facilitate the recording of events from different categories to the same file. You can construct more log agents. The log agents can gather different event data. These agents use log_id to direct the data to the log file that was opened by the initial log agent. The first logcfg entry must be used to define the log agent. If the log agent is defined after the first log_id, no events for that category are logged.

In the following example, events from the http.agent category are directed to the abc.log file. The log agent has the log_id=httplogs identifier. Events from http.ref and http.clf audit categories are also logged to this file because the logcfg entry uses the same identifier log_id=httplogs:

```
[aznapi-configuration]
logcfg = http.agent:file path=/var/pdweb/log/abc.log,log_id=httplogs
logcfg = http.ref:file log_id=httplogs
logcfg = http.clf:file log_id=httplogs
```

Auditing using logaudit

WebSEAL and Plug-in for Web Servers continue to support audit logging that uses the logaudit entries and its related entries in the [aznapi-configuration] stanza. This approach uses the following stanza entries:

```
[aznapi-configuration]
logaudit
auditlog
auditcfg
logsize
logflush
```

This approach is comparable to the logcfg entry with a file agent.

For example, to capture authentication events, you can set the configuration file entries as follows:

```
[aznapi-configuration]
logaudit = yes
auditcfg = authn
auditlog = /var/pdweb/log/audit.log
logsize = 2000000
logflush = 20
```

If you are still using the logaudit approach, consider by using either the logcfg approach or the Common Auditing Service. The logcfg approach provides more configuration options, such as buffer size and event queues, and the ability to use the console, pipe, and remote log agents.

Chapter 20. WebSEAL HTTP logging

This chapter describes WebSEAL HTTP logging.

HTTP log files

WebSEAL maintains the following HTTP log files that record HTTP activity:
- `request.log`
- `agent.log`
- `referer.log`

By default, these log files are in the following directory:

AIX, Linux, and Solaris operating systems

```
/var/pdweb/www-instance/log
```

Windows operating systems

```
C:\Program Files\Tivoli\PDWeb\www-instance_name\log
```

where *instance_name* is the instance to which WebSEAL is configured.

Stanza entries for configuring traditional HTTP logging are in the `[logging]` stanza of the WebSEAL configuration file.

Table 43 illustrates the relationship among the HTTP logs and the configuration file entries:

Table 43. Relationship between HTTP logs and the stanza entries

File name	Log file entry	Enablement entry
request.log	`requests-file`	`requests`
referer.log	`referers-file`	`referers`
agent.log	`agents-file`	`agents`

For example, the following entry shows the default location of the `request.log` file:

AIX, Linux, and Solaris operating systems

```
[logging]
requests-file = /var/pdweb/www-instance_name/log/request.log
```

Windows operating systems

```
[logging]
requests-file = /
 C:\Program Files\Tivoli\PDWeb\www-instance_name\log\request.log
```

where *instance_name* is the instance to which WebSEAL is configured.

Enabling HTTP logging

By default, HTTP logging is enabled in the WebSEAL configuration file. For example:

```
[logging]
requests = yes
referers = yes
agents = yes
```

You can enable or disable each log independently from the others. If any stanza entry is set to no, logging is disabled for that file.

Configuring HTTP logging in the [logging] stanza implements the standard event logging mechanism that is described in Chapter 19, "Audit event logging," on page 173.

The following configurations are created when the WebSEAL HTTP logging stanza entries are enabled. These configurations accept the values of the requests-file, referers-file, agents-file, flush-time, and max-size stanza entries from the WebSEAL configuration file [logging] stanza:

request.log

```
logcfg = http.clf:file path=requests-file,flush=flush-time,
rollover=max-size,log=clf,buffer_size=8192,queue_size=48
```

referer.log

```
logcfg = http.ref:file path=referers-file,flush=flush-time,
rollover=max-size,log=ref,buffer_size=8192,queue_size=48
```

agent.log (common log format)

```
logcfg = http.agent:file path=agents-file,flush=flush-time,
rollover=max-size,log=agent,buffer_size=8192,queue_size=48
```

See "Process flow for logcfg logging" on page 195 for special considerations and conditions when you use both traditional HTTP logging ([logging] stanza) and the event logging mechanism ([aznapi-configuration] stanza).

Specifying the timestamp

You can choose to have timestamps in each HTTP log file that is recorded in Greenwich Mean Time (GMT). This GMT choice overrides the local time zone. By default, the local time zone is used.

To use GMT timestamps, set the value of the gmt-time entry to yes as shown in the following entry:

```
gmt-time = yes
```

Specifying rollover thresholds

The max-size stanza entry specifies the maximum size to which each of the HTTP log files can grow and has the following default value in bytes:

```
[logging]
max-size = 2000000
```

When a log file reaches its rollover threshold:

- The existing file is backed up to a file of the same name. The file name is appended with the current date and timestamp.
- A new log file is started.

The various possible max-size values are interpreted as follows:

- If the max-size value is less than zero (< 0), a new log file is created:
  - With each invocation of the logging process.

  - Every 24 hours from that instance.
- If the max-size value is equal to zero (= 0), no rollover is completed and the log file grows indefinitely. If a log file exists, new data is appended to it.
- If the max-size value is greater than zero (> 0), a rollover is completed when a log file reaches the configured threshold value. If a log file exists at startup, new data is appended to it.

Specifying the frequency for flushing buffers

Log files are written to buffered data streams. If you are monitoring the log files in real time, alter the frequency with which the server flushes the log file buffers.

By default, log files are flushed every 20 seconds as shown in the following example:

```
[logging]
flush-time = 20
```

If you specify a negative value, a flush is forced after each record is written.

Distinguishing virtual hosts

When you use virtual hosts, you can use the request-log-format entry in the [logging] stanza to distinguish between requests to different virtual hosts.

Use the **%v** directive at the start of the request-log-format configuration item to include the header at the front of each line in the request log.

When you use the **%R** directive entry in the request-log-format configuration item, the log contains the absolute URI.

Customizing the HTTP request log

You can customize the content of the request.log file by adding a configuration entry in the [logging] stanza. The syntax is as follows:

```
request-log-format=directives
```

The following directives can be used to customize the log format:

Table 44. Directives for customizing the format of the request.log file

Directive	Description
%a	Remote IP address
%A	Local IP address
%b	Bytes in the response excluding HTTP headers in CLF format: '-' instead of 0 when no bytes are returned
%B	Bytes in the response excluding HTTP headers
%{Attribute}C	Attribute from the Security Access Manager credential named 'Attribute'
%d	Transaction identifier, or session sequence number
%F	Time that it takes to serve the request in microseconds
%h	Remote host
%H	Request protocol
%{header-name}i	Contents of the Header 'header-name' in the request

Table 44. Directives for customizing the format of the request.log file (continued)

Directive	Description
%j	The name of the junction that services the request
%l	Remote logname
%m	Request method (that is, GET, POST, HEAD)
%{header-name}o	Contents of the Header 'header-name' in the response
%p	Port over which the request was received
%q	The query string (prefixed with '?' or empty)
%r	First line of the request
%R	First line of the request including HTTP://HOSTNAME
%s	Response status
%t	Time and date in CLF format
%{format}t	The time and date in the specified format
%T	Time that it takes to serve the request in seconds
%u	Remote user
%U	The URL requested
%v	Canonical ServerName of the server that serves the request
%{cookie-name}e	Contents of the cookie 'cookie-name' in the request
%{cookie-name}E	Contents of the cookie 'cookie-name' in the response

The following configuration entry shows an example of customizing the `request.log` file:

```
request-log-format = %h %l %u %t "%r" %s %b
```

Customized HTTP logs also support the new line (\n), carriage return (\r), and tab (\t) special characters. Any character that is either not part of a directive or not a special character is written out in the log entry. You can direct the system to ignore the % and \ characters by prefixing them with the backslash (\) character. For example:

```
log-request-format = \%{header}i\t->\t%{header}i
```

renders the following output:

```
%{header}i -> header
```

Process flow for [logging] and logcfg logging

You can configure WebSEAL auditing you use both the `[logging]` stanza and the `[aznapi-configuration]` stanza.

When you use both configuration settings, WebSEAL processes the `[aznapi-configuration]` stanza before the `[logging]` stanza.

For example, assuming the following entries in the WebSEAL configuration file:

```
[logging]
requests = yes
requests-file = /var/pdweb/www-instance/log/request.log
```

```
[aznapi-configuration]
logcfg = stats.pdweb.authn:file path=/var/pdweb/log/stats.log,log_id=stats
logcfg = http.agent:file path=/var/pdweb/log/abc.log,log_id=httplogs
logcfg = http.ref:file log_id=httplogs
```

WebSEAL processes these entries in the following manner:

1. The [aznapi-configuration] stanza is read.
2. The stats.log file with log_id=stats is opened. All stats.pdweb.authn events are logged to this file.
3. The abc.log file with log_id=httplogs is opened. All http.agent events are logged to this file.
4. Because the next log agent uses log_id=httplogs, all http.ref events are logged to the previously opened abc.log file.
5. The [logging] stanza is read.
6. HTTP request logging is enabled. All http.clf events are logged to the request.log file that uses the default log_id=clf. See the following example for an explanation of this default identifier.

HTTP logging using the [logging] stanza operates by generating its own default log agent entries. Each HTTP log file has a default value for the log_id parameter.

Log file	log_id
request.log	log_id=clf
referer.log	log_id=ref
agent.log	log_id=agent

If a logcfg entry in the [aznapi-configuration] stanza contains the same log_id as one used in the [logging] stanza, the HTTP log file is not created. Audit events with the same log_id are directed to 1 log file only. That 1 log file is always the first one opened.

In the following example, the abc.log file with log_id=clf is opened first. Because the HTTP requests logging defined in the [logging] stanza uses a default log_id=clf, the requests.log file is never created and all http.clf (requests) events are directed to abc.log file.

```
[logging]
requests = yes
requests-file = /var/pdweb/www-instance/log/request.log

[aznapi-configuration]
logcfg = http.agent:file path=/var/pdweb/log/abc.log,log_id=clf
logcfg = http.ref:file log_id=clf
```

HTTP logging can be configured in the [logging] and [aznapi-configuration] stanzas. Therefore, it is possible to have duplicate entries for HTTP events in a log file when both mechanisms are enabled.

In the following example, http.clf audit events are recorded twice in the abc.log file:

- From the event logging configuration.
- From the enabled request logging, which uses log_id=clf by default. The requests.log is not created because the abc.log file with log_id=clf is opened first.

```
[logging]
requests = yes
requests-file = /var/pdweb/www-instance/log/request.log

[aznapi-configuration]
logcfg = http.agent:file path=/var/pdweb/log/abc.log,log_id=clf
logcfg = http.ref:file log_id=clf
logcfg = http.clf:file log_id=clf
```

Sample request.log file

The content of the request.log file is set by the request-log-format configuration item. The following table shows all the possible initial request-log-format combinations that are based on the existing absolute-uri-in-request-log and host-header-in-request-log configuration items:

Table 45. Example output of the request.log file

absolute-uri-in-request-log	host-header-in-request-log	request-log-format	Example output
No	No	%h %l %u %t "%r" %s %b	10.251.173.1 - sec_master [04/Jan/2009:11:13:07 +1000] "GET /pics/iv30.gif HTTP/1.1" 200 46498
No	Yes	%v %h %l %u %t "%r" %s %b	tamtestbed 10.251.173.1 - sec_master [04/Jan/2009:11:10:04 +1000] "GET /pics/iv30.gif HTTP/1.1" 200 46498
Yes	No	%h %l %u %t "%R" %s %b	10.251.173.1 - sec_master [04/Jan/2009:11:14:51 +1000] "GET HTTP://tamtestbed/pics/iv30.gif HTTP/1.1" 200 46498
Yes	Yes	%v %h %l %u %t "%R" %s %b	tamtestbed 10.251.173.1 - sec_master [04/Jan/2009:11:16:40 +1000] "GET HTTP://tamtestbed/pics/iv30.gif HTTP/1.1" 200 46498

Sample agent.log file

The agent.log file records the contents of the User_Agent: header in the HTTP request.

This log reveals information about the client browser, such as architecture or version number, for each request. The following example shows a sample version of the agent.log file:

```
Mozilla/4.01 [en] (WinNT; U)
Mozilla/4.01 [en] (WinNT; U)
Mozilla/4.01 [en] (WinNT; U)
Mozilla/4.01 [en] (WinNT; U)
```

Sample referer.log

The `referer.log` records the `Referer:` header of the HTTP request. For each request, the log records the document that contained the link to the requested document.

The log uses the following format:

```
referer -> object
```

This information is useful for tracking external links to documents in your web space. The log reveals that the source indicated by *referer* contains a link to a page (*object*). With this log, you can track stale links and to find out who is creating links to your documents.

The following example shows a sample version of a `referer` log file:

```
http://manuel/maybam/index.html -> /pics/tivoli_logo.gif
http://manuel/maybam/pddl/index.html -> /pics/tivoli_logo.gif
http://manuel/maybam/ -> /pddl/index.html
http://manuel/maybam/ -> /pddl/index.html
http://manuel/maybam/pddl/index.html -> /pics/tivoli_logo.gif
http://manuel/maybam/ -> /pddl/index.html
```

Chapter 21. Working with statistics

This chapter provides information about working with the Security Access Manager modules that can monitor and collect statistical information.

Using stats commands for statistics

Use the **server tasks stats** command that is provided as by the **pdadmin** utility to manage statistics components. You can use the **stats** command to complete the following operations:

stats on
: Enable statistics for a specific component.

stats off
: Disable statistics for a specific component or for all components.

stats show
: List enabled components.

stats get
: Display current statistics values for a specific component or for all components.

stats reset
: Reset statistics values for a specific component or for all components.

stats list
: List all statistics components.

For more information about the **server task stats** command, see "Using stats commands for statistics."

Enabling statistics

You can enable statistics reporting with the **stats on** command or with stanza entries in the configuration file for the specific server.

For details about using stanza entries to enable statistics, see "Using stanza entries for statistics" on page 209.

To enable the gathering of statistics with the **stats on** command, set the statistics report frequency, event count, and destination for the component. For more information about the **stats on** command, see "server task stats" on page 412.

Note:

- By default, the WebSEAL pdweb.threads, pdweb.doccache, and pdweb.jmt components are always enabled and cannot be disabled.
- Using **stats on** and changing the runtime Policy server trace settings affects only the current run of the Policy server. If the Policy server is stopped and then started later, the default trace settings take effect. To persist trace settings across multiple runs of the Policy server, modify the `/etc/pdmgrd_routing` file.

When you enable statistics, you can specify one log file for the statistics report. If you specify two equivalent commands that differ only on the destination, the second invocation deactivates the first log file and activates the second log file. The following example illustrates this limitation:

```
#pdadmin> server task default-webseald-abc.ibm.com stats on pdweb.http 20 \
  file path=/tmp/A.log
```

```
#pdadmin> server task default-webseald-abc.ibm.com stats on pdweb.http 20 \
  file path=/tmp/B.log
```

The first command enables the `pdweb.http` component and sends statistics reports to the `A.log` file. The second command attempts to activate a second log file, `B.log`. However, this action actually deactivates the `A.log` file while it also activates the `B.log` file.

Enabling basic statistics

To enable basic statistics gathering, use the **`stats on`** command and specify only the *component* option. Because the *interval* option is not specified, you can obtain statistics information only for this component with the **`stats get`** command. Because the *destination* option is not specified, the information is sent to the standard log file for that component.

The following example enables the gathering of statistics for the pdweb.http component of a WebSEAL instance:

```
#pdadmin> server task default-webseald-abc.ibm.com stats on pdweb.http
```

Enabling statistics with frequency and count

To enable the gathering of statistics at a designated frequency and event count, use the **`stats on`** command and specify the following options:

- *component*
- *interval*
- *count*

The *interval* and *count* options:

- Cause the buffer to accumulate a specific number of entries that represent a statistics report.
- Flush the buffer after a specific number of seconds elapse.

Because the *destination* option is not specified, the information is sent to the standard log file for that component.

The following example enables the gathering of statistics for the pdweb.http component of a WebSEAL instance. In this example, the buffer accumulates 100 entries and sends statistics reports every 20 seconds:

```
#pdadmin> server task default-webseald-abc.ibm.com stats on pdweb.http 20 100
```

Enabling statistics with frequency and destination

To enable gathering of statistics at a designated frequency and write the statistics to a specific file, use the **`stats on`** command and specify the following options:

- *component*
- *interval*
- *destination*

The *interval* option, without a *count* option, indefinitely sends statistics reports after a specific number of seconds elapses. The *destination* option specifies the exact file where the statistics are written. When you specify a file that is different for the file log agent for the component, you can specify more configuration options.

The following example enables the gathering of statistics for the pdweb.http component of a WebSEAL instance where:

- A statistics report is sent to the `/tmp/jmt-stats.log` file every 20 seconds.
- A new file is created each time that the buffer is flushed.

```
#pdadmin> server task default-webseald-abc.ibm.com stats on pdweb.http 20 \
  file path=/tmp/jmt-stats.log,rollover_size=-1
```

The growth of the log file is controlled by the `rollover_size` configuration option. For complete details about configuring event logging, see the *IBM Security Access Manager for Web Troubleshooting Guide*.

Disabling statistics

You can disable statistics reporting with the **`stats off`** command for a specific component or for all components. By default, the pdweb.threads, pdweb.doccache, and pdweb.jmt components are always enabled and cannot be disabled.

Disabling statistics for all components

To disable the gathering of statistics for all components, use the **`stats off`** command without options.

The following example disables statistics for all components of a WebSEAL instance:

```
#pdadmin> server task default-webseald-abc.ibm.com stats off
```

Disabling statistics for a single component

To disable the gathering of statistics for a single component, use the **`stats off`** command with the *component* option.

The following example disables statistics for the pdweb.sescache component of a WebSEAL instance:

```
#pdadmin> server task default-webseald-abc.ibm.com stats off pdweb.sescache
```

Listing enabled components

You can use the **`stats show`** command to:

- List all enabled statistics components.
- Determine whether a specific component is enabled.

Listing all enabled components

To display a list of all components, use the **`stats show`** command without options.

The following example displays a list of the enabled component of a WebSEAL instance:

```
#pdadmin> server task default-webseald-abc.ibm.com stats show

pdweb.authn
pdweb.doccache
pdweb.jmt
pdweb.sescache
pdweb.threads
```

Because the pdweb.threads, pdweb.doccache, and pdweb.jmt components are always enabled, the output for a WebSEAL instance always contains these entries.

Determining whether a component is enabled

To determine whether a component is enabled, use the **stats show** command with the *component* option.

If the component is enabled, the output lists that component. If the component is not enabled, no output is displayed.

Displaying statistics

You can display the current statistics for all enabled components or for a single component with the **stats get** command.

Displaying statistics for all components

To display statistics for all components, use the **stats get** command without options. For each enabled component, the name of the component is displayed followed by its statistics. For details about the specifics of the statistics for each component, see the information for that specific component in one of the following sections:

- "Security Access Manager components and activity types" on page 211
- "WebSEAL components and activity types" on page 212
- "Plug-in for Web Servers components and activity types" on page 219

The following example displays the current statistics for all enabled components of a WebSEAL instance:

```
#pdadmin> server task default-webseald-abc.ibm.com stats get

pd.ras.stats.monitor
  ...
pd.log.EventPool.queue
  ...
pd.log.file.clf
  ...
pd.log.file.ref
  ...
pd.log.file.agent
  ...
pdweb.authn
  ...
pdweb.authz
  ...
pdweb.http
  ...
pdweb.https
  ...
pdweb.threads
  ...
pdweb.sescache
  ...
pdweb.doccache
  ...
pdweb.jct.1
  ...
pdweb.jct.2
...
```

Displaying statistics for a single component

To display statistics for a single component, use the **stats get** command with the *component* option.

The following example displays the current statistics for the pdweb.threads component of a WebSEAL instance:

```
#pdadmin> server task default-webseald-abc.ibm.com stats get pdweb.threads

active : 4
total : 50
'default' active : 4
'default' total : 50
```

Resetting statistics

You can reset the current statistics for all enabled components or for a single component with the **stats reset** command.

To reset statistics for all components, use the **stats reset** command without options.

To reset statistics for a single component, use the **stats reset** command with the *component* option.

Listing components

You can list all components that are available to gather and report statistics with the **stats list** command.

To determine which queues are implemented on a server, use the **stats list** command. The following example lists all available components of a WebSEAL instance:

```
#pdadmin> server task default-webseald-abc.ibm.com stats list

pd.ras.stats.monitor
pd.log.EventPool.queue
pd.log.file.clf
pd.log.file.ref
pd.log.file.agent
pdweb.authn
pdweb.authz
pdweb.http
pdweb.https
pdweb.threads
pdweb.jmt
pdweb.sescache
pdweb.doccache
pdweb.jct.1
```

Using stanza entries for statistics

The configuration file for each server contains the following stanza entries that can be set to:

- Enable the statistics interface.
- Specify the destination for statistics reports.
- `stats`
- `logcfg`

The following segment of a configuration file shows the structure of the `stats` and `logcfg` stanza entries:

```
[aznapi-configuration]
stats = component [interval [count]]
logcfg = stats.component:destination
```

For information about the *interval* and *count* options, see "server task stats" on page 412. For complete details about configuring event logging, see the *IBM Security Access Manager for Web Troubleshooting Guide*.

Enabling statistics for a single component

In a server configuration file, you can enable gathering of statistics by using the stats and logcfg entries. These entries are in the [aznapi-configuration] stanza.

In the following example:

- The stats stanza entry enables gathering of statistics for the pdweb.jmt component. The frequency is 20 seconds.
- The logcfg stanza entry specifies the destination for the statistics report as the /tmp/jmt.log file. The entry contains more configuration information for the rollover_size and flush configuration settings:

```
[aznapi-configuration]
stats = pdweb.jmt 20
logcfg = stats.pdweb.jmt:file path=/tmp/jmt.log,rollover_size=-1,flush=20
```

For detailed information about configuration files, see the *IBM Security Access Manager for Web Administration Guide*.

Enabling statistics for multiple components

Unlike the **stats on** command, you enable gathering of statistics for multiple components by using multiple stats and logcfg entries in the [aznapi-configuration] stanza. The stanza is in the server configuration file.

In the following example, statistics gathering is enabled for the following WebSEAL components:

pdweb.authn

For the pdweb.authn component:

- The frequency is set to 40 seconds.
- The destination for the statistics report is the /tmp/an.log file.

The component has more configuration information for the rollover_size and flush configuration settings.

pdweb.jct.1

For the pdweb.jct.1 component:

- The frequency is set to 50 seconds,
- The destination for the statistics report is the /tmp/jct.log file.

The component has more configuration information for the rollover_size and flush configuration settings.

pdweb.jmt

For the pdweb.jmt component:

- The frequency is set to 20 seconds.
- The destination for the statistics report is the /tmp/jmtA.log and the /tmp/jmtB.log files.

The component has more configuration information for the rollover_size and flush configuration settings.

```
[aznapi-configuration]
stats = pdweb.jmt 20
stats = pdweb.authn 40
stats = pdweb.jct.1 50
logcfg = stats.pdweb.jmt:file path=/tmp/jmtA.log,rollover_size=-1,flush=20
logcfg = stats.pdweb.jmt:file path=/tmp/jmtB.log,rollover_size=-1,flush=20
logcfg = stats.pdweb.authn:file path=/tmp/an.log,rollover_size=-1,flush=20
logcfg = stats.pdweb.jct.1:file path=/tmp/jct.log,rollover_size=-1,flush=20
```

For detailed information about configuration files, see the *IBM Security Access Manager for Web Administration Guide*.

Security Access Manager components and activity types

The following statistics components are available to Security Access Manager servers:
- pd.log.EventPool.queue
- pd.log.file.agent
- pd.log.file.audit
- pd.log.file.clf
- pd.log.file.ref
- pd.ras.stats.monitor

pd.log.EventPool.queue

The pd.log.EventPool.queue component is the main event propagation queue. Use the statistics interface to monitor:
- The queuing profiles that are configured for the main propagation queue.
- Each file agent.
- Remote agent.
- Pipe log agent.

Each queue that is created as an instance of the EventQueue object registers itself with the statistics subsystem with its category name. The category name is constructed from the logging agent type and the pd.log string.

The following example shows the output from a **stats get** command for the pd.log.EventPool.queue component:

```
#pdadmin> server task ivacld-instance stats get \
  pd.log.EventPool.queue

dispatcher wakes on timeout (20) : 3617
dispatcher wakes by notify : 0
   notifies above highwater (100) : 0
   notifies below highwater : 0
   spurious notifies : 0
total events processed : 24
average number of events handled per activation : 1
greatest number of events handled per activation : 7
blocks in queue requests : 0
```

In the previous output:
- The flush frequency for the queue is 20, the value that is denoted in the parentheses after timeout.
- The high water setting for the queue is 100, the value that is denoted in the parentheses after highwater.

The settings that are defined for the various queue configuration options must attempt to balance:

- The maximum amount of memory that is consumed between queue activations, and
- The rate at which a particular log agent can consume events.

Set the queue high water mark such that the number of events that are processed during a queue activation fills a processing time slice. This setting avoids unnecessary thread context-switching. However, setting these options to large values is not productive. The reason is that event log processing must be done at some point and cannot be deferred indefinitely. Consuming large amounts of memory has its own drawbacks.

pd.log.file.agent

```
dispatcher wakes on timeout (20) : 299
dispatcher wakes by notify : 0
    notifies above highwater (33) : 0
    notifies below highwater : 0
    spurious notifies : 0
total events processed : 146
average number of events handled per activation : 0
greatest number of events handled per activation : 1
blocks in queue requests : 0
```

pd.log.file.clf

```
dispatcher wakes on timeout (20) : 299
dispatcher wakes by notify : 0
    notifies above highwater (33) : 0
    notifies below highwater : 0
    spurious notifies : 0
total events processed : 147
average number of events handled per activation : 0
greatest number of events handled per activation : 1
blocks in queue requests : 0
```

pd.log.file.ref

```
dispatcher wakes on timeout (20) : 300
dispatcher wakes by notify : 0
    notifies above highwater (33) : 0
    notifies below highwater : 0
    spurious notifies : 0
total events processed : 148
average number of events handled per activation : 0
greatest number of events handled per activation : 1
blocks in queue requests : 0
```

pd.ras.stats.monitor

```
5 components reporting statistics
5 reports generated
```

WebSEAL components and activity types

The following statistics components are available to WebSEAL instances:

- pdweb.authn
- pdweb.authz
- pdweb.doccache
- pdweb.http
- pdweb.https
- pdweb.jct.#
- pdweb.jmt

- pdweb.sescache
- pdweb.threads

pdweb.authn component

The pdweb.authn statistics component gathers information about WebSEAL authentication. The following list describes the types of available information:

pass The total number of successful authentications.

fail The total number of failed authentications.

pwd exp
The total number of authentication attempts that were made with an expired password.

max The maximum time for a single authentication process.

avg The average time for a single authentication process.

total The total time for all authentication processing.

The following example shows the output from a **stats get** command for the pdweb.authn component:

```
#pdadmin> server task default-webseald-instance stats get pdweb.authn

pass     : 2
fail     : 1
pwd exp  : 0
max      : 0.178
avg      : 0.029
total    : 0.382
```

pdweb.authz component

The pdweb.authz statistics component gathers information about WebSEAL authorization. The following list describes the types of available information:

pass The total number of successful authorization requests. That is, the total number of resources that were successfully accessed.

fail The total number of failed authorization requests.

The following example shows the output from a **stats get** command for the pdweb.authz component:

```
#pdadmin> server task default-webseald-instance stats get pdweb.authz

pass     : 2
fail     : 1
```

pdweb.doccache component

The pdweb.doccache statistics component gathers information about WebSEAL document-caching activity. This component reports statistics for all MIME types that aer enabled in the [content-cache] stanza of the WebSEAL configuration file. This component is always enabled by default and cannot be disabled.

The following list describes the types of global information available for all MIME types:

General Errors
The number of errors reported by the pdweb.doccache component when there are memory allocation failures, initialization failures, or invalid MIME type header values.

Uncachable
The number of instances when there is no cache that is defined for the MIME type of the document to be cached.

Pending Deletes
The number of entries that are marked for deletion, but these entries are still in use.

Pending Size
The number of bytes that are used by entries that are marked for deletion, but these entries are still in use.

Misses
The number of times a URL is looked up in the document cache and is not found. A found cached document eliminates the need to access the real document again.

Cache MIME type
The MIME type of documents that is stored in this cache. The following list describes the cache MIME types:

Max size
The maximum combined byte size of all documents in the cache.

Max entry size
The maximum byte size for any single cached document. If the document size exceeds this internally calculated value, it is not cached.

Size The total byte count for all documents currently located in the cache.

Count The current number of entries in the cache.

Hits The number of successful lookups. (Documents that are successfully found in the cache.)

Stale hits
The number of successful lookups that found an entry that was too old and was purged instead.

Create waits
The number of times subsequent requests for a document are blocked (made to wait) while the document content is initially being cached.

Cache no room
The number of times a document that is valid for caching cannot fit into the cache. The reason is that there are too many entries that are being created at the same time.

Additions
The number of successful new entries in the cache.

Aborts
The number of times the creation of a new cache entry is canceled. The reason might be a header that indicates the entry must not be cached.

Deletes
The number of cache entries that were deleted because the entry is stale (expired) or because the creation was canceled.

Updates
The number of entries that had expiry times updated.

Too big error
The number of attempts to cache documents that exceed the maximum entry size (and therefore are not cached).

MT errors
The number of times more than one thread tries to create the same entry in the cache. (MT=Multi-Threading)

The following example shows the output from a **stats get** command for the pdweb.doccache component:

```
#pdadmin> server task default-webseald-instance stats get pdweb.doccache

General Errors : 0
Uncachable     : 0
Pending Deletes: 0
Pending Size   : 0
Misses         : 0
Cache MIME type  : text/html
  Max size       : 2048000
  Max entry size : 128000
  Size           : 0
  Count          : 0
  Hits           : 0
  Stale hits     : 0
  Create waits   : 0
  Cache no room  : 0
  Additions      : 0
  Aborts         : 0
  Deletes        : 0
  Updates        : 0
  Too big errors : 0
  MT errors      : 0
```

pdweb.http component

The pdweb.http statistics component gathers information about WebSEAL HTTP communication. The following list describes the types of available information:

reqs The total number of HTTP requests received.

max-worker
The maximum time that is used by a single worker thread to process an HTTP request.

total-worker
The total time that is used by all worker threads that process HTTP requests.

max-webseal
The maximum time that is used to process a single HTTP request - measured inside the worker thread, after the request headers are read, and eliminating connection setup overhead.

total-webseal
The total time that is used to process all HTTP requests - measured inside the worker threads, after the request headers are read, and eliminating connection setup overhead.

The following example shows the output from a **stats get** command for the pdweb.http component:

```
#pdadmin> server task default-webseald-instance stats get pdweb.http
reqs            : 0
max-worker      : 0.000
total-worker    : 0.000
max-webseal     : 0.000
total-webseal   : 0.000
```

pdweb.https component

The pdweb.https statistics component gathers information about WebSEAL HTTPS communication. The following list describes the types of available information:

reqs
The total number of HTTPS requests received.

max-worker
The maximum time that is used by a single worker thread to process an HTTPS request.

total-worker
The total time that is used by all worker threads that process HTTPS requests.

max-webseal
The maximum time that is used to process a single HTTPS request - measured inside the worker thread, after the request headers are read, and eliminating connection setup overhead.

total-webseal
The total time that is used to process all HTTPS requests - measured inside the worker threads, after the request headers are read, and eliminating connection setup overhead.

The following example shows the output from a **stats get** command for the pdweb.https component:

```
#pdadmin> server task default-webseald-instance stats get pdweb.https

reqs            : 0
max-worker      : 0.000
total-worker    : 0.000
max-webseal     : 0.000
total-webseal   : 0.000
```

pdweb.jct.# component

The pdweb.jct.# statistics component gathers information about configured junctions. The following list describes the types of available information:

[/]
The actual junction name (listed as the number in the command)

reqs
The total number of requests that are routed across this junction

max
The maximum time that is consumed by a single request across this junction

total
The total time that is consumed by requests across this junction

The following example shows the output from a **stats get** command for the pdweb.jct.1 component:

```
#pdadmin> server task default-webseald-instance stats get pdweb.jct.1

[/]
reqs    : 0
max     : 0.000
total   : 0.000
```

pdweb.jmt component

The pdweb.jmt statistics component gathers information about the WebSEAL junction mapping table. This component is always enabled by default and cannot be disabled. The following list describes the types of available information:

hits The total number of requests that required URL mapping with the junction mapping table.

The following example shows the output from a **stats get** command for the pdweb.jmt component:

```
#pdadmin> server task default-webseald-instance stats get pdweb.jmt

hits  : 5
```

pdweb.sescache component

The pdweb.sescache component gathers statistics about the WebSEAL session cache. This component gathers the following activity information:

hit The number of requests where a cache entry for a user was referenced successfully. That is, the number of requests that resulted in a session cache hit.

miss The number of requests that missed a session cache hit.

add The number of cache entries that was added to the session cache.

del The number of cache entries that was deleted from the session cache.

inactive
The number of times where a cache entry hit the inactivity timeout.

lifetime
The number of times where a cache entry hit the lifetime timeout.

LRU expired
The number of times that a “least recently used” cache entry was deleted from the session cache to make room for a new cache entry.

The following example shows the output from a **stats get** command for the pdweb.sescache component:

```
pdadmin sec_master>  server task default-webseald-instance stats get pdweb.sescache
hit      : 225
miss     : 75
add      : 375
del      : 150
inactive : 60
lifetime : 15
LRU expired : 75
```

In the previous release, the pdweb.sescache component contained activity that was associated with callback certificates and user session mappings. These statistics are now managed by the following components:

pdweb.certcallbackcache
This cache stores the SSL IDs of sessions that require certificate validation

when a user is stepping up. The reported information has the same categories as pdweb.sescache. These activities are internal.

pdweb.usersessidcache
This cache stores a mapping of users to their sessions. The reported information has the same categories as pdweb.sescache. These activities are internal.

Therefore, the first time that you gather statistics for the pdweb.sescache component and compare it to your last report, the figures might appear to be wrong. To set a new baseline, add the statistics from the following components and then compare them to your previous baseline (last pdweb.sescache report):

- pdweb.sescache
- pdweb.certcallbackcache
- pdweb.usersessidcache

The output against the `pdweb.sescache` component must be your new baseline.

pdweb.threads component

The pdweb.threads statistics component gathers information about WebSEAL worker thread activity. Its report is the overall thread usage statistics that include not just request traffic, but all the worker threads for the WebSEAL process.

WebSEAL, version 6.0, and later can be configured to use multiple interfaces. Each separately configured interface can use a separate worker thread pool. The thread pool has the same name as the specified interface.

Alternatively, all configured interfaces can share worker thread pool. The default WebSEAL interface configuration uses the **default** name to differentiate between that interface and the corresponding thread pool, from other separately configured interfaces. The default WebSEAL interface configuration is defined under the [server] stanza. A separately configured WebSEAL interface (defined under the [interfaces] stanza) uses the specified name.

The pdweb.threads component is always enabled by default and cannot be disabled. The following list describes the types of available information:

active The total number of active worker threads of all WebSEAL interfaces that are handling requests.

total The total number of worker threads that are configured for all WebSEAL interfaces.

'default' active
The total number of active worker threads in the default interface thread pool that are handling requests. If you do not configure one or more more WebSEAL interfaces, the value of **default active** matches the value of **active**.

'default' total
The total number of configured worker threads for the default interface thread pool. If you do not configure one or more more WebSEAL interfaces, the value of **default total** matches the value of **total**.

'*other_interface*' active
The total number of active worker threads in the thread pool that is handling requests for an additional configured interface. *other_interface* is the name that is assigned to the interface.

'*other_interface*' total
The total number of worker threads in the thread pool that is used by an additional interface named *other_interface*.

The following example shows the output from a **stats get** command for the pdweb.threads component. The example assumes that no additional WebSEAL interface is configured:

```
#pdadmin> server task default-webseald-instance stats get pdweb.threads
active   : 0
total    : 50
'default' active : 0
'default' total : 50
```

Plug-in for Web Servers components and activity types

The following statistics components are available to Plug-in for Web Servers instances:

- pdwebpi.authn
- pdwebpi.authz
- pdwebpi.vhost.#
- pdwebpi.sescache
- pdwebpi.threads

pdwebpi.authn component

The pdwebpi.authn statistics component gathers information about plug-in authentication. The following list describes the types of available information:

pass The total number of successful authentications.

fail The total number of failed authentications.

pwd exp
The total number of authentication attempts made with an expired password.

max The maximum time for a single authentication process.

avg The average time for a single authentication process.

total The total time for all authentication processing.

The following example shows the output from a **stats get** command for the pdwebpi.authn component:

```
#pdadmin> server task PDWebPI-instance stats get pdwebpi.authn

pass     : 2
fail     : 1
pwd exp  : 0
max      : 0.178
avg      : 0.029
total    : 0.382
```

pdwebpi.authz component

The pdwebpi.authz statistics component gathers information about plug-in authorization. The following list describes the types of available information:

pass The total number of successful authorization requests. That is, the total number of resources that were successfully accessed.

fail The total number of failed authorization requests.

The following example shows the output from a **stats get** command for the pdwebpi.authz component:

```
#pdadmin> server task PDWebPI-instance stats get pdwebpi.authz

pass     : 2
fail     : 1
```

pdwebpi.sescache component

The pdwebpi.sescache component gathers statistics that are related to the plug-in session credential cache. This component gathers the following activity information:

hit The number of requests where a cache entry for a user was referenced successfully. That is, the number of requests that resulted in a session cache hit.

miss The number of requests that missed a session cache hit.

add The number of cache entries that were added to the session cache.

del The number of cache entries that were deleted from the session cache.

inactive
The number of times where a cache entry hit the inactivity timeout.

lifetime
The number of times where a cache entry hit the lifetime timeout.

expired
The number of times that a “least recently used” cache entry was deleted from the session cache to make room for a new cache entry.

The following example shows the output from a **stats get** command for the pdwebpi.sescache component:

```
#pdadmin> server task PDWebPI-instance stats get pdwebpi.sescache

hit         : 0
miss        : 0
add         : 0
del         : 0
inactive    : 0
lifetime    : 0
expired     : 0
```

pdwebpi.threads component

The pdwebpi.threads statistics component gathers information about plug-in worker thread activity. This component is always enabled by default and cannot be disabled. The following list describes the types of available information:

active The total number of active worker threads that are handling requests.

total The total number of configured worker threads.

The following example shows the output from a **stats get** command for the pdwebpi.threads component:

```
#pdadmin> server task PDWebPI-instance stats get pdwebpi.threads

active   : 0
total    : 50
```

pdwebpi.vhost.# component

The pdwebpi.vhost.# statistics component gathers information about configured virtual hosts. The following list describes the types of available information:

[/] The actual virtual host name (listed as the number in the command)

reqs The total number of requests routed across this virtual host

max The maximum time consumed by a single request across this virtual host

avg The average time consumed by a single request across this virtual host

total The total time consumed by requests across this virtual host

The following example shows the output from a **stats get** command for the pdwebpi.vhost.1 component:

```
#pdadmin> server task PDWebPI-instance stats get pdwebpi.vhost.1

[/]
reqs    : 0
max     : 0.000
total   : 0.000
```

Part 6. Audit events

Chapter 22. XML output of native audit events

When you use native Security Access Manager auditing, audit events are captured in the audit trail in a standard format with the Extensible Markup Language (XML) elements. XML is only an intermediary step to delivering a presentation view of the data. The XML file is in ASCII format and can be read directly or passed to other external parsing engines for further analysis.

DTD intermediate format

As an audit administrator, you are expected to select and extract events according to your own criteria. This activity might include reformatting each event by applying an appropriate Document Type Definition (DTD) or schema for the analysis tool that you are using. The DTD is an intermediate format that provides a description of the data that can be captured.

Data blocks and output elements

An entire audit trail does not represent a single XML document. Each audit event within the file is written as an isolated XML data block. Each data block conforms to the rules of standard XML syntax.

Sample authorization event

For example, the following data block is an audit record for getting user authorization credentials:

```
<event rev="1.2">
  <date>2005-11-14-16:25:08.341+00:00I-----</date>
  <outcome status="0">0</outcome>
  <originator blade="pdmgrd">
    <component rev="1.2">azn</component>
    <action>0</action>
    <location>phaedrus</location>
  </originator>
  <accessor name="">
    <principal auth="IV_LDAP_V3.0">sec_master</principal>
  </accessor>
  <target resource="3">
    <object>IV_LDAP_V3.0:sec_master</object>
  </target>
  <data>azn_id_get_creds</data>
</event>
```

Sample resource access event

For example, the following data block is an audit record for an HTTP request:

```
<event rev="1.2">
  <date>2005-10-02-22:01:36.187-04:00I-----</date>
  <outcome status="953091111" reason="unauthorized">1</outcome>
  <originator blade="webseald" instance="default">
    <component rev="1.2">http</component>
    <event_id>109</event_id>
    <action>1</action>
    <location>cmd.wma.ibm.com</location>
  </originator>
  <accessor name="unauthenticated">
    <principal auth="IV_UNAUTH_V3.0" domain="Default">Unauthenticated</principal>
    <user_location>9.54.83.206</user_location>
    <user_location_type>IPV4</user_location_type>
  </accessor>
  <target resource="5">
    <object>/</object>
    <object_nameinapp>HTTP://cmd.wma.ibm.com:80/</object_nameinapp>
  </target>
  <resource_access>
    <action>httpRequest</action>
    <httpurl>HTTP://cmd.wma.ibm.com:80/</httpurl>
    <httpmethod>GET</httpmethod>
    <httpresponse>200</httpresponse>
  </resource_access>
  <data>
    GET HTTP://cmd.wma.ibm.com:80/ HTTP/1.0
    1970
    Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
  </data>
</event>
```

Sample successful authentication events

For example, the following data block is an audit record for a successful authentication:

```
<event rev="1.2">
  <date>2005-10-02-21:59:31.980-04:00I-----</date>
  <outcome status="0">0</outcome>
  <originator blade="webseald" instance="default">
    <component rev="1.4">authn</component>
    <event_id>101</event_id>
    <action>0</action>
    <location>cmd.wma.ibm.com</location>
  </originator>
  <accessor name="">
    <principal auth="IV_LDAP_V3.0" domain="Default">testuser268</principal>
    <name_in_rgy>cn=testuser268,dc=ibm,dc=com</name_in_rgy>
    <session_id>56a701a4-33b1-11da-a8d3-00096bc369d2</session_id>
    <user_location>9.54.83.206</user_location>
    <user_location_type>IPV4</user_location_type>
  </accessor>
  <target resource="7">
    <object></object>
  </target>
  <authntype>formsPassword</authntype>
  <data></data>
</event>
```

Sample failed authentication events

For example, the following data block is an audit record for a failed authentication:

```
<event rev="1.2">
  <date>2005-10-02-21:59:31.977-04:00I-----</date>
  <outcome status="320938184" reason="authenticationFailure">1</outcome>
  <originator blade="webseald" instance="default">
    <component rev="1.4">authn</component>
    <event_id>101</event_id>
    <action>0</action>
    <location>cmd.wma.ibm.com</location>
  </originator>
  <accessor name="">
    <principal auth="" domain="">testuser335</principal>
    <user_location>9.54.83.206</user_location>
    <user_location_type>IPV4</user_location_type>
  </accessor>
  <target resource="7">
    <object></object>
  </target>
  <authntype>formsPassword</authntype>
  <data>
    Password Failure: testuser335
  </data>
</event>
```

Sample authentication terminate event

For example, the following data block is an audit record for the termination of an authentication:

```
<event rev="1.2">
  <date>2005-10-04-11:45:27.487-04:00I-----</date>
  <outcome status="0">0</outcome>
  <originator blade="webseald" instance="default">
    <component rev="1.4">authn</component>
    <event_id>103</event_id>
    <action>103</action>
    <location>cmd.wma.ibm.com</location>
  </originator>
  <accessor name="">
    <principal auth="IV_LDAP_V3.0" domain="Default">testuser1</principal>
    <name_in_rgy>cn=testuser1,dc=ibm,dc=com</name_in_rgy>
    <session_id>e005b3ae-34ed-11da-a016-00096bc369d2</session_id>
    <user_location>9.65.85.162</user_location>
    <user_location_type>IPV4</user_location_type>
  </accessor>
  <target resource="7">
    <object></object>
  </target>
  <authntype>formsPassword</authntype>
  <terminateinfo>
    <terminatereason>userLoggedOut</terminatereason>
  </terminateinfo>
  <data></data>
</event>
```

XML output elements

Table 46 on page 228 describes the XML output elements that are possible by using the default Security Access Manager DTD elements. If you create your own DTD, each element must represent the events that you selected and extracted according to your own criteria.

Table 46. Names and descriptions for XML output elements

Output element name	Description

`<event>`
`...`
`</event>`
	Auditing event. Each auditing event captures the result of an action. A principal attempts an action on a target object.
	The `event` element can include the following elements:
	• `date`
	• `outcome`
	• `originator`
	• `accessor`
	• `target`
	• `resource_access` (for resource access events)
	• `authntype` (for authentication events)
	• `terminationinfo` (for authentication terminate events)
	• `data`
	Because Security Access Manager auditing uses a standard record format, not all elements are relevant to each event that is recorded. Fields that are not relevant for a particular event might contain a default value.
	The `event` element can include the following attribute:
	• **rev**
	Example:
	`<event rev="1.2">`
	`  <date>2003-11-14-16:25:08.341+00:00I-----</date>`
	`  <outcome status="0">0</outcome>`
	`  ...`
	`</event>`

`<date>`
`...`
`</date>`
	Current date and timestamp. The date element has the following format:
	yyyy-*mm*-*dd*-*hh*:*mm*:*ss*.*xxx*-*xx*:*xx*I-----
	Where:
	yyyy-*mm*-*dd*
	Relates to the year (*yyyy*), the month (*mm*), and the day (*dd*).
	hh:*mm*:*ss*
	Relates to hours (*hh*), minutes (*mm*), and seconds (*ss*).
	xxx-*xx*:*xx***I**
	Refers to the time zone.
	Example:
	`<event rev="1.2">`
	`<date>2005-11-14-16:25:08.341+00-----</date>`
	`...`
	`</event>`

Table 46. Names and descriptions for XML output elements (continued)

Output element name	Description

`<outcome>`
`...`
`</outcome>`
	Outcome of the event. The `outcome` element can be one of the following values:
	0 Success
	1 Failure
	2 Pending
	3 Unknown
	The following information is captured in a common format header of the audit record:
	• The outcome.
	• The action.
	• The credentials for the principal.
	• The target object.
	This element can include the following attributes:
	• **status**
	• **reason**
	Example of a failed event:
	`<outcome status="320938184" reason="authenticationFailure">`
	`  1`
	`</outcome>`
	For information about the contents of the **status** attribute, use the **`errtext`** command. The command provides the error message that is associated with the status code (`320938184`) of a failed event. If the error is not identified by the **`errtext`** command, the error did not originate in Security Access Manager. See your third-party documentation for more status code definitions.
	For information about the contents of the **reason** attribute, see "Outcome output for failures" on page 252.
	Example of a successful event:
	`<event rev="1.2">`
	`...`
	`<outcome status="0">0</outcome>`
	`...`
	`</event>`

Table 46. Names and descriptions for XML output elements (continued)

Output element name	Description
<originator> ... </originator>	Server that originated the event being logged. The originator element can include the following elements: • component • event_id • action • location The originator element can include the following attributes: • **blade** • **instance** The **blade** attributes represents the server that originated the event. For example, pdmgrd is the Security Access Manager policy server, webseald is the Security Access Manager WebSEAL server. The **instance** attribute applies to WebSEAL and represents the name of the instance. Example: <event rev="1.2"> ... **<originator blade="webseald">** <component rev="1.4">authn</component> <event_id>101</event_id> <action>0</action> <location>cmd.wma.ibm.com</location> **</originator>** ... </event>
<component> ... </component>	Audit events, categorized by the server functionality that generates them. Some functionality is common across Security Access Manager servers while other functionality is server-specific. The component element can be one of the following values: **authz** or **azn** Captures authorization events. **authn** Captures authentication events. **mgmt** Captures management events. **http** Captures WebSEAL HTTP events. See the *IBM Security Access Manager for Web WebSEAL Administration Guide* for more information about this value. The component element can contain the **rev** attribute. Example: <originator blade="webseald"> **<component rev="1.4">authn</component>** <event_id>101</event_id> <action>0</action> <location>cmd.wma.ibm.com</location> </originator>

Table 46. Names and descriptions for XML output elements (continued)

Output element name	Description

`<event_id>`
`...`
`</event_id>`
	The category of the event ID. The `event_id` element can be one of the following values:
	101 Login
	102 Password change
	103 Logout
	104 Authenticate
	105 Step-up
	106 Re-authentication
	107 Credentials refresh
	108 Authorization check
	109 Resource access
	110 Get credentials
	111 Modify credentials/combine credentials
	112 Get credentials from pac
	113 Get pac
	114 Get entitlements
	115 Runtime start
	116 Runtime stop
	117 Runtime audit start
	118 Runtime audit stop
	119 Runtime audit level change
	120 Runtime statistic
	121 Runtime heartbeat up
	122 Runtime heartbeat down
	123 Runtime lost contact
	124 Runtime contact restored
	125 Runtime monitor
	126 Switch-user login
	127 Switch-user logout
	
	Example:
	`<originator blade="webseald">`
	   `<component rev="1.4">authn</component>`
	   `<event_id>101</event_id>`
	   `<action>0</action>`
	   `<location>cmd.wma.ibm.com</location>`
	`</originator>`

Table 46. Names and descriptions for XML output elements (continued)

Output element name	Description
<action> ... </action>	Audit record action code, which can be for one of the following groups of events: **Authentication or authorization events** Audit records for authentication or authorization events contain one of the following event action codes: **0** Authentication or authorization events **1** Change password events **2** WebSEAL events **Management events** Audit records for management events contain an action code that identifies the **pdadmin** utility. For example, the <action>13702</action> action code relates to the POP_MODIFY action for the **pop modify** command. See "Action codes for management commands" on page 246, which relates the action code reference number for each command. A common format header of the audit record captures information about: • The action. • The credentials of the principal. • The target object. • The outcome. Example: <originator blade="webseald"> <component rev="1.4">authn</component> <event_id>101</event_id> **<action>0</action>** <location>cmd.wma.ibm.com</location> </originator>

<location>
...
</location>
	The host name (location) of the machine. If there is no host name specified, a notation of "location not specified" is substituted in the location element.
	Example:
	<originator blade="webseald">
	   <component rev="1.4">authn</component>
	   <event_id>101</event_id>
	   <action>0</action>
	   <location>cmd.wma.ibm.com</location>
	</originator>

Table 46. Names and descriptions for XML output elements (continued)

<table>
<tr><th>Output element name</th><th>Description</th></tr>
<tr><td>

```
<accessor>
...
</accessor>
```

</td><td>

The name of the user that caused the event. If there is no user name specified, a notation of “name="user not specified"” or “name=""” is substituted in the accessor element.

The accessor element can include the following elements:

- principal
- name_in_rgy (for authenticated users)
- session_id (for authenticated users)
- principal
- user_location
- user_location_type

The accessor element includes the **name** attribute.

The following example shown the accessor element for an unauthenticated user:

```
<event rev="1.2>
  ...
  <accessor name="unauthenticated">
    <principal auth="IV_UNAUTH_V3.0" domain="Default">
      testuser2
    </principal>
    <user_location>9.65.85.162</user_location>
    <user_location_type>IPV4</user_location_type>
  </accessor>
  ...
</event>
```

The following example shown the accessor element for an authenticated user:

```
<event rev="1.2>
  ...
  <accessor name="">
    <principal auth="IV_LDAP_V3.0" domain="Default">
      testuser2
    </principal>
    <name_in_rgy>
      cn=testuser1,dc=ibm,dc=com
    </name_in_rgy>
    <session_id>
      e005ba3-34ed-11da-a016-00096bc369d
    </session_id>
    <user_location>9.65.85.162</user_location>
    <user_location_type>IPV4</user_location_type>
  </accessor>
  ...
</event>
```

</td></tr>
</table>

Table 46. Names and descriptions for XML output elements (continued)

Output element name	Description

`<principal>`
`...`
`</principal>`
	User authorization credentials. Generally each event captures the result of an action that a user (principal) attempts on a target object. If there is no user name specified, a notation of "`auth="invalid"`" is substituted in the `principal` element.
	
	The `principal` element can contain the following attributes:
	• **auth**
	• **domain**
	
	To determine the actual authentication method, use the data in the `authntype` element.
	
	A common format header of the audit record captures information about:
	• The credentials of the principal.
	• The action.
	• The target object.
	• The outcome.
	
	Example:
	`<accessor name="">`
	`  <principal auth="IV_LDAP_V3.0" domain="Default">`
	`    testuser2`
	`  </principal>`
	`  <name_in_rgy>`
	`    cn=testuser1,dc=ibm,dc=com`
	`  </name_in_rgy>`
	`  <session_id>`
	`    e005ba3-34ed-11da-a016-00096bc369d`
	`  </session_id>`
	`  <user_location>9.65.85.162</user_location>`
	`  <user_location_type>IPV4</user_location_type>`
	`  </accessor>`

`<name_in_rgy>`
`...`
`</name_in_rgy>`
	The name in the registry for the user.
	
	Example:
	`<accessor name="">`
	`  <principal auth="IV_LDAP_V3.0" domain="Default">`
	`    testuser2`
	`  </principal>`
	`  <name_in_rgy>`
	`    cn=testuser1,dc=ibm,dc=com`
	`  </name_in_rgy>`
	`  <session_id>`
	`    e005ba3-34ed-11da-a016-00096bc369d`
	`  </session_id>`
	`  <user_location>9.65.85.162</user_location>`
	`  <user_location_type>IPV4</user_location_type>`
	`</accessor>`

Table 46. Names and descriptions for XML output elements (continued)

Output element name	Description

<session_id>
...
</session_id>
	The session ID that is associated with this session. This ID can be used to trace a series of events back to the authentication data that was initially provided by the user. For example, the data in the session_id element could be used to determine when a user logged in and when a user logged out.
	Example:
	<accessor name="">
	  <principal auth="IV_LDAP_V3.0" domain="Default">
	    testuser2
	  </principal>
	  <name_in_rgy>
	    cn=testuser1,dc=ibm,dc=com
	  </name_in_rgy>
	  <session_id>
	    e005ba3-34ed-11da-a016-00096bc369d
	  </session_id>
	  <user_location>9.65.85.162</user_location>
	  <user_location_type>IPV4</user_location_type>
	</accessor>

<user_location>
...
</user_location>
	The IP address in IPv4 or IPv6 format.
	Example:
	<accessor name="">
	  <principal auth="IV_LDAP_V3.0" domain="Default">
	    testuser2
	  </principal>
	  <name_in_rgy>
	    cn=testuser1,dc=ibm,dc=com
	  </name_in_rgy>
	  <session_id>
	    e005ba3-34ed-11da-a016-00096bc369d
	  </session_id>
	  <user_location>9.65.85.162</user_location>
	  <user_location_type>IPV4</user_location_type>
	</accessor>

<user_location_type>
...
</user_location_type>
	The format of the data in the user_location element. Values are:
	• IPV4
	• IPV6
	Example:
	<accessor name="">
	  <principal auth="IV_LDAP_V3.0" domain="Default">
	    testuser2
	  </principal>
	  <name_in_rgy>
	    cn=testuser1,dc=ibm,dc=com
	  </name_in_rgy>
	  <session_id>
	    e005ba3-34ed-11da-a016-00096bc369d
	  </session_id>
	  <user_location>9.65.85.162</user_location>
	  <user_location_type>IPV4</user_location_type>
	</accessor>

Table 46. Names and descriptions for XML output elements (continued)

Output element name	Description

`<target>`
`...`
`</target>`
	Target information. The `target` element can include the following elements:
	• `object`
	• `object_nameinapp`
	• `process`
	• `azn`
	The `target` element includes the **resource** attribute, which represents a broad categorization of the target object: The **resource** attribute can be one of the following values:
	0 AUTHORIZATION
	1 PROCESS
	2 TCB
	3 CREDENTIAL
	5 GENERAL
	6 APPLICATION
	7 AUTHENTICATION
	Examples:
	`<target resource="7">`
	`  <object></object>`
	`</target>`
	`<target resource="3">`
	`  <object>IV_LDAP_V3.0:sec_master</object>`
	`</target>`

`<object>`
`...`
`</object>`
	Target object. Authorization audit records can be captured when a target object in the policy database (protected object space) has a POP attached to it. The POP must enable audit functionality. For example:
	`<object>/Management</object>`
	A common format header of the audit record captures information about:
	• The target object.
	• The action.
	• The user credentials.
	• The outcome.
	Example:
	`<target resource="3">`
	`  <object>IV_LDAP_V3.0:sec_master</object>`
	`</target>`

Table 46. Names and descriptions for XML output elements (continued)

Output element name	Description

`<azn>`
`...`
`</azn>`
	Authorization service information. The authorization service:
	• Checks the access permissions on the target requested object.
	• Compares these access permissions with the capabilities of the requesting user.
	The `azn` element can include the following elements:
	• `perm`
	• `result`
	• `qualifier`
	`<target resource="3">`
	`  ...`
	`  `**`<azn>`**
	`    <perm>64</perm>`
	`    <result>0</result>`
	`    <qualifier>0</qualifier>`
	`  `**`</azn>`**
	`  ...`
	`</target>`

`<perm>`
`...`
`</perm>`
	Set of controls (permissions) that specifies the conditions necessary to complete certain operations on that resource. The permission can be specified in this element by using either the binary number such as `<perm>64</perm>` or the letters for the specified action permissions such as `<perm>Tr</perm>`.
	Example:
	`<target resource="3">`
	`  ...`
	`  <azn>`
	`    `**`<perm>64</perm>`**
	`    <result>0</result>`
	`    <qualifier>0</qualifier>`
	`  </azn>`
	`  ...`
	`</target>`

`<result>`
`...`
`</result>`
	Results of the authorization service check.
	Example:
	`<target resource="3">`
	`  ...`
	`  <azn>`
	`    <perm>64</perm>`
	`    `**`<result>0</result>`**
	`    <qualifier>0</qualifier>`
	`  </azn>`
	`  ...`
	`</target>`

Table 46. Names and descriptions for XML output elements (continued)

Output element name	Description

<qualifier>
...
</qualifier>
	Qualifier information.
	Example:
	<target resource="3">
	  ...
	  <azn>
	    <perm>64</perm>
	    <result>0</result>
	    <qualifier>0</qualifier>
	  </azn>
	  ...
	</target>

<process> ... </process>	Type of process. The process element can include the following elements: • pid (process ID) • uid (user ID) • eid (effective user ID) • gid (group ID) • egid (effective group ID) The process element includes the **architecture** attribute, which is one of the following values: **0** For AIX, Linux, and Solaris operating systems. **1** For Windows operating systems. Example: **<process architecture="0">** ... <pid></pid> **</process>**
<pid></pid> <eid></eid> <uid></uid> <gid></gid> <egid></egid>	The identifier of the process, which is contained in one of the following elements: **pid** Process ID **eid** Effective user ID **uid** User ID **gid** Group ID **egid** Effective group ID Example: <process architecture="0"> ... **<pid>3899</pid>** </process>
<policy> ... </policy>	The security policy information. The policy element can include the following elements: • name • type • descr Example of name element for policy element: **<policy>** <name>real-traders-only</name> <type>rule</type> **</policy>**

Table 46. Names and descriptions for XML output elements (continued)

Output element name	Description

`<name>`
`...`
`</name>`
	Name of the policy attribute that you want to audit. The name matches the name that you specified in a list of attributes in the `[aznapi-configuration]` stanza of the appropriate configuration file. For example:
	`[aznapi-configuration]`
	`audit-attribute = real-traders-only`
	
	Example:
	`<policy>`
	  `<name>real-traders-only</name>`
	  `<type>rule</type>`
	`</policy>`

`<type>`
`...`
`</type>`
	Type of security policy being audited. The `type` element can contain the following values:
	• `ACL`
	• `POP`
	• `rule`
	
	Example:
	`<policy>`
	  `<name>traders-pop</name>`
	  `<type>POP</type>`
	`</policy>`

`<descr>`
`...`
`</descr>`
	Description of the security policy. This element is empty if no description was created for the policy.
	
	Example:
	`<policy><name>traders-acl</name>`
	  `<type>ACL</type>`
	  `<descr>traders that have ACL security policies</descr>`
	`</policy>`

`<attribute>`
`...`
`</attribute>`
	The container for the characteristics of the access decision information (ADI) attribute to audit. An attribute can establish accountability by providing information to help identify potentially inappropriate access of assets. You can grant or deny access based on rules applied to attributes.
	
	The `attribute` element can include the following elements:
	• `name`
	• `source`
	• `type`
	• `value`
	
	Example:
	`<attribute>`
	  `<name>tagvalue_su-admin</name>`
	  `<source>cred</source>`
	  `<type>string</type>`
	  `<value>test_customer_service_rep_1</value>`
	`</attribute>`

Table 46. Names and descriptions for XML output elements (continued)

Output element name	Description
<name> ... </name>	Name of the ADI to audit. This ADI can be for auditing either a user credential if for the authn component or an app_context if for an azn component. The name of the authorization attribute matches the name that you specified in a list of attributes in the [aznapi-configuration] stanza of the appropriate configuration file. For example: [aznapi-configuration] audit-attribute = AZN_CRED_AUTH_METHOD Example of name element for the attribute element: <attribute> **<name>AZN_CRED_AUTH_METHOD</name>** <source>credADI</source> <type>string</type> <value>su-forms</value> </attribute>
<source> ... </source>	The source event. The source element can contain one of the following values: **cred** Applies to any Security Access Manager component. **app** Applies only to an authorization (azn) component. **credADI** Applies only to the authorization (azn) component when evaluating a Boolean rule. **appADI** Applies only to the authorization (azn) component when evaluating a Boolean rule. **engineADI** Applies only to the authorization (azn) component when evaluating a Boolean rule. **dynADI** Applies only to the authorization (azn) component when evaluating a Boolean rule. If the ADI attribute is multi-valued, a separate attribute element is written for each value. Example: <attribute> <name>AZN_CRED_AUTH_METHOD</name> **<source>credADI</source>** <type>string</type> <value>su-forms</value> </attribute>

Table 46. Names and descriptions for XML output elements (continued)

Output element name	Description
<type> ... </type>	Type of security policy that is being audited. The type element can contain one of the following values: • string • ulong • pobj If <type>pobj</type>, the value is the name of the protected object. Example: <attribute> <name>AZN_CRED_AUTH_METHOD</name> <source>credADI</source> **<type>string</type>** <value>su-forms</value> </attribute>

<value>
...
</value>
	Value for the aznAPI attribute. If the ADI attribute is multi-valued, then a separate attribute element is written for each value.
	Example:
	<attribute>
	  <name>AZN_CRED_AUTH_METHOD</name>
	  <source>credADI</source>
	  <type>string</type>
	  <value>su-forms</value>
	</attribute>

<resource_access>
...
</resource_access>
	Example:
	<event rev="1.2">
	  ...
	  <resource_access>
	    <action>httpRequest</action>
	    <httpurl>HTTP://cmd.wma.ibm.com:80/</httpurl>
	    <httpmethod>GET</httpmethod>
	    <httpresponse>200</httpresponse>
	  </resource_access>
	  ...
	</event>

<action>
...
</action>
	Example:
	<event rev="1.2">
	  ...
	  <resource_access>
	    <action>httpRequest</action>
	    <httpurl>HTTP://cmd.wma.ibm.com:80/</httpurl>
	    <httpmethod>GET</httpmethod>
	    <httpresponse>200</httpresponse>
	  </resource_access> ...
	</event>

Table 46. Names and descriptions for XML output elements (continued)

<table>
<tr><th>Output element name</th><th>Description</th></tr>
<tr><td><pre>
<httpurl>
...
</httpurl>
</pre></td><td>Example:
<pre>
<event rev="1.2">
  ...
  <resource_access>
    <action>httpRequest</action>
    <httpurl>HTTP://cmd.wma.ibm.com:80/</httpurl>
    <httpmethod>GET</httpmethod>
    <httpresponse>200</httpresponse>
  </resource_access> ...
</event>
</pre></td></tr>
<tr><td><pre>
<httpmethod>
...
</httpmethod>
</pre></td><td>Example:
<pre>
<event rev="1.2">
  ...
  <resource_access>
    <action>httpRequest</action>
    <httpurl>HTTP://cmd.wma.ibm.com:80/</httpurl>
    <httpmethod>GET</httpmethod>
    <httpresponse>200</httpresponse>
  </resource_access> ...
</event>
</pre></td></tr>
<tr><td><pre>
<httpresponse>
...
</httpresponse>
</pre></td><td>Example:
<pre>
<event rev="1.2">
  ...
  <resource_access>
    <action>httpRequest</action>
    <httpurl>HTTP://cmd.wma.ibm.com:80/</httpurl>
    <httpmethod>GET</httpmethod>
    <httpresponse>200</httpresponse>
  </resource_access> ...
</event>
</pre></td></tr>
</table>

Table 46. Names and descriptions for XML output elements (continued)

Output element name	Description

`<authntype>`
`...`
`</authntype>`
	The type of authentication that the user completed. The following strings are authentication types that are associated with WebSEAL and Plug-in for Web Servers:
	itamFailoverCookie
	Failover cookie
	itamCDSSO
	WebSEAL or Plug-in for Web Servers authentication using cross domain single-sign on (CDSSO)
	itamECSSO
	WebSEAL or Plug-in for Web Servers authentication using e-Community single-sign on (ECSSO)
	certificate
	SSL certificate authentication
	twoFactor
	WebSEAL or Plug-in for Web Servers using token authentication
	formsPassword
	Password authentication using an HTML form
	basicAuthRFC2617
	Password authentication using HTTP Basic Authentication (BA)
	passwordOther
	Password authentication using an undetermined mechanism
	itamHTTPHeader
	WebSEAL or Plug-in for Web Servers using HTTP header authentication
	itamIPAddress
	WebSEAL or Plug-in for Web Servers using IP address-based authentication
	kerberos
	WebSEAL or Plug-in for Web Servers using SPNEGO authentication
	itamEAI
	WebSEAL or Plug-in for Web Servers using external authentication interface (EAI) authentication
	itamIVCreds
	Plug-in for Web Servers authentication using the IV_CREDS header
	itamIVUser
	Plug-in for Web Servers authentication using the IV_USER header
	tokenLTPA
	Plug-in for Web Servers authentication using a lightweight third-party authentication (LTPA) token
	ntlm Plug-in for Web Servers using NTLM authentication
	itamWebServerAuthentication
	Plug-in for Web Servers authentication that is provided by the hosting Web server
	
	Example:
	`<event rev="1.2">`
	`  ...`
	`  <authntype>formsPassword</authntype>`
	`  ...`
	`</event>`

Table 46. Names and descriptions for XML output elements (continued)

Output element name	Description

`<terminateinfo>`
`...`
`</terminateinfo>`
	Contains information about why a session ended. The `terminateinfo` element contains the `terminatereason` element.
	Example:
	`<event rev="1.2">`
	`  ...`
	`  `**`<terminateinfo>`**
	`    <terminatereason>userLoggedOut</terminatereason>`
	`  `**`</terminateinfo>`**
	`  ...`
	`</event>`

`<terminatereason>`
`...`
`</terminatereason>`
	The reason why the session ended. The following values are possible:
	idleTimeout
	The session timed out because the user was inactive.
	sessionExpired
	The session timed out because the user was logged in for too long.
	sessionDisplaced
	The session ended because another user with the same user ID logged in.
	sessionTerminatedByAdmin
	The session ended because an administrator logged out the user.
	userLoggedOut
	The session ended because the user logged out.
	reathLockOut
	The session ended because the user did not authenticate again.
	Example:
	`<terminateinfo>`
	`  `**`<terminatereason>userLoggedOut</terminatereason>`**
	`</terminateinfo>`

Table 46. Names and descriptions for XML output elements (continued)

<table>
<tr><th>Output element name</th><th>Description</th></tr>
<tr><td>
<pre>
<data>
...
</data>
</pre>
</td><td>

Event-specific data. The `data` element can contain the `audit` element.

Additional event-specific information is recorded in a free format data area at the end of the event record. For failed authentication attempts, "Data output for errors" on page 251 provides details about the data information that is returned.
Note: Decoding the meaning of certain data values in the record might require an advanced knowledge of the Security Access Manager code and architecture.

Command arguments are listed in the `data` element of the event record in their internal format. For example:

`<data>azn_id_get_creds</data>`

Commands that do not result in an effective state change (**`list`** and **`show`**) are never captured.

Examples:

<pre>
•

  <event>
    ...
    <b><data></b>
    POST /pkmspasswd.form HTTP/1.1 0 Mozilla/4.0
    (compatible; MSIE 6.0; Windows NT 5.0)
    https://c03comcrit2.somecompany.com/pkmspasswd
    <b></data></b>
  </event>
•

  <b><data></b>
    "2019"
    "1002"
    "pop1"
    "0"
    ""
  <b></data></b>
</pre>
</td></tr>
</table>

Table 46. Names and descriptions for XML output elements (continued)

Output element name	Description

`<audit/>`	Beginning and ending of an audit event. The `audit` element can include the **event** attribute, which can be one of the following values:
	• `Start`
	• `Stop`
	Example:
	`<event rev="1.2">`
	`  ...`
	`  <data>`
	`    <audit event="Start"/>`
	`  </data>`
	`</event>`
	`...`
	`<event rev="1.2">`
	`  ...`
	`  <data>`
	`    <audit event="Stop"/>`
	`  </data>`
	`</event>`

Action codes for management commands

The action code identifies one of the **pdadmin** management commands. The tables in this section relate the action code reference number for each management command. For example, the action code 13702 relates to the POP_MODIFY action command. In other words, the **pdadmin pop modify** command.

Command arguments are listed in the data section of the event record in their internal format. Commands that do not result in an effective change of state of the database (such as the **list** and **show** commands) are never captured.

Table 47 maps the action codes to the management commands.

Table 47. Mapping of action codes to management commands

Action code	Management command
13000	ACL_LIST
13001	ACL_GET
13002	ACL_SET_LEGACY
13003	ACL_DELETE
13005	ACL_FIND
13006	ACTION_LIST
13007	ACTION_SET
13008	ACTION_DELETE
13009	ACTION_GROUPLIST
13010	ACTION_GROUPCREATE
13011	ACTION_GROUPDELETE
13012	ACTION_LISTGROUP
13013	ACTION_CREATEGROUP

Table 47. Mapping of action codes to management commands (continued)

Action code	Management command
13014	ACTION_DELETEGROUP
13020	ACL_CREATE
13021	ACL_SET
13100	OBJ_GET
13101	OBJ_ACL_SET (deprecated)
13102	OBJ_GET_OBJ
13103	OBJSPC_CREATE
13104	OBJSPC_DELETE
13105	OBJSPC_LIST
13106	OBJ_CREATE
13107	OBJ_DELETE
13110	OBJ_MOD_SET_NAME
13111	OBJ_MOD_SET_DESC
13112	OBJ_MOD_SET_TYPE
13113	OBJ_MOD_SET_ISLF
13114	OBJ_MOD_SET_ISPOL
13115	OBJ_MOD_SET_ATTR
13116	OBJ_MOD_DEL_ATTR
13117	OBJ_MOD_DEL_ATTRVAL
13118	OBJ_SHOW_ATTR
13119	OBJ_LIST_ATTR
13120	ACL_ATTACH
13121	ACL_DETACH
13123	ACL_MOD_SET_ATTR
13124	ACL_MOD_DEL_ATTR
13125	ACL_MOD_DEL_ATTRVAL
13126	ACL_SHOW_ATTR
13127	ACL_LIST_ATTR
13128	POP_MOD_SET_ATTR
13129	POP_MOD_DEL_ATTR
13130	POP_MOD_DEL_ATTRVAL
13131	POP_SHOW_ATTR
13132	POP_LIST_ATTR
13133	OBJ_SHOW_ATTRS
13134	ACL_SHOW_ATTRS
13135	POP_SHOW_ATTRS
13136	OBJ_SHOW_V417
13137	OBJ_LIST
13138	OBJ_LISTANDSHOW_V417
13139	OBJ_EXISTS (deprecated)

Table 47. Mapping of action codes to management commands (continued)

Action code	Management command
13140	OBJ_ACCESS_CHECK
13141	OBJ_SHOW
13142	OBJ_LISTANDSHOW
13150	ACL_CREATE_ATTR (deprecated, see 13134)
13200	SERVER_GET
13201	SERVER_RESTORE
13202	SERVER_DELETE (deprecated)
13203	SERVER_LIST
13204	SERVER_PERFORMTASK
13205	SERVER_GETTASKLIST
13206	SERVER_REPLICATE
13207	SERVER_ACTION
13208	SERVER_STATUS_GET
13209	SERVER_ENABLE (deprecated)
13210	SERVER_DISABLE (deprecated)
13400	ADMIN_SHOWCONF
13401	USER_CREATE
13402	USER_IMPORT
13403	USER_MODDESC
13404	USER_MODPWD
13405	USER_MODAUTHMECH
13406	USER_MODACCVALID
13407	USER_MODPWDVALID
13408	USER_DELETE
13409	USER_SHOWGROUPS
13410	USER_SHOW
13411	USER_SHOWDN
13412	USER_LIST
13413	USER_LISTDN
13414	GROUP_CREATE
13415	GROUP_IMPORT
13416	GROUP_MODDESC
13417	GROUP_MODADD
13418	GROUP_MODREMOVE
13419	GROUP_DELETE
13420	GROUP_SHOW
13421	GROUP_SHOWDN
13422	GROUP_LIST
13423	GROUP_LISTDN
13424	GROUP_SHOWMEMB

Table 47. Mapping of action codes to management commands (continued)

Action code	Management command
13425	USER_MODGSOUSER
13426	USER_SET (deprecated)
13427	GROUP_SET (deprecated)
13428	GROUP_MODADD2
13500	GSO_RESOURCE_CREATE
13501	GSO_RESOURCE_DELETE
13502	GSO_RESOURCE_LIST
13503	GSO_RESOURCE_SHOW
13504	GSO_RESOURCE_CRED_CREATE
13505	GSO_RESOURCE_CRED_DELETE
13506	GSO_RESOURCE_CRED_MODIFY
13507	GSO_RESOURCE_CRED_LIST
13508	GSO_RESOURCE_CRED_SHOW
13509	GSO_RESOURCE_GROUP_CREATE
13510	GSO_RESOURCE_GROUP_DELETE
13511	GSO_RESOURCE_GROUP_ADD
13512	GSO_RESOURCE_GROUP_REMOVE
13513	GSO_RESOURCE_GROUP_LIST
13514	GSO_RESOURCE_GROUP_SHOW
13600	POLICY_SET_MAX_LOGIN_FAILURES
13601	POLICY_GET_MAX_LOGIN_FAILURES
13602	POLICY_SET_DISABLE_TIME_INTERVAL
13603	POLICY_GET_DISABLE_TIME_INTERVAL
13604	POLICY_SET_MAX_ACCOUNT_AGE
13605	POLICY_GET_MAX_ACCOUNT_AGE
13606	POLICY_SET_ACCOUNT_EXPIRY_DATE
13607	POLICY_GET_ACCOUNT_EXPIRY_DATE
13608	POLICY_SET_MAX_INACTIVITY_TIME
13609	POLICY_GET_MAX_INACTIVITY_TIME
13610	POLICY_GET_ACCOUNT_CREATION_DATE
13611	POLICY_GET_LAST_LOGIN_ATTEMPT_DATE
13612	POLICY_SET_MAX_PASSWORD_AGE
13613	POLICY_GET_MAX_PASSWORD_AGE
13614	POLICY_SET_MIN_PASSWORD_AGE
13615	POLICY_GET_MIN_PASSWORD_AGE
13616	POLICY_SET_MAX_PASSWORD_REPEATED_CHARS
13617	POLICY_GET_MAX_PASSWORD_REPEATED_CHARS
13618	POLICY_SET_MIN_PASSWORD_ALPHAS
13619	POLICY_GET_MIN_PASSWORD_ALPHAS
13620	POLICY_SET_MIN_PASSWORD_NON_ALPHAS

Table 47. Mapping of action codes to management commands (continued)

Action code	Management command
13621	POLICY_GET_MIN_PASSWORD_NON_ALPHAS
13622	POLICY_SET_MIN_PASSWORD_DIFFERENT_CHARS
13623	POLICY_GET_MIN_PASSWORD_DIFFERENT_CHARS
13624	POLICY_SET_PASSWORD_SPACES
13625	POLICY_GET_PASSWORD_SPACES
13626	POLICY_SET_MIN_PASSWORD_LENGTH
13627	POLICY_GET_MIN_PASSWORD_LENGTH
13628	POLICY_SET_MIN_PASSWORD_REUSE_TIME
13629	POLICY_GET_MIN_PASSWORD_REUSE_TIME
13630	POLICY_GET_PASSWORD_FAILURES
13631	POLICY_GET_LAST_PASSWORD_CHANGE_DATE
13632	POLICY_SET_NUMBER_WARN_DAYS
13633	POLICY_GET_NUMBER_WARN_DAYS
13634	POLICY_SET_PASSWORD_REUSE_NUM
13635	POLICY_GET_PASSWORD_REUSE_NUM
13636	POLICY_SET_TOD_ACCESS
13637	POLICY_GET_TOD_ACCESS
13638	POLICY_GET_ALL_POLICY
13639	POLICY_SET_MAX_CONCURRENT_WEB_SESSIONS
13640	POLICY_GET_MAX_CONCURRENT_WEB_SESSIONS
13700	POP_CREATE
13701	POP_DELETE
13702	POP_MODIFY
13703	POP_SHOW
13704	POP_LIST
13705	POP_ATTACH
13706	POP_DETACH
13707	POP_FIND
13800	CFG_CONFIG
13801	CFG_UNCONFIG
13802	CFG_RENEWCERT
13803	CFG_SETPORT
13804	CFG_SETLISTENING
13805	CFG_SETKEYRINGPWD
13806	CFG_SETSSLTIMEOUT
13807	CFG_SETAPPLCERT
13808	CFG_ADDREPLICA
13809	CFG_CHGREPLICA
13810	CFG_RMVREPLICA
13811	CFG_GETVALUE

Table 47. Mapping of action codes to management commands (continued)

Action code	Management command
13812	CFG_SETVALUE
13813	CFG_RMVVALUE
13814	CFG_SETSVRPWD
13900	DOMAIN_CREATE
13901	DOMAIN_DELETE
13902	DOMAIN_MODIFY_DESC
13903	DOMAIN_SHOW
13904	DOMAIN_LIST
13950	AUTHZRULE_CREATE
13951	AUTHZRULE_DELETE
13952	AUTHZRULE_MODIFYTEXT
13953	AUTHZRULE_MODIFYREASON
13954	AUTHZRULE_MODIFYDESC
13955	AUTHZRULE_SHOW
13956	AUTHZRULE_LIST
13957	AUTHZRULE_ATTACH
13958	AUTHZRULE_DETACH
13959	AUTHZRULE_FIND
13960	AUTHZRULE_MOD_SET_ATTR
13961	AUTHZRULE_MOD_DEL_ATTR
13962	AUTHZRULE_MOD_DEL_ATTRVAL
13963	AUTHZRULE_SHOW_ATTRS
13964	AUTHZRULE_SHOW_ATTR
13965	AUTHZRULE_LIST_ATTR

Authentication failures

The reason for authentication failure is included in two different locations in the authentication audit event:

- The `data` element
- The `outcome` element

Primarily, the `data` element is for compatibility with the earlier version of audit events. Later versions of audit events use the `outcome` element.

Data output for errors

Table 48 on page 252 lists the authentication error codes and the `data` output element structures that are returned when an authentication attempt fails.

Table 48. Authentication errors

Error type	Error code (in hex)	Error code (in decimal)	Generated XML

Password failure	132120c8	320938184	`<data>`
			`Password failure:` *`user`*
			`</data>`

Account lock-out	13212132	320938290	`<data>`
			`Account lock-out:` *`user`*
			`</data>`

General failure	All others	All others	`<data>`
			`<username>`*`user`*`</username>`
			`</data>`

Outcome output for failures

The `outcome` element provides more detailed information about the authentication failure. The following snippet of an audit event shows the `outcome` element:

```
<outcome status="320938184" reason="authenticationFailure">
```

The following list explains the meaning for the **reason** attribute of the `outcome` element:

accountDisabled
The account is disabled.

accountDisabledRetryViolation
The account was disabled because of a violation of the max-login-failures policy. The account was permanently disabled.

accountExpired
The account is expired or disabled.

accountLockedOutMaxLoginFail
The login failed because the account is temporarily disabled due to the max-login-failures policy.

authenticationFailure
General authentication failure, including incorrect password.

certificateFailure
Incorrect SSL certificate.

invalidUserName
Incorrect user name.

nextToken
Token authentication requires next token.

passwordExpired
The password expired and must be changed.

pinRequired
Token authentication requires a new PIN (personal identification number).

policyViolationMaxLotginsReached
Violation of the max-concurrent-web-session policy.

policyViolationTOD
Violation of the time-of-day policy.

userNameMismatch

Attempt at authentication or step-up authenticate failed because the user name that was provided did not match the previous user name.

Chapter 23. Elements by event types

This section lists the elements that are available for each common audit event type.

For each event type, this documentation provides a description of the event and a listing of the available element. For each available element, the table provides the element name, whether it is always in the event output, and its abbreviated XPath statement.

The abbreviated XPath statement is represented in one of the following ways:

element

element_type.element

When the representation is *element*, the full XPath statement would be:

```
CommonBaseEvent/extendedDataElements[@name='element']/values
```

When the representation is *element_type.element*, the full XPath statement would be:

```
CommonBaseEvent/extendedDataElements[@name='element_type']/children
[@name='element']/values
```

For detailed information about these elements and element types, see Chapter 24, “Reference information about elements and element types,” on page 289.

Elements for AUDIT_AUTHN events

This event type identifies authentication events.

The following table lists the elements that can be displayed in the output of an AUDIT_AUTHN event and their abbreviated XPath statements.

Table 49. Elements used in AUDIT_AUTHN events

Element	Always in output	Abbreviated XPath
action	No	`action`
auditMsg	No	`auditMsg`
auditMsgElement	No	Neither this element, nor its children, should be defined in the shredder configuration file.
auditTrailId	No	`auditTrailId`
authenProvider	No	`authenProvider`
authnType	Yes	`authnType`
authnTypeVersion	No	`authnTypeVersion`
endTime	No	`endTime [type='dateTime']`
extensionName	Yes	`"'AUDIT_AUTHN'"`
globalInstanceId	Yes	Not applicable. This value is an internal number that is not related to `#GLOBAL_ID`.
outcome	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditOutcomeType element type.
failureReason	No	`outcome.failureReason`

Table 49. Elements used in AUDIT_AUTHN events (continued)

Element	Always in output	Abbreviated XPath
majorStatus	No	`outcome.majorStatus`
minorStatus	No	`outcome.minorStatus`
result	Yes	`outcome.result`
partner	No	`partner`
progName	No	`progName`
registryInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the registryInfoType element type.
serverLocation	Yes	`registryInfo.serverLocation`
serverLocationType	Yes	`registryInfo.serverLocationType`
serverPort	Yes	`registryInfo.serverPort`
type	Yes	`registryInfo.type`
reporterComponentId	When different from sourceComponentId	Neither this element, nor its children, should be defined in the shredder configuration file. These elements are generated by the code. This container element uses the children of the auditComponentIdType element type.
sequenceNumber	Yes	Not applicable. This value is an internal number that is not related to `#RECORD_ID`.
sourceComponentId	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditComponentIdType element type.
application	Yes	`CommonBaseEvent/SourceComponentId/@application`
component	Yes	`CommonBaseEvent/SourceComponentId/@component`
componentIdType	Yes	`CommonBaseEvent/SourceComponentId/@componentIdType`
componentType	Yes	`CommonBaseEvent/SourceComponentId/@componentType`
executionEnvironment	No	`CommonBaseEvent/SourceComponentId/@executionEnvironment`
instanceId	No	`CommonBaseEvent/SourceComponentId/@instanceId`
location	Yes	`CommonBaseEvent/SourceComponentId/@location`
locationType	Yes	`CommonBaseEvent/SourceComponentId/@locationType`
processed	No	`CommonBaseEvent/SourceComponentId/@processed`
subComponent	Yes	`CommonBaseEvent/SourceComponentId/@subComponent`
threadId	No	`CommonBaseEvent/SourceComponentId/@threadId`
startTime	No	`startTime [type='dateTime']`
timestamp	Yes	`CommonBaseEvent/@creationTime`
userInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the userInfoType element type.
appUserName	Yes	`userInfo.appUserName`

Table 49. Elements used in AUDIT_AUTHN events (continued)

Element	Always in output	Abbreviated XPath
attributes	No	`userInfo.attributes`
callerList	No	`userInfo.callerList`
domain	No	`userInfo.domain`
location	No	`userInfo.location`
locationType	No	`userInfo.locationType`
realm	No	`userInfo.realm`
registryUserName	Yes	`userInfo.registryUserName`
sessionId	No	`userInfo.sessionId`
uniqueId	No	`userInfo.uniqueId`

Elements for AUDIT_AUTHN_CREDS_MODIFY events

This event type modifies credentials for a given user identity.

The following table lists the elements that can be displayed in the output of an AUDIT_AUTHN_CREDS_MODIFY event and their abbreviated XPath statements.

Table 50. Elements used in AUDIT_AUTHN_CREDS_MODIFY events

Element	Always in output	Abbreviated XPath
action	Yes	`action`
auditMsg	No	`auditMsg`
auditMsgElement	No	Neither this element, nor its children, should be defined in the shredder configuration file.
auditTrailId	No	`auditTrailId`
endTime	No	`endTime [type='dateTime']`
extensionName	Yes	`"'AUDIT_AUTHN_CREDS_MODIFY'"`
globalInstanceId	Yes	Not applicable. This value is an internal number that is not related to `#GLOBAL_ID`.
outcome	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditOutcomeType element type.
failureReason	No	`outcome.failureReason`
majorStatus	No	`outcome.majorStatus`
majorStatus	No	`outcome.minorStatus`
result	Yes	`outcome.result`
registryInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the registryInfoType element type.
serverLocation	Yes	`registryInfo.serverLocation`
serverLocationType	Yes	`registryInfo.serverLocationType`
serverPort	Yes	`registryInfo.serverPort`
type	Yes	`registryInfo.type`
reporterComponentId	When different from sourceComponentId	Neither this element, nor its children, should be defined in the shredder configuration file. These elements are generated by the code. This container element uses the children of the auditComponentIdType element type.

Table 50. Elements used in AUDIT_AUTHN_CREDS_MODIFY events (continued)

Element	Always in output	Abbreviated XPath
sequenceNumber	Yes	Not applicable. This value is an internal number that is not related to `#RECORD_ID`.
sourceComponentId	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditComponentIdType element type.
application	Yes	`CommonBaseEvent/SourceComponentId/@application`
component	Yes	`CommonBaseEvent/SourceComponentId/@component`
componentIdType	Yes	`CommonBaseEvent/SourceComponentId/@componentIdType`
componentType	Yes	`CommonBaseEvent/SourceComponentId/@componentType`
executionEnvironment	No	`CommonBaseEvent/SourceComponentId/@executionEnvironment`
instanceId	No	`CommonBaseEvent/SourceComponentId/@instanceId`
location	Yes	`CommonBaseEvent/SourceComponentId/@location`
locationType	Yes	`CommonBaseEvent/SourceComponentId/@locationType`
processed	No	`CommonBaseEvent/SourceComponentId/@processed`
subComponent	Yes	`CommonBaseEvent/SourceComponentId/@subComponent`
threadId	No	`CommonBaseEvent/SourceComponentId/@threadId`
startTime	No	`startTime [type='dateTime']`
timestamp	Yes	`CommonBaseEvent/@creationTime`
userInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the userInfoType element type.
appUserName	Yes	`userInfo.appUserName`
attributes	No	`userInfo.attributes`
callerList	No	`userInfo.callerList`
domain	No	`userInfo.domain`
location	No	`userInfo.location`
locationType	No	`userInfo.locationType`
realm	No	`userInfo.realm`
registryUserName	Yes	`userInfo.registryUserName`
sessionId	No	`userInfo.sessionId`
uniqueId	No	`userInfo.uniqueId`

Elements for AUDIT_AUTHN_MAPPING events

This event type records the mapping of principal and credentials where there are two user identities involved.

The following table lists the elements that can be displayed in the output of an AUDIT_AUTHN_MAPPING event and their abbreviated XPath statements.

Table 51. Elements used in AUDIT_AUTHN events

Element	Always in output	Abbreviated XPath
auditMsg	No	`auditMsg`
auditMsgElement	No	Neither this element, nor its children, should be defined in the shredder configuration file.
auditTrailId	No	`auditTrailId`
endTime	No	`endTime [type='dateTime']`
extensionName	Yes	`"'AUDIT_AUTHN_MAPPING'"`
globalInstanceId	Yes	Not applicable. This value is an internal number that is not related to `#GLOBAL_ID`.
mappedRealm	No	`mappedRealm`
mappedSecurityDomain	Yes	`mappedSecurityDomain`
mappedUserName	Yes	`mappedUserName`
originalRealm	No	`originalRealm`
originalSecurityDomain	Yes	`originalSecurityDomain`
originalUserName	Yes	`originalUserName`
outcome	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditOutcomeType element type.
failureReason	No	`outcome.failureReason`
majorStatus	No	`outcome.majorStatus`
minorStatus	No	`outcome.minorStatus`
result	Yes	`outcome.result`
reporterComponentId	When different from sourceComponentId	Neither this element, nor its children, should be defined in the shredder configuration file. These elements are generated by the code. This container element uses the children of the auditComponentIdType element type.
sequenceNumber	Yes	Not applicable. This value is an internal number that is not related to `#RECORD_ID`.
sourceComponentId	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditComponentIdType element type.
application	Yes	`CommonBaseEvent/SourceComponentId/@application`
component	Yes	`CommonBaseEvent/SourceComponentId/@component`
componentIdType	Yes	`CommonBaseEvent/SourceComponentId/@componentIdType`
componentType	Yes	`CommonBaseEvent/SourceComponentId/@componentType`
executionEnvironment	No	`CommonBaseEvent/SourceComponentId/@executionEnvironment`
instanceId	No	`CommonBaseEvent/SourceComponentId/@instanceId`
location	Yes	`CommonBaseEvent/SourceComponentId/@location`
locationType	Yes	`CommonBaseEvent/SourceComponentId/@locationType`
processed	No	`CommonBaseEvent/SourceComponentId/@processed`

Table 51. Elements used in AUDIT_AUTHN events (continued)

Element	Always in output	Abbreviated XPath
subComponent	Yes	`CommonBaseEvent/SourceComponentId/@subComponent`
threadId	No	`CommonBaseEvent/SourceComponentId/@threadId`
startTime	No	`startTime [type='dateTime']`
timestamp	Yes	`CommonBaseEvent/@creationTime`

Elements for AUDIT_AUTHN_TERMINATE events

This event type identifies authentication termination events.

The following table lists the elements that can be displayed in the output of an AUDIT_AUTHN_TERMINATE event and their abbreviated XPath statements.

Table 52. Elements used in AUDIT_AUTHN_TERMINATE events

Element	Always in output	Abbreviated XPath
action	No	`action`
auditMsg	No	`auditMsg`
auditMsgElement	No	Neither this element, nor its children, should be defined in the shredder configuration file.
auditTrailId	No	`auditTrailId`
authnType	Yes	`authnType`
authnTypeVersion	No	`authnTypeVersion`
endTime	No	`endTime [type='dateTime']`
extensionName	Yes	`"'AUDIT_AUTHN_TERMINATE'"`
globalInstanceId	Yes	Not applicable. This value is an internal number that is not related to `#GLOBAL_ID`.
loginTime	Yes	`loginTime`
outcome	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditOutcomeType element type.
failureReason	No	`outcome.failureReason`
majorStatus	No	`outcome.majorStatus`
minorStatus	No	`outcome.minorStatus`
result	Yes	`outcome.result`
progName	No	`progName`
registryInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the registryInfoType element type.
serverLocation	Yes	`registryInfo.serverLocation`
serverLocationType	Yes	`registryInfo.serverLocationType`
serverPort	Yes	`registryInfo.serverPort`
type	Yes	`registryInfo.type`
reporterComponentId	When different from the sourceComponentId	Neither this element, nor its children, should be defined in the shredder configuration file. These elements are generated by the code. This container element uses the children of the auditComponentIdType element type.

Table 52. Elements used in AUDIT_AUTHN_TERMINATE events (continued)

Element	Always in output	Abbreviated XPath
sequenceNumber	Yes	Not applicable. This value is an internal number that is not related to #RECORD_ID.
sourceComponentId	Yes	This element is a container element and has no valid XPath. A valid XPath requires a values declaration. This container element uses the children of the auditComponentIdType element type.
application	Yes	CommonBaseEvent/SourceComponentId/@application
component	Yes	CommonBaseEvent/SourceComponentId/@component
componentIdType	Yes	CommonBaseEvent/SourceComponentId/@componentIdType
componentType	Yes	CommonBaseEvent/SourceComponentId/@componentType
executionEnvironment	No	CommonBaseEvent/SourceComponentId/@executionEnvironment
instanceId	No	CommonBaseEvent/SourceComponentId/@instanceId
location	Yes	CommonBaseEvent/SourceComponentId/@location
locationType	Yes	CommonBaseEvent/SourceComponentId/@locationType
processed	No	CommonBaseEvent/SourceComponentId/@processed
subComponent	Yes	CommonBaseEvent/SourceComponentId/@subComponent
threadId	No	CommonBaseEvent/SourceComponentId/@threadId
startTime	No	startTime [type='dateTime']
terminateReason	When action is logout	terminateReason
timestamp	Yes	CommonBaseEvent/@creationTime
userInfo	When action is logout	This element is a container element and has no valid XPath. A valid XPath requires a values declaration. This container element uses the children of the userInfoType element type.
appUserName	Yes	userInfo.appUserName
attributes	No	userInfo.attributes
callerList	No	userInfo.callerList
domain	No	userInfo.domain
location	No	userInfo.location
locationType	No	userInfo.locationType
realm	No	userInfo.realm
registryUserName	Yes	userInfo.registryUserName
sessionId	No	userInfo.sessionId
uniqueId	No	userInfo.uniqueId

Elements for AUDIT_AUTHZ events

This event type identifies authorization events.

The following table lists the elements that can be displayed in the output of an AUDIT_AUTHZ event and their abbreviated XPath statements.

Table 53. Elements used in AUDIT_AUTHZ events

Element	**Always in output**	**Abbreviated XPath**
accessDecision	When `outcome.result` is `SUCCESSFUL`	`accessDecision`
accessDecisionReason	When `accessDecision` is `Denied`	`accessDecisionReason`
action	No	`action`
appName	No	`appName`
attributePermissionInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the attributePermissionInfoType element type.
attributeNames	Yes	`attributePermissionInfo.attributeNames`
checked	Yes	`attributePermissionInfo.checked`
denied	No	`attributePermissionInfo.denied`
granted	No	`attributePermissionInfo.granted`
attributes	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the attributeType element type.
name	Yes	`attributes.name`
source	No	`attributes.source`
value	Yes	`attributes.value`
auditMsg	No	`auditMsg`
auditMsgElement	No	Neither this element, nor its children, should be defined in the shredder configuration file.
auditTrailId	No	`auditTrailId`
endTime	No	`endTime [type='dateTime']`
extensionName	Yes	`"'AUDIT_AUTHZ'"`
globalInstanceId	Yes	Not applicable. This value is an internal number that is not related to `#GLOBAL_ID`.
outcome	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditOutcomeType element type.
failureReason	No	`outcome.failureReason`
majorStatus	No	`outcome.majorStatus`
minorStatus	No	`outcome.minorStatus`
result	Yes	`outcome.result`
permissionInfo	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the permissionInfoType element type.
checked	Yes	`permissionInfo.checked`
denied	No	`permissionInfo.denied`
granted	No	`permissionInfo.granted`
J2EERolesChecked	No	`permissionInfo.J2EERolesChecked`
J2EERolesGranted	No	`permissionInfo.J2EERolesGranted`

Table 53. Elements used in AUDIT_AUTHZ events (continued)

Element	Always in output	Abbreviated XPath
policyInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the policyInfoType element type.
attributes	No	`policyInfo.attributes`
branch	No	`policyInfo.branch`
description	Yes	`policyInfo.description`
name	Yes	`policyInfo.name`
type	Yes	`policyInfo.type`
registryInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the registryInfoType element type.
serverLocation	Yes	`registryInfo.serverLocation`
serverLocationType	Yes	`registryInfo.serverLocationType`
serverPort	Yes	`registryInfo.serverPort`
type	Yes	`registryInfo.type`
reporterComponentId	When different from sourceComponentId	Neither this element, nor its children, should be defined in the shredder configuration file. These elements are generated by the code. This container element uses the children of the auditComponentIdType element type.
resourceInfo	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the resourceInfoType element type.
attributes	No	`resourceInfo.attributes`
nameInApp	Yes	`resourceInfo.nameInApp`
nameInPolicy	Yes	`resourceInfo.nameInPolicy`
type	Yes	`resourceInfo.type`
sequenceNumber	Yes	Not applicable. This value is an internal number that is not related to `#RECORD_ID`.
sourceComponentId	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditComponentIdType element type.
application	Yes	`CommonBaseEvent/SourceComponentId/@application`
component	Yes	`CommonBaseEvent/SourceComponentId/@component`
componentIdType	Yes	`CommonBaseEvent/SourceComponentId/@componentIdType`
componentType	Yes	`CommonBaseEvent/SourceComponentId/@componentType`
executionEnvironment	No	`CommonBaseEvent/SourceComponentId/@executionEnvironment`
instanceId	No	`CommonBaseEvent/SourceComponentId/@instanceId`
location	Yes	`CommonBaseEvent/SourceComponentId/@location`
locationType	Yes	`CommonBaseEvent/SourceComponentId/@locationType`

Table 53. Elements used in AUDIT_AUTHZ events (continued)

Element	Always in output	Abbreviated XPath
processed	No	CommonBaseEvent/SourceComponentId/@processed
subComponent	Yes	CommonBaseEvent/SourceComponentId/@subComponent
threadId	No	CommonBaseEvent/SourceComponentId/@threadId
startTime	No	startTime [type='dateTime']
timestamp	Yes	CommonBaseEvent/@creationTime
userInfo	Yes	This element is a container element and has no valid XPath. A valid XPath requires a values declaration. This container element uses the children of the userInfoType element type.
appUserName	Yes	userInfo.appUserName
attributes	No	userInfo.attributes
callerList	No	userInfo.callerList
domain	No	userInfo.domain
location	No	userInfo.location
locationType	No	userInfo.locationType
realm	No	userInfo.realm
registryUserName	Yes	userInfo.registryUserName
sessionId	No	userInfo.sessionId
uniqueId	No	userInfo.uniqueId

Elements for AUDIT_COMPLIANCE events

This event type records whether a specified security policy was being complied with.

The following table lists the elements that can be displayed in the output of an AUDIT_COMPLIANCE event and their abbreviated XPath statements.

Table 54. Elements used in AUDIT_COMPLIANCE events

Element	Always in output	Abbreviated XPath
auditMsg	No	auditMsg
auditMsgElement	No	Neither this element, nor its children, should be defined in the shredder configuration file.
auditTrailId	No	auditTrailId
complianceStatus	Yes	complianceStatus
endTime	No	endTime [type='dateTime']
extensionName	Yes	"'AUDIT_COMPLIANCE'"
fixDescription	No	fixDescription
fixId	No	fixId
globalInstanceId	Yes	Not applicable. This value is an internal number that is not related to #GLOBAL_ID.
message	No	message
outcome	Yes	This element is a container element and has no valid XPath. A valid XPath requires a values declaration. This container element uses the children of the auditOutcomeType element type.
failureReason	No	outcome.failureReason

Table 54. Elements used in AUDIT_COMPLIANCE events (continued)

Element	Always in output	Abbreviated XPath
majorStatus	No	`outcome.majorStatus`
minorStatus	No	`outcome.minorStatus`
result	Yes	`outcome.result`
policyDescription	No	`policyDescription`
policyName	No	`policyName`
recommendation	No	`recommendation`
reporterComponentId	When different from sourceComponentId	Neither this element, nor its children, should be defined in the shredder configuration file. These elements are generated by the code. This container element uses the children of the auditComponentIdType element type.
sequenceNumber	Yes	Not applicable. This value is an internal number that is not related to `#RECORD_ID`.
severity	No	`severity`
sourceComponentId	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditComponentIdType element type.
application	Yes	`CommonBaseEvent/SourceComponentId/@application`
component	Yes	`CommonBaseEvent/SourceComponentId/@component`
componentIdType	Yes	`CommonBaseEvent/SourceComponentId/@componentIdType`
componentType	Yes	`CommonBaseEvent/SourceComponentId/@componentType`
executionEnvironment	No	`CommonBaseEvent/SourceComponentId/@executionEnvironment`
instanceId	No	`CommonBaseEvent/SourceComponentId/@instanceId`
location	Yes	`CommonBaseEvent/SourceComponentId/@location`
locationType	Yes	`CommonBaseEvent/SourceComponentId/@locationType`
processed	No	`CommonBaseEvent/SourceComponentId/@processed`
subComponent	Yes	`CommonBaseEvent/SourceComponentId/@subComponent`
threadId	No	`CommonBaseEvent/SourceComponentId/@threadId`
suppressed	No	`suppressed`
startTime	No	`startTime [type='dateTime']`
targetAccount	No	`targetAccount`
targetResource	No	`targetResource`
targetUser	No	`targetUser`
timestamp	Yes	`CommonBaseEvent/@creationTime`
violationClassification	No	`violationClassification`
violationDescription	No	`violationDescription`
violationName	When `complianceStatus` is `nonCompliant`	`violationName`

Elements for AUDIT_DATA_SYNC events

The event type provides information on data synchronization events.

The following table lists the elements that can be displayed in the output of an AUDIT_DATA_SYNC event and their abbreviated XPath statements.

Table 55. Elements used in AUDIT_DATA_SYNC events

Element	Always in output	Abbreviated XPath
action	Yes	`action`
auditMsg	No	`auditMsg`
auditMsgElement	No	Neither this element, nor its children, should be defined in the shredder configuration file.
auditTrailId	No	`auditTrailId`
extensionName	No	`endTime [type='dateTime']`
eventType	Yes	`"'AUDIT_DATA_SYNC'"`
globalInstanceId	Yes	Not applicable. This value is an internal number that is not related to `#GLOBAL_ID`.
outcome	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditOutcomeType element type.
failureReason	No	`outcome.failureReason`
majorStatus	No	`outcome.majorStatus`
minorStatus	No	`outcome.minorStatus`
result	Yes	`outcome.result`
registryInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the registryInfoType element type.
serverLocation	Yes	`registryInfo.serverLocation`
serverLocationType	Yes	`registryInfo.serverLocationType`
serverPort	Yes	`registryInfo.serverPort`
type	Yes	`registryInfo.type`
reporterComponentId	When different from sourceComponentId	Neither this element, nor its children, should be defined in the shredder configuration file. These elements are generated by the code. This container element uses the children of the auditComponentIdType element type.
resourceInfo	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the resourceInfoType element type.
attributes	No	`resourceInfo.attributes`
nameInApp	Yes	`resourceInfo.nameInApp`
nameInPolicy	Yes	`resourceInfo.nameInPolicy`
type	Yes	`resourceInfo.type`
sequenceNumber	Yes	Not applicable. This value is an internal number that is not related to `#RECORD_ID`.
sourceComponentId	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditComponentIdType element type.

Table 55. Elements used in AUDIT_DATA_SYNC events (continued)

Element	Always in output	Abbreviated XPath
application	Yes	`CommonBaseEvent/SourceComponentId/@application`
component	Yes	`CommonBaseEvent/SourceComponentId/@component`
componentIdType	Yes	`CommonBaseEvent/SourceComponentId/@componentIdType`
componentType	Yes	`CommonBaseEvent/SourceComponentId/@componentType`
executionEnvironment	No	`CommonBaseEvent/SourceComponentId/@executionEnvironment`
instanceId	No	`CommonBaseEvent/SourceComponentId/@instanceId`
location	Yes	`CommonBaseEvent/SourceComponentId/@location`
locationType	Yes	`CommonBaseEvent/SourceComponentId/@locationType`
processed	No	`CommonBaseEvent/SourceComponentId/@processed`
subComponent	Yes	`CommonBaseEvent/SourceComponentId/@subComponent`
threadId	No	`CommonBaseEvent/SourceComponentId/@threadId`
startTime	No	`startTime [type='dateTime']`
timestamp	Yes	`CommonBaseEvent/@creationTime`
userInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the userInfoType element type.
appUserName	Yes	`userInfo.appUserName`
attributes	No	`userInfo.attributes`
callerList	No	`userInfo.callerList`
domain	No	`userInfo.domain`
location	No	`userInfo.location`
locationType	No	`userInfo.locationType`
realm	No	`userInfo.realm`
registryUserName	Yes	`userInfo.registryUserName`
sessionId	No	`userInfo.sessionId`
uniqueId	No	`userInfo.uniqueId`

Elements for AUDIT_MGMT_CONFIG events

This event type identifies configuration and other management events for a server.

The following table lists the elements that can be displayed in the output of an AUDIT_MGMT_CONFIG event and their abbreviated XPath statements.

Table 56. Elements used in AUDIT_MGMT_CONFIG events

Element	Always in output	Abbreviated XPath
action	Yes	`action`
auditMsg	No	`auditMsg`
auditMsgElement	No	Neither this element, nor its children, should be defined in the shredder configuration file.

Table 56. Elements used in AUDIT_MGMT_CONFIG events (continued)

Element	Always in output	Abbreviated XPath
auditTrailId	No	`auditTrailId`
endTime	No	`endTime [type='dateTime']`
extensionName	Yes	`"'AUDIT_MGMT_CONFIG'"`
globalInstanceId	Yes	Not applicable. This value is an internal number that is not related to `#GLOBAL_ID`.
mgmtInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the mgmtInfoType element type.
command	No	`mgmtInfo.command`
targetInfo	No	`mgmtInfo.targetInfo`
outcome	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditOutcomeType element type.
failureReason	No	`outcome.failureReason`
majorStatus	No	`outcome.majorStatus`
minorStatus	No	`outcome.minorStatus`
result	Yes	`outcome.result`
registryInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the registryInfoType element type.
serverLocation	Yes	`registryInfo.serverLocation`
serverLocationType	Yes	`registryInfo.serverLocationType`
serverPort	Yes	`registryInfo.serverPort`
type	Yes	`registryInfo.type`
reporterComponentId	When different from sourceComponentId	Neither this element, nor its children, should be defined in the shredder configuration file. These elements are generated by the code. This container element uses the children of the auditComponentIdType element type.
sequenceNumber	Yes	Not applicable. This value is an internal number that is not related to `#RECORD_ID`.
sourceComponentId	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditComponentIdType element type.
application	Yes	`CommonBaseEvent/SourceComponentId/@application`
component	Yes	`CommonBaseEvent/SourceComponentId/@component`
componentIdType	Yes	`CommonBaseEvent/SourceComponentId/@componentIdType`
componentType	Yes	`CommonBaseEvent/SourceComponentId/@componentType`
executionEnvironment	No	`CommonBaseEvent/SourceComponentId/@executionEnvironment`
instanceId	No	`CommonBaseEvent/SourceComponentId/@instanceId`
location	Yes	`CommonBaseEvent/SourceComponentId/@location`

Table 56. Elements used in AUDIT_MGMT_CONFIG events (continued)

Element	Always in output	Abbreviated XPath
locationType	Yes	`CommonBaseEvent/SourceComponentId/@locationType`
processed	No	`CommonBaseEvent/SourceComponentId/@processed`
subComponent	Yes	`CommonBaseEvent/SourceComponentId/@subComponent`
threadId	No	`CommonBaseEvent/SourceComponentId/@threadId`
startTime	No	`startTime [type='dateTime']`
timestamp	Yes	`CommonBaseEvent/@creationTime`
type	Yes	`type`
userInfo	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the userInfoType element type.
appUserName	Yes	`userInfo.appUserName`
attributes	No	`userInfo.attributes`
callerList	No	`userInfo.callerList`
domain	No	`userInfo.domain`
location	No	`userInfo.location`
locationType	No	`userInfo.locationType`
realm	No	`userInfo.realm`
registryUserName	Yes	`userInfo.registryUserName`
sessionId	No	`userInfo.sessionId`
uniqueId	No	`userInfo.uniqueId`

Elements for AUDIT_MGMT_POLICY events

This event type identifies the security policy management events, such as creation of access control lists.

The following table lists the elements that can be displayed in the output of an AUDIT_MGMT_POLICY event and their abbreviated XPath statements.

Table 57. Elements used in AUDIT_MGMT_POLICY events

Element	Always in output	Abbreviated XPath
action	Yes	`action`
auditMsg	No	`auditMsg`
auditMsgElement	No	Neither this element, nor its children, should be defined in the shredder configuration file.
auditTrailId	No	`auditTrailId`
endTime	No	`endTime [type='dateTime']`
extensionName	Yes	`"'AUDIT_MGMT_POLICY'"`
globalInstanceId	Yes	Not applicable. This value is an internal number that is not related to `#GLOBAL_ID`.
memberships	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the membershipInfoType element type.
id	No	`memberships.id`

Table 57. Elements used in AUDIT_MGMT_POLICY events (continued)

Element	Always in output	Abbreviated XPath
name	No	memberships.name
type	Yes	memberships.type
mgmtInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a values declaration. This container element uses the children of the mgmtInfoType element type.
command	No	mgmtInfo.command
targetInfo	No	mgmtInfo.targetInfo
outcome	Yes	This element is a container element and has no valid XPath. A valid XPath requires a values declaration. This container element uses the children of the auditOutcomeType element type.
failureReason	No	outcome.failureReason
majorStatus	No	outcome.majorStatus
minorStatus	No	outcome.minorStatus
result	Yes	outcome.result
policyInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a values declaration. This container element uses the children of the policyInfoType element type.
attributes	No	policyInfo.attributes
branch	No	policyInfo.branch
description	Yes	policyInfo.description
name	Yes	policyInfo.name
type	Yes	policyInfo.type
registryInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a values declaration. This container element uses the children of the registryInfoType element type.
serverLocation	Yes	registryInfo.serverLocation
serverLocationType	Yes	registryInfo.serverLocationType
serverPort	Yes	registryInfo.serverPort
type	Yes	registryInfo.type
reporterComponentId	When different from sourceComponentId	Neither this element, nor its children, should be defined in the shredder configuration file. These elements are generated by the code. This container element uses the children of the auditComponentIdType element type.
resourceInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a values declaration. This container element uses the children of the resourceInfoType element type.
attributes	No	resourceInfo.attributes
nameInApp	Yes	resourceInfo.nameInApp
nameInPolicy	Yes	resourceInfo.nameInPolicy
type	Yes	resourceInfo.type
sequenceNumber	Yes	Not applicable. This value is an internal number that is not related to #RECORD_ID.

Table 57. Elements used in AUDIT_MGMT_POLICY events (continued)

Element	Always in output	Abbreviated XPath
sourceComponentId	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditComponentIdType element type.
application	Yes	`CommonBaseEvent/SourceComponentId/@application`
component	Yes	`CommonBaseEvent/SourceComponentId/@component`
componentIdType	Yes	`CommonBaseEvent/SourceComponentId/@componentIdType`
componentType	Yes	`CommonBaseEvent/SourceComponentId/@componentType`
executionEnvironment	No	`CommonBaseEvent/SourceComponentId/@executionEnvironment`
instanceId	No	`CommonBaseEvent/SourceComponentId/@instanceId`
location	Yes	`CommonBaseEvent/SourceComponentId/@location`
locationType	Yes	`CommonBaseEvent/SourceComponentId/@locationType`
processed	No	`CommonBaseEvent/SourceComponentId/@processed`
subComponent	Yes	`CommonBaseEvent/SourceComponentId/@subComponent`
threadId	No	`CommonBaseEvent/SourceComponentId/@threadId`
startTime	No	`startTime [type='dateTime']`
timestamp	Yes	`CommonBaseEvent/@creationTime`
userInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the userInfoType element type.
appUserName	Yes	`userInfo.appUserName`
attributes	No	`userInfo.attributes`
callerList	No	`userInfo.callerList`
domain	No	`userInfo.domain`
location	No	`userInfo.location`
locationType	No	`userInfo.locationType`
realm	No	`userInfo.realm`
registryUserName	Yes	`userInfo.registryUserName`
sessionId	No	`userInfo.sessionId`
uniqueId	No	`userInfo.uniqueId`

Elements for AUDIT_MGMT_PROVISIONING events

This event type identifies provisioning events, such as creating an account for a user on a specific machine.

The following table lists the elements that can be displayed in the output of an AUDIT_MGMT_PROVISIONING event and their abbreviated XPath statements.

Table 58. Elements used in AUDIT_MGMT_PROVISIONING events

Element	Always in output	Abbreviated XPath
action	Yes	`action`
auditMsg	No	`auditMsg`
auditMsgElement	No	Neither this element, nor its children, should be defined in the shredder configuration file.
auditTrailId	No	`auditTrailId`
endTime	No	`endTime [type='dateTime']`
extensionName	Yes	`"'AUDIT_MGMT_PROVISIONING'"`
globalInstanceId	Yes	Not applicable. This value is an internal number that is not related to `#GLOBAL_ID`.
outcome	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditOutcomeType element type.
failureReason	No	`outcome.failureReason`
majorStatus	No	`outcome.majorStatus`
minorStatus	No	`outcome.minorStatus`
result	Yes	`outcome.result`
provisioningInfo	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the provisioningInfoType element type.
accountId	No	`provisioningInfo.accountId`
resourceId	Yes	`provisioningInfo.resourceId`
resourceType	Yes	`provisioningInfo.resourceType`
registryInfo	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the registryInfoType element type.
serverLocation	Yes	`registryInfo.serverLocation`
serverLocationType	Yes	`registryInfo.serverLocationType`
serverPort	Yes	`registryInfo.serverPort`
type	Yes	`registryInfo.type`
reporterComponentId	When different from sourceComponentId	Neither this element, nor its children, should be defined in the shredder configuration file. These elements are generated by the code. This container element uses the children of the auditComponentIdType element type.
sequenceNumber	Yes	Not applicable. This value is an internal number that is not related to `#RECORD_ID`.
sourceComponentId	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditComponentIdType element type.
application	Yes	`CommonBaseEvent/SourceComponentId/@application`
component	Yes	`CommonBaseEvent/SourceComponentId/@component`
componentIdType	Yes	`CommonBaseEvent/SourceComponentId/@componentIdType`
componentType	Yes	`CommonBaseEvent/SourceComponentId/@componentType`

Table 58. Elements used in AUDIT_MGMT_PROVISIONING events (continued)

Element	Always in output	Abbreviated XPath
executionEnvironment	No	CommonBaseEvent/SourceComponentId/@executionEnvironment
instanceId	No	CommonBaseEvent/SourceComponentId/@instanceId
location	Yes	CommonBaseEvent/SourceComponentId/@location
locationType	Yes	CommonBaseEvent/SourceComponentId/@locationType
processed	No	CommonBaseEvent/SourceComponentId/@processed
subComponent	Yes	CommonBaseEvent/SourceComponentId/@subComponent
threadId	No	CommonBaseEvent/SourceComponentId/@threadId
startTime	No	startTime [type='dateTime']
targetUserInfo	Yes	This element is a container element and has no valid XPath. A valid XPath requires a values declaration. This container element uses the children of the registryObjectInfoType element type.
attributes	No	registryObjectInfo.attributes
description	No	registryObjectInfo.description
name	Yes	registryObjectInfo.name
registryName	No	registryObjectInfo.registryName
type	Yes	registryObjectInfo.type
targetUserRegistryInfo	Yes	This element is a container element and has no valid XPath. A valid XPath requires a values declaration. This container element uses the children of the targetUserRegistryInfoType element type.
serverLocation	Yes	registryInfo.serverLocation
serverLocationType	Yes	registryInfo.serverLocationType
serverPort	Yes	registryInfo.serverPort
type	Yes	registryInfo.type
timestamp	Yes	CommonBaseEvent/@creationTime
userInfo	Yes	This element is a container element and has no valid XPath. A valid XPath requires a values declaration. This container element uses the children of the userInfoType element type.
appUserName	Yes	userInfo.appUserName
attributes	No	userInfo.attributes
callerList	No	userInfo.callerList
domain	No	userInfo.domain
location	No	userInfo.location
locationType	No	userInfo.locationType
realm	No	userInfo.realm
registryUserName	Yes	userInfo.registryUserName
sessionId	No	userInfo.sessionId
uniqueId	No	userInfo.uniqueId

Elements for AUDIT_MGMT_REGISTRY events

This event type identifies registry management events, such as creating users and groups, changing passwords by the administrator, and changing the properties for users and groups.

The following table lists the elements that can be displayed in the output of an AUDIT_MGMT_REGISTRY event and their abbreviated XPath statements.

Table 59. Elements used in AUDIT_MGMT_REGISTRY events

Element	Always in output	Abbreviated XPath
action	Yes	`action`
auditMsg	No	`auditMsg`
auditMsgElement	No	Neither this element, nor its children, should be defined in the shredder configuration file.
auditTrailId	No	`auditTrailId`
endTime	No	`endTime [type='dateTime']`
extensionName	Yes	`"'AUDIT_MGMT_REGISTRY'"`
globalInstanceId	Yes	Not applicable. This value is an internal number that is not related to `#GLOBAL_ID`.
mgmtInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the mgmtInfoType element type.
command	No	`mgmtInfo.command`
targetInfo	No	`mgmtInfo.targetInfo`
outcome	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditOutcomeType element type.
failureReason	No	`outcome.failureReason`
majorStatus	No	`outcome.majorStatus`
minorStatus	No	`outcome.minorStatus`
result	Yes	`outcome.result`
registryInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the registryInfoType element type.
serverLocation	Yes	`registryInfo.serverLocation`
serverLocationType	Yes	`registryInfo.serverLocationType`
serverPort	Yes	`registryInfo.serverPort`
type	Yes	`registryInfo.type`
registryObjectInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the registryObjectInfoType element type.
attributes	No	`registryObjectInfo.attributes`
description	No	`registryObjectInfo.description`
name	Yes	`registryObjectInfo.name`
registryName	No	`registryObjectInfo.registryName`
type	Yes	`registryObjectInfo.type`

Table 59. Elements used in AUDIT_MGMT_REGISTRY events (continued)

Element	Always in output	Abbreviated XPath
reporterComponentId	When different from sourceComponentId	Neither this element, nor its children, should be defined in the shredder configuration file. These elements are generated by the code. This container element uses the children of the auditComponentIdType element type.
sequenceNumber	Yes	Not applicable. This value is an internal number that is not related to `#RECORD_ID`.
sourceComponentId	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditComponentIdType element type.
application	Yes	`CommonBaseEvent/SourceComponentId/@application`
component	Yes	`CommonBaseEvent/SourceComponentId/@component`
componentIdType	Yes	`CommonBaseEvent/SourceComponentId/@componentIdType`
componentType	Yes	`CommonBaseEvent/SourceComponentId/@componentType`
executionEnvironment	No	`CommonBaseEvent/SourceComponentId/@executionEnvironment`
instanceId	No	`CommonBaseEvent/SourceComponentId/@instanceId`
location	Yes	`CommonBaseEvent/SourceComponentId/@location`
locationType	Yes	`CommonBaseEvent/SourceComponentId/@locationType`
processed	No	`CommonBaseEvent/SourceComponentId/@processed`
subComponent	Yes	`CommonBaseEvent/SourceComponentId/@subComponent`
threadId	No	`CommonBaseEvent/SourceComponentId/@threadId`
startTime	No	`startTime [type='dateTime']`
timestamp	Yes	`CommonBaseEvent/@creationTime`
userInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the userInfoType element type.
appUserName	Yes	`userInfo.appUserName`
attributes	No	`userInfo.attributes`
callerList	No	`userInfo.callerList`
domain	No	`userInfo.domain`
location	No	`userInfo.location`
locationType	No	`userInfo.locationType`
realm	No	`userInfo.realm`
registryUserName	Yes	`userInfo.registryUserName`
sessionId	No	`userInfo.sessionId`
uniqueId	No	`userInfo.uniqueId`

Elements for AUDIT_MGMT_RESOURCE events

This event type identifies resource management events.

The following table lists the elements that can be displayed in the output of an AUDIT_MGMT_RESOURCE event and their abbreviated XPath statements.

Table 60. Elements used in AUDIT_MGMT_RESOURCE events

Element	Always in output	Abbreviated XPath
Action	Yes	`action`
auditMsg	No	`auditMsg`
auditMsgElement	No	Neither this element, nor its children, should be defined in the shredder configuration file.
auditTrailId	No	`auditTrailId`
endTime	No	`endTime [type='dateTime']`
extensionName	Yes	`"'AUDIT_MGMT_RESOURCE'"`
globalInstanceId	Yes	Not applicable. This value is an internal number that is not related to `#GLOBAL_ID`.
mgmtInfo	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the mgmtInfoType element type.
command	No	`mgmtInfo.command`
targetInfo	No	`mgmtInfo.targetInfo`
outcome	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditOutcomeType element type.
failureReason	No	`outcome.failureReason`
majorStatus	No	`outcome.majorStatus`
minorStatus	No	`outcome.minorStatus`
result	Yes	`outcome.result`
registryInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the registryInfoType element type.
serverLocation	Yes	`registryInfo.serverLocation`
serverLocationType	Yes	`registryInfo.serverLocationType`
serverPort	Yes	`registryInfo.serverPort`
type	Yes	`registryInfo.type`
reporterComponentId	When different from sourceComponentId	Neither this element, nor its children, should be defined in the shredder configuration file. These elements are generated by the code. This container element uses the children of the auditComponentIdType element type.
registryInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the registryObjectInfoType element type.
attributes	No	`registryObjectInfo.attributes`
description	No	`registryObjectInfo.description`
name	Yes	`registryObjectInfo.name`
registryName	No	`registryObjectInfo.registryName`
type	Yes	`registryObjectInfo.type`
resourceInfo	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the resourceInfoType element type.

Table 60. Elements used in AUDIT_MGMT_RESOURCE events (continued)

Element	Always in output	Abbreviated XPath
attributes	No	`resourceInfo.attributes`
nameInApp	Yes	`resourceInfo.nameInApp`
nameInPolicy	Yes	`resourceInfo.nameInPolicy`
type	Yes	`resourceInfo.type`
sequenceNumber	Yes	Not applicable. This value is an internal number that is not related to `#RECORD_ID`.
sourceComponentId	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditComponentIdType element type.
application	Yes	`CommonBaseEvent/SourceComponentId/@application`
component	Yes	`CommonBaseEvent/SourceComponentId/@component`
componentIdType	Yes	`CommonBaseEvent/SourceComponentId/@componentIdType`
componentType	Yes	`CommonBaseEvent/SourceComponentId/@componentType`
executionEnvironment	No	`CommonBaseEvent/SourceComponentId/@executionEnvironment`
instanceId	No	`CommonBaseEvent/SourceComponentId/@instanceId`
location	Yes	`CommonBaseEvent/SourceComponentId/@location`
locationType	Yes	`CommonBaseEvent/SourceComponentId/@locationType`
processed	No	`CommonBaseEvent/SourceComponentId/@processed`
subComponent	Yes	`CommonBaseEvent/SourceComponentId/@subComponent`
threadId	No	`CommonBaseEvent/SourceComponentId/@threadId`
startTime	No	`startTime [type='dateTime']`
timestamp	Yes	`CommonBaseEvent/@creationTime`
userInfo	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the userInfoType element type.
appUserName	Yes	`userInfo.appUserName`
attributes	No	`userInfo.attributes`
callerList	No	`userInfo.callerList`
domain	No	`userInfo.domain`
location	No	`userInfo.location`
locationType	No	`userInfo.locationType`
realm	No	`userInfo.realm`
registryUserName	Yes	`userInfo.registryUserName`
sessionId	No	`userInfo.sessionId`
uniqueId	No	`userInfo.uniqueId`

Elements for AUDIT_PASSWORD_CHANGE events

This event type identifies password changes initiated by the end user.

The following table lists the elements that can be displayed in the output of an AUDIT_PASSWORD_CHANGE event and their abbreviated XPath statements.

Table 61. Elements used in AUDIT_PASSWORD_CHANGE events

Element	Always in output	Abbreviated XPath
auditMsg	No	`auditMsg`
auditMsgElement	No	Neither this element, nor its children, should be defined in the shredder configuration file.
auditTrailId	No	`auditTrailId`
endTime	No	`endTime [type='dateTime']`
extensionName	Yes	`"'AUDIT_PASSWORD_CHANGE'"`
globalInstanceId	Yes	Not applicable. This value is an internal number that is not related to `#GLOBAL_ID`.
outcome	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditOutcomeType element type.
failureReason	No	`outcome.failureReason`
majorStatus	No	`outcome.majorStatus`
minorStatus	No	`outcome.minorStatus`
result	Yes	`outcome.result`
provisioningInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the provisioningInfoType element type.
accountId	No	`provisioningInfo.accountId`
resourceId	Yes	`provisioningInfo.resourceId`
resourceType	Yes	`provisioningInfo.resourceType`
registryInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the registryInfoType element type.
serverLocation	Yes	`registryInfo.serverLocation`
serverLocationType	Yes	`registryInfo.serverLocationType`
serverPort	Yes	`registryInfo.serverPort`
type	Yes	`registryInfo.type`
reporterComponentId	When different from sourceComponentId	Neither this element, nor its children, should be defined in the shredder configuration file. These elements are generated by the code. This container element uses the children of the auditComponentIdType element type.
sequenceNumber	Yes	Not applicable. This value is an internal number that is not related to `#RECORD_ID`.
sourceComponentId	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditComponentIdType element type.
application	Yes	`CommonBaseEvent/SourceComponentId/ @application`

Table 61. Elements used in AUDIT_PASSWORD_CHANGE events (continued)

Element	Always in output	Abbreviated XPath
component	Yes	`CommonBaseEvent/SourceComponentId/@component`
componentIdType	Yes	`CommonBaseEvent/SourceComponentId/@componentIdType`
componentType	Yes	`CommonBaseEvent/SourceComponentId/@componentType`
executionEnvironment	No	`CommonBaseEvent/SourceComponentId/@executionEnvironment`
instanceId	No	`CommonBaseEvent/SourceComponentId/@instanceId`
location	Yes	`CommonBaseEvent/SourceComponentId/@location`
locationType	Yes	`CommonBaseEvent/SourceComponentId/@locationType`
processed	No	`CommonBaseEvent/SourceComponentId/@processed`
subComponent	Yes	`CommonBaseEvent/SourceComponentId/@subComponent`
threadId	No	`CommonBaseEvent/SourceComponentId/@threadId`
startTime	No	`startTime [type='dateTime']`
timestamp	Yes	`CommonBaseEvent/@creationTime`
userInfo	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the userInfoType element type.
appUserName	Yes	`userInfo.appUserName`
attributes	No	`userInfo.attributes`
callerList	No	`userInfo.callerList`
domain	No	`userInfo.domain`
location	No	`userInfo.location`
locationType	No	`userInfo.locationType`
realm	No	`userInfo.realm`
registryUserName	Yes	`userInfo.registryUserName`
sessionId	No	`userInfo.sessionId`
uniqueId	No	`userInfo.uniqueId`

Elements for AUDIT_RESOURCE_ACCESS events

This event type identifies all accesses to a resource, such as a file or HTTP request or response events outside of the AUDIT_AUTHZ events.

The following table lists the elements that can be displayed in the output of an AUDIT_RESOURCE_ACCESS event and their abbreviated XPath statements.

Table 62. Elements used in AUDIT_RESOURCE_ACCESS events

Element	Always in output	Abbreviated XPath
accessDecision	No	`accessDecision`
accessDecisionReason	When `accessDecision` is `Denied`	`accessDecisionReason`
action	Yes	`action`
appName	No	`appName`

Table 62. Elements used in AUDIT_RESOURCE_ACCESS events (continued)

Element	Always in output	Abbreviated XPath
attributePermissionInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the attributePermissionInfoType element type.
attributeNames	Yes	`attributePermissionInfo.attributeNames`
checked	Yes	`attributePermissionInfo.checked`
denied	No	`attributePermissionInfo.denied`
granted	No	`attributePermissionInfo.granted`
auditMsg	No	`auditMsg`
auditMsgElement	No	Neither this element, nor its children, should be defined in the shredder configuration file.
auditTrailId	No	`auditTrailId`
endTime	No	`endTime [type='dateTime']`
extensionName	Yes	`"'AUDIT_RESOURCE_ACCESS'"`
globalInstanceId	Yes	Not applicable. This value is an internal number that is not related to `#GLOBAL_ID`.
httpURLInfo	When `action` is `HTTPRequest`	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the HTTPURLInfoType element type.
method	No	`HTTPURLInfo.method`
requestHeaders		`HTTPURLInfo.requestHeaders`
responseCode		`HTTPURLInfo.responseCode`
responseHeaders		`HTTPURLInfo.responseHeaders`
url		`HTTPURLInfo.url`
outcome	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditOutcomeType element type.
failureReason	No	`outcome.failureReason`
majorStatus	No	`outcome.majorStatus`
minorStatus	No	`outcome.minorStatus`
result	Yes	`outcome.result`
permissionInfo	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the permissionInfoType element type.
checked	Yes	`permissionInfo.checked`
denied	No	`permissionInfo.denied`
granted	No	`permissionInfo.granted`
J2EERolesChecked	No	`permissionInfo.J2EERolesChecked`
J2EERolesGranted	No	`permissionInfo.J2EERolesGranted`
registryInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the registryInfoType element type.
serverLocation	Yes	`registryInfo.serverLocation`
serverLocationType	Yes	`registryInfo.serverLocationType`

Table 62. Elements used in AUDIT_RESOURCE_ACCESS events (continued)

Element	Always in output	Abbreviated XPath
serverPort	Yes	registryInfo.serverPort
type	Yes	registryInfo.type
reporterComponentId	When different from the sourceComponentId	Neither this element, nor its children, should be defined in the shredder configuration file. These elements are generated by the code. This container element uses the children of the auditComponentIdType element type.
resourceInfo	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the resourceInfoType element type.
attributes	No	resourceInfo.attributes
nameInApp	Yes	resourceInfo.nameInApp
nameInPolicy	Yes	resourceInfo.nameInPolicy
type	Yes	resourceInfo.type
sequenceNumber	Yes	Not applicable. This value is an internal number that is not related to `#RECORD_ID`.
sourceComponentId	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditComponentIdType element type.
application	Yes	CommonBaseEvent/SourceComponentId/@application
component	Yes	CommonBaseEvent/SourceComponentId/@component
componentIdType	Yes	CommonBaseEvent/SourceComponentId/@componentIdType
componentType	Yes	CommonBaseEvent/SourceComponentId/@componentType
executionEnvironment	No	CommonBaseEvent/SourceComponentId/@executionEnvironment
instanceId	No	CommonBaseEvent/SourceComponentId/@instanceId
location	Yes	CommonBaseEvent/SourceComponentId/@location
locationType	Yes	CommonBaseEvent/SourceComponentId/@locationType
processed	No	CommonBaseEvent/SourceComponentId/@processed
subComponent	Yes	CommonBaseEvent/SourceComponentId/@subComponent
threadId	No	CommonBaseEvent/SourceComponentId/@threadId
startTime	No	startTime [type='dateTime']
timestamp	Yes	CommonBaseEvent/@creationTime
userInfo	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the userInfoType element type.
appUserName	Yes	userInfo.appUserName
attributes	No	userInfo.attributes
callerList	No	userInfo.callerList
domain	No	userInfo.domain

Table 62. Elements used in AUDIT_RESOURCE_ACCESS events (continued)

Element	Always in output	Abbreviated XPath
location	No	userInfo.location
locationType	No	userInfo.locationType
realm	No	userInfo.realm
registryUserName	Yes	userInfo.registryUserName
sessionId	No	userInfo.sessionId
uniqueId	No	userInfo.uniqueId

Elements for AUDIT_RUNTIME events

This event type identifies runtime events, such as starting, stopping, and capacity planning-related events for security servers. This event type is not meant for administrative operations performed by a system administrator. Such operations need to use the AUDIT_MGMT_* event types.

The following table lists the elements that can be displayed in the output of an AUDIT_RUNTIME event and their abbreviated XPath statements.

Table 63. Elements used in AUDIT_RUNTIME events

Element	Always in output	Abbreviated XPath
action	Yes	action
auditMsg	No	auditMsg
auditMsgElement	No	Neither this element, nor its children, should be defined in the shredder configuration file.
auditTrailId	No	auditTrailId
endTime	No	endTime [type='dateTime']
extensionName	Yes	"'AUDIT_RUNTIME'"
globalInstanceId	Yes	Not applicable. This value is an internal number that is not related to #GLOBAL_ID.
outcome	Yes	This element is a container element and has no valid XPath. A valid XPath requires a values declaration. This container element uses the children of the auditOutcomeType element type.
failureReason	No	outcome.failureReason
majorStatus	No	outcome.majorStatus
minorStatus	No	outcome.minorStatus
result	Yes	outcome.result
perfInfo	When action is statistic	This element is a container element and has no valid XPath. A valid XPath requires a values declaration. This container element uses the children of the perfInfoType element type.
aggregate	Yes	perfInfo.aggregate
description	Yes	perfInfo.description
name	Yes	perfInfo.name
maxValue	No	perfInfo.maxValue
minValue	No	perfInfo.minValue
numDataPoints	Yes	perfInfo.numDataPoints
unit	Yes	perfInfo.unit
value	Yes	perfInfo.value

Table 63. Elements used in AUDIT_RUNTIME events (continued)

Element	Always in output	Abbreviated XPath
registryInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the registryInfoType element type.
serverLocation	Yes	`registryInfo.serverLocation`
serverLocationType	Yes	`registryInfo.serverLocationType`
serverPort	Yes	`registryInfo.serverPort`
type	Yes	`registryInfo.type`
reporterComponentId	When different from the sourceComponentId	Neither this element, nor its children, should be defined in the shredder configuration file. These elements are generated by the code. This container element uses the children of the auditComponentIdType element type.
resourceInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the resourceInfoType element type.
attributes	No	`resourceInfo.attributes`
nameInApp	Yes	`resourceInfo.nameInApp`
nameInPolicy	Yes	`resourceInfo.nameInPolicy`
type	Yes	`resourceInfo.type`
sequenceNumber	Yes	Not applicable. This value is an internal number that is not related to `#RECORD_ID`.
sourceComponentId	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditComponentIdType element type.
application	Yes	`CommonBaseEvent/SourceComponentId/@application`
component	Yes	`CommonBaseEvent/SourceComponentId/@component`
componentIdType	Yes	`CommonBaseEvent/SourceComponentId/@componentIdType`
componentType	Yes	`CommonBaseEvent/SourceComponentId/@componentType`
executionEnvironment	No	`CommonBaseEvent/SourceComponentId/@executionEnvironment`
instanceId	No	`CommonBaseEvent/SourceComponentId/@instanceId`
location	Yes	`CommonBaseEvent/SourceComponentId/@location`
locationType	Yes	`CommonBaseEvent/SourceComponentId/@locationType`
processed	No	`CommonBaseEvent/SourceComponentId/@processed`
subComponent	Yes	`CommonBaseEvent/SourceComponentId/@subComponent`
threadId	No	`CommonBaseEvent/SourceComponentId/@threadId`
startTime	No	`startTime [type='dateTime']`
timestamp	Yes	`CommonBaseEvent/@creationTime`

Table 63. Elements used in AUDIT_RUNTIME events (continued)

Element	Always in output	Abbreviated XPath
userInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the userInfoType element type.
appUserName	Yes	`userInfo.appUserName`
attributes	No	`userInfo.attributes`
callerList	No	`userInfo.callerList`
domain	No	`userInfo.domain`
location	No	`userInfo.location`
locationType	No	`userInfo.locationType`
realm	No	`userInfo.realm`
registryUserName	Yes	`userInfo.registryUserName`
sessionId	No	`userInfo.sessionId`
uniqueId	No	`userInfo.uniqueId`

Elements for AUDIT_RUNTIME_KEY events

This event type identifies certificate expiration and expiration check events that occur during runtime.

The following table lists the elements that can be displayed in the output of an AUDIT_RUNTIME_KEY event and their abbreviated XPath statements.

Table 64. Elements used in AUDIT_RUNTIME_KEY events

Element	Always in output	Abbreviated XPath
action	Yes	`action`
auditMsg	No	`auditMsg`
auditMsgElement	No	Neither this element, nor its children, should be defined in the shredder configuration file.
auditTrailId	No	`auditTrailId`
endTime	No	`endTime [type='dateTime']`
extensionName	Yes	`"'AUDIT_RUNTIME_KEY'"`
globalInstanceId	Yes	Not applicable. This value is an internal number that is not related to `#GLOBAL_ID`.
keyLabel	Yes	`keyLabel`
lifetime	No	`lifetime`
location	Yes	`location`
locationType	Yes	`locationType`
outcome	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditOutcomeType element type.
failureReason	No	`outcome.failureReason`
majorStatus	No	`outcome.majorStatus`
minorStatus	No	`outcome.minorStatus`
result	Yes	`outcome.result`

Table 64. Elements used in AUDIT_RUNTIME_KEY events (continued)

Element	Always in output	Abbreviated XPath
registryInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the registryInfoType element type.
serverLocation	Yes	`registryInfo.serverLocation`
serverLocationType	Yes	`registryInfo.serverLocationType`
serverPort	Yes	`registryInfo.serverPort`
type	Yes	`registryInfo.type`
reporterComponentId	When different from the sourceComponentId	Neither this element, nor its children, should be defined in the shredder configuration file. These elements are generated by the code. This container element uses the children of the auditComponentIdType element type.
sequenceNumber	Yes	Not applicable. This value is an internal number that is not related to `#RECORD_ID`.
sourceComponentId	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditComponentIdType element type.
application	Yes	`CommonBaseEvent/SourceComponentId/@application`
component	Yes	`CommonBaseEvent/SourceComponentId/@component`
componentIdType	Yes	`CommonBaseEvent/SourceComponentId/@componentIdType`
componentType	Yes	`CommonBaseEvent/SourceComponentId/@componentType`
executionEnvironment	No	`CommonBaseEvent/SourceComponentId/@executionEnvironment`
instanceId	No	`CommonBaseEvent/SourceComponentId/@instanceId`
location	Yes	`CommonBaseEvent/SourceComponentId/@location`
locationType	Yes	`CommonBaseEvent/SourceComponentId/@locationType`
processed	No	`CommonBaseEvent/SourceComponentId/@processed`
subComponent	Yes	`CommonBaseEvent/SourceComponentId/@subComponent`
threadId	No	`CommonBaseEvent/SourceComponentId/@threadId`
startTime	No	`startTime [type='dateTime']`
timestamp	Yes	`CommonBaseEvent/@creationTime`
userInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the userInfoType element type.
appUserName	Yes	`userInfo.appUserName`
attributes	No	`userInfo.attributes`
callerList	No	`userInfo.callerList`
domain	No	`userInfo.domain`
location	No	`userInfo.location`
locationType	No	`userInfo.locationType`

Table 64. Elements used in AUDIT_RUNTIME_KEY events (continued)

Element	Always in output	Abbreviated XPath
realm	No	`userInfo.realm`
registryUserName	Yes	`userInfo.registryUserName`
sessionId	No	`userInfo.sessionId`
uniqueId	No	`userInfo.uniqueId`

Elements for AUDIT_WORKFLOW events

This event type identifies workflow events.

The following table lists the elements that can be displayed in the output of an AUDIT_WORKFLOW event and their abbreviated XPath statements.

Table 65. Elements used in AUDIT_WORKFLOW events

Element	Always in output	Abbreviated XPath
action	Yes	`action`
auditMsg	No	`auditMsg`
auditMsgElement	No	Neither this element, nor its children, should be defined in the shredder configuration file.
auditTrailId	No	`auditTrailId`
endTime	No	`endTime [type='dateTime']`
extensionName	Yes	`"'AUDIT_WORKFLOW'"`
globalInstanceId	Yes	Not applicable. This value is an internal number that is not related to `#GLOBAL_ID`.
outcome	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditOutcomeType element type.
failureReason	No	`outcome.failureReason`
majorStatus	No	`outcome.majorStatus`
minorStatus	No	`outcome.minorStatus`
result	Yes	`outcome.result`
registryInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the registryInfoType element type.
serverLocation	Yes	`registryInfo.serverLocation`
serverLocationType	Yes	`registryInfo.serverLocationType`
serverPort	Yes	`registryInfo.serverPort`
type	Yes	`registryInfo.type`
reporterComponentId	When different from sourceComponentId	Neither this element, nor its children, should be defined in the shredder configuration file. These elements are generated by the code. This container element uses the children of the auditComponentIdType element type.
sequenceNumber	Yes	Not applicable. This value is an internal number that is not related to `#RECORD_ID`.

Table 65. Elements used in AUDIT_WORKFLOW events (continued)

Element	Always in output	Abbreviated XPath
sourceComponentId	Yes	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the auditComponentIdType element type.
application	Yes	`CommonBaseEvent/SourceComponentId/@application`
component	Yes	`CommonBaseEvent/SourceComponentId/@component`
componentIdType	Yes	`CommonBaseEvent/SourceComponentId/@componentIdType`
componentType	Yes	`CommonBaseEvent/SourceComponentId/@componentType`
executionEnvironment	No	`CommonBaseEvent/SourceComponentId/@executionEnvironment`
instanceId	No	`CommonBaseEvent/SourceComponentId/@instanceId`
location	Yes	`CommonBaseEvent/SourceComponentId/@location`
locationType	Yes	`CommonBaseEvent/SourceComponentId/@locationType`
processed	No	`CommonBaseEvent/SourceComponentId/@processed`
subComponent	Yes	`CommonBaseEvent/SourceComponentId/@subComponent`
threadId	No	`CommonBaseEvent/SourceComponentId/@threadId`
startTime	No	`startTime [type='dateTime']`
targetUserInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the userInfoType element type.
appUserName	Yes	`userInfo.appUserName`
attributes	No	`userInfo.attributes`
callerList	No	`userInfo.callerList`
domain	No	`userInfo.domain`
location	No	`userInfo.location`
locationType	No	`userInfo.locationType`
realm	No	`userInfo.realm`
registryUserName	Yes	`userInfo.registryUserName`
sessionId	No	`userInfo.sessionId`
uniqueId	No	`userInfo.uniqueId`
targetUserRegistryInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the registryInfoType element type.
serverLocation	Yes	`registryInfo.serverLocation`
serverLocationType	Yes	`registryInfo.serverLocationType`
serverPort	Yes	`registryInfo.serverPort`
type	Yes	`registryInfo.type`
timestamp	Yes	`CommonBaseEvent/@creationTime`

Table 65. Elements used in AUDIT_WORKFLOW events (continued)

Element	Always in output	Abbreviated XPath
userInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the userInfoType element type.
appUserName	Yes	`userInfo.appUserName`
attributes	No	`userInfo.attributes`
callerList	No	`userInfo.callerList`
domain	No	`userInfo.domain`
location	No	`userInfo.location`
locationType	No	`userInfo.locationType`
realm	No	`userInfo.realm`
registryUserName	Yes	`userInfo.registryUserName`
sessionId	No	`userInfo.sessionId`
uniqueId	No	`userInfo.uniqueId`
userInputs	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the attributeType element type.
name	Yes	`attributeType.name`
source	No	`attributeType.source`
value	Yes	`attributeType.value`
workItemInfo	No	This element is a container element and has no valid XPath. A valid XPath requires a `values` declaration. This container element uses the children of the workItemInfoType element type.
id	Yes	`workItemInfoType.id`
type	Yes	`workItemInfoType.type`

Chapter 24. Reference information about elements and element types

This section defines the various elements and element types that are available for the common audit event types.

For each element and element type that can be used in an audit event, this documentation provides a description, the values that can be displayed in the output, and the XPath statement that can be used when modifying the shredder configuration file.

For information on the elements and element types described in this section, refer to the Common Base Event specification at the following Web site: http://www.eclipse.org/tptp/platform/documents/index.php

accessDecision

Reference information about the accessDecision element.

Description

Decision of the authorization call.

Values

String

The following strings are suggested values:

denied
Access was denied.

permitted
Access was permitted.

permittedWarning
Access was permitted in warning mode.

unknown
Cannot determine whether access is denied or not. May be due to a non-access error (configuration problem or internal problem) or because more access decision information is needed.

XPath

```
CommonBaseEvent/extendedDataElements[@name='accessDecision']/values
```

accessDecisionReason

Reference information about the accessDecisionReason element.

Description

Additional information about the access decision.

For example when `accessDecision='denied'`, provides the reason for the denial.

Values

String

The following strings are suggested values:

authnLevelUnauthorized

The user is not authenticated at a sufficiently high level to access the resource.

authzRuleUnauthorized

The authorization rule policy denied access.

delegateUnauthorized

Delegate principal is unauthorized to perform delegation.

qopUnauthorized

The communication channel being used to access the resource has an insufficient level of quality of protection.

reauthnUnauthorized

Access is denied until the user interactively re-authenticates.

timeOfDayUnauthorized

Access denied due to time of day policy.

unauthorized

Operation is not authorized. Use this only if you cannot provide a more specific reason.

XPath

```
CommonBaseEvent/extendedDataElements[@name='accessDecisionReason']/values
```

action

Reference information about the action element.

Description

The action being performed.

Values

String

- For the AUDIT_AUTHN event type, the following strings are suggested values:

  authentication

  An authentication operation. Note that multiple authentications can occur as part of a single login.

  credsRefresh

  Refresh of a credential. For example, in the case of Kerberos.

  login A login operation.

  reauthentication

  Re-authentication operation

  stepUp

  Step-up authentication.

  tokenIssue

  Used when the Trust Service issues a token on behalf of an identity.

  tokenReceipt

  Used when an incoming security token is validated by the Trust Service.

  switchUser

  A switch user operation.

- For the AUDIT_AUTHN_CREDS_MODIFY event type, the following strings are suggested values:

  credsCombine
  Caller is adding an additional user to a credential chain.

  credsModify
  Caller is creating a modified copy of existing user credentials.

  getCreds
  Caller is getting credentials based on user information.

  getCredsFromPAC
  Resolve credentials from transferable object (privilege attribute certificate [PAC]).

  getEntitlements
  Add to credentials using an entitlements service.

  getPAC
  Convert credentials to a transferable object (privilege attribute certificate [PAC]).

- For the AUDIT_AUTHN_TERMINATE event type, the following strings are suggested values:

  logout A logout operation.

  switchUserTerminate
  Used when the switch user session is ended.

- For the AUDIT_DATA_SYNC event type, the following strings are suggested values:

  reconcile
  Reconcile accounts. Example: Tivoli Identity Manager server sends request to the remote provisioning resource to synchronize account data into the Tivoli Identity Manager repository.

  unsolicitedNotification
  Notify of operations. Example: The remote provisioning resource sends notification to Tivoli Identity Manager server to notify changes on the account data.

- For the AUDIT_MGMT_CONFIG, AUDIT_MGMT_POLICY, AUDIT_MGMT_REGISTRY, and AUDIT_MGMT_RESOURCE event types, the following strings are suggested values:

  associate
  Associate entities. For example: user associated with groups, group associated with users, and policy associated with objects.

  challengeResponse
  Change the challenge and response configurations.

  changePolicyEnforcementAction
  Change the policy enforcement action of the management object. The currently allowable actions are:
  Correct
  Suspend
  Mark
  Non-Compliant

  checkAccess
  An authorization decision was made.

  create Create a management object.

  delegate
  Delegate authorities the user has to another user for a period of time specified.

delete
Delete a management object. For example, delete a file from the Trusted Computing Base.

disable
Disable an account for login activity.

disassociate
Disassociate entities. For example, disassociate a user from groups, disassociate a group from users, and disassociate a policy from objects.

enable
Enable an account for login activity.

markTrusted
Mark as trusted. For example, mark a file as trusted in the Trusted Computing Base.

markUntrusted
Mark as untrusted. For example, mark a file as untrusted in the Trusted Computing Base.

modify
Modify a management object.

passthru
Indicates that request is passed to another server.

passwordChange
Indicates a password change operation initiated by the administrator.

passwordPickup
Pick up password for account.

register
To register. For example, register a daemon with the kernel.

restore
To restore. For example, to restore a suspended user or account.

retire
To retire. For example, a federation is retired when it is no longer used. This is archived for future reference.

retrieve
A credential was retrieved.

show
Show a management object.

suspend
To suspend. For example, suspend a partner in a federation.

transfer
Transfer a user between different organization containers.

validate
To validate. For example, verify a security token representing a user.

- For the AUDIT_MGMT_PROVISIONING event type, the following strings are suggested values:

  add
  Provision a new account on the target resource identified by provisioningTargetInfo.

  adopt
  Adopt an orphan account identified by provisioningTargetInfo.

  changePassword
  Change password for an account identified by provisioningTargetInfo.

  delete
  Delete an account identified by provisioningTargetInfo.

  modify
  Modify an existing account identified by provisioningTargetInfo.

  passwordPickup
  Pick up password for an account identified by provisioningTargetInfo.

  restore
  Restore a suspended account identified by provisioningTargetInfo.

  suspend
  Suspend an existing account identified by provisioningTargetInfo.

- For the AUDIT_RESOURCE_ACCESS event type, the following strings are suggested values:

  fileExec
  A program execution occurred.

  fileTrace
  A file access occurred.

  httpRequest
  A request was made to access a given resource using HTTP.

- For the AUDIT_RUNTIME event type, the following strings are suggested values:

  auditLevelChange
  An audit or warning level change request has been sent to the server.

  auditStart
  Auditing has started for a server component.

  auditStop
  Auditing has stopped for a server component.

  contactRestored
  Restored contact. For example, the server has regained contact with the Security Access Manager user registry.

  heartbeatDown
  Heartbeat information that a server or API is down.

  heartbeatUp
  Heartbeat information that a server or API is up.

  lostContact
  Lost contact. For example, the server currently has no contact with the Security Access Manager user registry.

  monitor
  A process was adopted in to the set of monitored processes.

  start A server has successfully started.

  statistic
  Statistical information for a server for capacity planning purposes.

  stop A server has successfully stopped.

- For the AUDIT_RUNTIME_KEY event type, the following strings are suggested values:

  keyRetire
  The key has been retired.

  keyCRLInvalidated
  The CRL in the key is not valid.

  keyCertExpired
  The certificate in the key has expired.

  keySetInvalid
  The key has been set as not valid.

  keyCertExpirationCheck
  The expiration of the certificate has been checked.

- For the AUDIT_WORKFLOW event type, the following strings are suggested values:

  assign A work item is assigned and routed to a user.

  complete
  A work item is completed by the user.

  defer Additional time is given for the completion of the work item.

  delegate
  A work item is being delegated to another user.

  escalate
  A work item is being escalated as a result of timeout.

lock A work item is being locked by a user. Once a work item is locked, no other potential work item owner can perform operation on the work item.

unlock
A work item is unlocked by a user.

XPath

```
CommonBaseEvent/extendedDataElements[@name='action']/values
```

appName

Reference information about the appName element.

Description

Name of the application accessing the resource.

Values

String

For example, an Emacs program accessing a file resource.

XPath

```
CommonBaseEvent/extendedDataElements[@name='appName']/values
```

attributePermissionInfo

Reference information about the attributePermissionInfo element.

Description

A container for the information about access permissions on the attributes of the target.

This container uses the children of attributePermissionInfoType:
- attributePermissionInfoType.attributeNames
- attributePermissionInfoType.checked
- attributePermissionInfoType.denied
- attributePermissionInfoType.granted

XPath

No valid XPath for the shredder configuration file. A valid XPath requires a `values` declaration.

attributePermissionInfo.attributeNames

Reference information about the attributePermissionInfo.attributeNames element.

Description

List of attributes in which permission are being checked.

Values

String[]

XPath

The XPath accesses the first attributeNames element from an array of attributeNames elements.

```
CommonBaseEvent/extendedDataElements
[@name='attributePermissionInfo']/children[1]/children
[@name='attributeNames']/values[1]
```

attributePermissionInfo.checked

Reference information about the attributePermissionInfo.checked element.

Description

Permission that are being checked during the authorization call.

Values

String[]

XPath

The XPath accesses the first checked element from an array of checked elements.

```
CommonBaseEvent/extendedDataElements
[@name='attributePermissionInfo']/children[1]/children
[@name='checked']/values[1]
```

attributePermissionInfo

Reference information about the attributePermissionInfo.denied element.

Description

Permission that are denied.

Values

String[]

XPath

The XPath accesses the first denied element from an array of denied elements.

```
CommonBaseEvent/extendedDataElements
[@name='attributePermissionInfo']/children[1]/children
[@name='denied']/values[1]
```

attributePermissionInfo.granted

Reference information about the attributePermissionInfo.granted element.

Description

Permission that are granted.

Values

String[]

XPath

The XPath accesses the first granted element from an array of granted elements.

```
CommonBaseEvent/extendedDataElements
[@name='attributePermissionInfo']/children[1]/children
[@name='granted']/values[1]
```

attributes

Reference information about the attributes element.

Description

A container for the array of application-specific attributes for this event.

This element type represents an attribute that is associated with an entity, such as a user, application, or authorization rule.

This element uses the children of the attributeType element:
- attributes.name
- attributes.source
- attributes.value

XPath

No valid XPath for the shredder configuration file. A valid XPath requires a `values` declaration.

attributes.name

Reference information about the attributes.name element.

Description

Name of the attribute.

Values

Any arbitrary string

XPath

```
CommonBaseEvent/extendedDataElements[@name='attributes']/children[1]/children
[@name='name']/values[1]
```

attributes.source

Reference information about the attributes.source element.

Description

Source of the attribute.

Values

String

The following strings are suggested values:

application
Provided by the application.

authzRuleADI
Provided as an input for authorization rules.

user
Provided by the user.

XPath

```
CommonBaseEvent/extendedDataElements[@name='attributes']/children[1]/children
[@name='source']/values[1]
```

attributes.value

Reference information about the attributes.value element.

Description

Value of the attribute.

Values

Any arbitrary string

XPath

```
CommonBaseEvent/extendedDataElements[@name='attributes']/children[1]/children
[@name='value']/values[1]
```

auditMsg

Reference information about the auditMsg element.

Description

Message for this audit event.

Values

xsd:string

Any arbitrary string

Refer to the msg field in the Common Base Event specification.

XPath

```
CommonBaseEvent/extendedDataElements[@name='auditMsg']/values
```

auditMsgElement

Reference information about the auditMsgElement element.

Description

Information associated with message.

This container uses the field of msgDataElement and its children. For additional details, refer to the Common Base Event specification.

Values

cbe:msgDataElement

XPath

No valid XPath for the shredder configuration file. A valid XPath requires a `values` declaration.

auditTrailId

Reference information about the auditTrailId element.

Description

ID that allows audit events that belong to a given transaction to be correlated.

For example, this could be populated using the propagationToken in WebSphere Application Server.

Values

Any arbitrary string

XPath

```
CommonBaseEvent/extendedDataElements[@name='auditTrailId']/values
```

authenProvider

Reference information about the authenProvider element.

Description

Provider of the authentication service.

Values

Any arbitrary string

XPath

```
CommonBaseEvent/extendedDataElements[@name='authenProvider']/values
```

authnType

Reference information about the authnType element.

Description

Provider of the authentication service.

Values

Any arbitrary string

The following strings are suggested values:

basicAuth
Browser authentication based on user ID and password.

challengeResponse
Challenge and response authentication.

digest Digest-based authentication.

form Form-based authentication.

identityAssertion
Authentication based on identity assertion.

kerberos
Authentication based on Kerberos credentials.

ldap_v3.0
Authentication using the LDAP protocol.

ltpa Lightweight third-party authentication.

sslAuthn
SSL-based authentication.

tokenAccessManagerCred
Authentication based on Security Access Manager credentials.

tokenLiberty
Authentication based on a Liberty token.

tokenSAML
Authentication based on a SAML token.

tokenUserName
Authentication based on user name based token.

trustAssociation
Authentication based on trust association.

XPath

```
CommonBaseEvent/extendedDataElements[@name='authnType']/values
```

authnTypeVersion

Reference information about the authnTypeVersion element.

Description

Version of the authentication type.

Values

String form of the version number

XPath

```
CommonBaseEvent/extendedDataElements[@name='authnTypeVersion']/values
```

complianceStatus

Reference information about the complianceStatus element.

Description

Status of compliance.

Values

String

The following strings are suggested values:

compliant
The reconciled account on the provisioning resource complies with the specified security policy.

disallowed
The reconciled account is not allowed by a provisioning policy.

nonCompliant
The reconciled account on the provisioning resource does not comply with the specified security policy.

orphan
No owner can be found for the reconciled account.

XPath

```
CommonBaseEvent/extendedDataElements[@name='complianceStatus']/values
```

endTime

Reference information about the endTime element.

Description

End time of the operation.

Values

xsd:DateTime

Refer to the creationTime field in the Common Base Event specification.

XPath

```
CommonBaseEvent/extendedDataElements[@name='endTime'][@type='dateTime']/values
```

extensionName

Reference information about the extensionName element.

Description

The event type.

This information relates to the following line in the `CARSShredder.conf` file:

```
cars_t_event, eventType, "'event_type'"
```

Values

String

The actual name of the event type, which is one of the following literal values:

- `AUDIT_AUTHN_CREDS_MODIFY`
- `AUDIT_AUTHN_MAPPING`
- `AUDIT_AUTHN_TERMINATE`
- `AUDIT_AUTHN`
- `AUDIT_AUTHZ`
- `AUDIT_COMPLIANCE`
- `AUDIT_DATA_SYNC`
- `AUDIT_MGMT_CONFIG`
- `AUDIT_MGMT_POLICY`
- `AUDIT_MGMT_PROVISIONING`
- `AUDIT_MGMT_REGISTRY`
- `AUDIT_MGMT_RESOURCE`
- `AUDIT_PASSWORD_CHANGE`
- `AUDIT_RESOURCE_ACCESS`
- `AUDIT_RUNTIME`
- `AUDIT_RUNTIME_KEY`
- `AUDIT_WORKFLOW`

XPath

`event_type`

For example, to specify the AUDIT_AUTHN event type, specify:

```
"'AUDIT_AUTHN'"
```

fixDescription

Reference information about the fixDescription element.

Description

Description of specific fix. For example, “Apply patch *xyz*”.

Values

String

Any arbitrary string allowed by the application.

XPath

```
CommonBaseEvent/extendedDataElements[@name='fixDescription']/values
```

fixId

Reference information about the fixId element.

Description

Identifier of specific fix.

Values

String

Any arbitrary string allowed by the application.

XPath

```
CommonBaseEvent/extendedDataElements[@name='fixId']/values
```

globalInstanceId

Reference information about the globalInstanceId element.

Description

An internal identifier for an audit event as shown in the XML output.

This information is not related to the following line in the `CARSShredder.conf` file:

```
cars_t_event, event_id, #GLOBAL_ID
```

httpURLInfo

Reference information about the httpURLInfo element.

Description

The container for information about the HTTP request.

This container uses the children of HTTPURLInfoType:
- HTTPURLInfoType.method
- HTTPURLInfoType.requestHeaders
- HTTPURLInfoType.responseCode
- HTTPURLInfoType.responseHeaders
- HTTPURLInfoType.url

XPath

No valid XPath for the shredder configuration file. A valid XPath requires a `values` declaration.

HTTPURLInfo.method

Reference information about the HTTPURLInfo.method element.

Description

Method used.

Values

String

Methods allowed by the HTTP protocol (for example, POST or GET). The following strings are suggested values:

GET Passed in information using the HTTP GET method.

POST Passed in information using the HTTP POST method.

XPath

```
CommonBaseEvent/extendedDataElements[@name='HTTPURLInfo']/children
[@name='method']/values
```

HTTPURLInfo.requestHeaders

Reference information about the HTTPURLInfo.requestHeaders element.

Description

HTTP request headers given by the client.

Values

String

XPath

```
CommonBaseEvent/extendedDataElements[@name='HTTPURLInfo']/children
[@name='requestHeaders']/values
```

HTTPURLInfo.responseCode

Reference information about the HTTPURLInfo.responseCode element.

Description

Response code returned by the server.

Values

Integer

XPath

```
CommonBaseEvent/extendedDataElements[@name='HTTPURLInfo']/children
[@name='responseCode']/values
```

HTTPURLInfo.responseHeaders

Reference information about the HTTPURLInfo.responseHeaders element.

Description

HTTP response headers returned by the server.

Values

String

XPath

```
CommonBaseEvent/extendedDataElements[@name='HTTPURLInfo']/children
[@name='responseHeaders']/values
```

HTTPURLInfo.url

Reference information about the HTTPURLInfo.url element.

Description

URL of the HTTP request.

Values

String

XPath

```
CommonBaseEvent/extendedDataElements[@name='HTTPURLInfo']/children
[@name='url']/values
```

keyLabel

Reference information about the keyLabel element.

Description

Indicates the key or certificate label.

Values

Any arbitrary string

XPath

```
CommonBaseEvent/extendedDataElements[@name='keyLabel']/values
```

lifetime

Reference information about the lifetime element.

Description

Indicates when a certificate will expire.

Values

xsd:DateTime

Refer to the creationTime field in the Common Base Event specification.

XPath

```
CommonBaseEvent/extendedDataElements[@name='lifetime']/values
```

location

Reference information about the location element.

Description

Physical location of the key database.

Values

xsd:string

Refer to the location field in the Common Base Event specification.

XPath

```
CommonBaseEvent/extendedDataElements[@name='location']/values
```

locationType

Reference information about the locationType element.

Description

Type of location.

Values

xsd:Name

Refer to the locationType field in the Common Base Event specification.

XPath

```
CommonBaseEvent/extendedDataElements[@name='locationType']/values
```

loginTime

Reference information about the loginTime element.

Description

The time that the login occurred.

Values

xsd:DateTime

Refer to the creationTime field in the Common Base Event specification.

XPath

```
CommonBaseEvent/@creationTime
```

mappedRealm

Reference information about the mappedRealm element.

Description

Indicate the realm after mapping.

Values

Any arbitrary string

XPath

```
CommonBaseEvent/extendedDataElements[@name='mappedRealm']/values
```

mappedSecurityDomain

Reference information about the mappedSecurityDomain element.

Description

Indicate the security domain after mapping.

Values

Any arbitrary string

XPath

```
CommonBaseEvent/extendedDataElements[@name='mappedSecurityDomain']/values
```

mappedUserName

Reference information about the mappedUserName element.

Description

Indicate the user name after mapping.

Values

Any arbitrary string

XPath

```
CommonBaseEvent/extendedDataElements[@name='mappedUserName']/values
```

membershipInfo

Reference information about the membershipInfo element.

Description

The container for list of memberships to which the policy applies.

The element uses the children of the membershipInfo element:
- membershipInfoType.id
- membershipInfoType.name
- membershipInfoType.type

XPath

No valid XPath for the shredder configuration file. A valid XPath requires a `values` declaration.

memberships.id

Reference information about the memberships.id element.

Description

Unique identifier of the member.

Values

String

For example, distinguished name of a role.

XPath

The XPath statement assumes the first membership element from an array of membership elements.

```
CommonBaseEvent/extendedDataElements
[@name='memberships']/children[1]/children
[@name='id']/values
```

memberships.name

Reference information about the memberships.name element.

Description

Name of the member.

Values

String

XPath

The XPath statement assumes the first membership element from an array of membership elements.

```
CommonBaseEvent/extendedDataElements
[@name='memberships']/children[1]/children
[@name='name']/values
```

memberships.type

Reference information about the memberships.type element.

Description

Membership type.

Values

String

The following strings are suggested values:

all Applies to all users.

orgContainer
Applies to users that belong in a given organization container.

other Is not one of the other types.

role Applies to users that belong in a given role.

XPath

The XPath statement assumes the first membership element from an array of membership elements.

```
CommonBaseEvent/extendedDataElements
[@name='memberships']/children[1]/children
[@name='type']/values
```

message

Reference information about the message element.

Description

Generated message that describes specifics about the violation. Can include dynamically inserted information. Example:

Invalid ACL for
c:\winnt\repair:
Account: BUILTIN\users

Values

String

Any arbitrary string allowed by the application.

XPath

```
CommonBaseEvent/extendedDataElements[@name='message']/values
```

mgmtInfo

Reference information about the mgmtInfo element.

Description

The container for information about this management operation.

This element type represents information that is common for events that are related to management operations, such as managing policies, resources, registry objects, and so forth.

This element uses the children of mgmtInfoType:
- mgmtInfoType.command
- mgmtInfoType.targetInfo

XPath

No valid XPath for the shredder configuration file. A valid XPath requires a `values` declaration.

mgmtInfo.command

Reference information about the mgmtInfo.command element.

Description

The application-specific command being performed. The command is particularly useful for modify actions to pinpoint what is being modified.

Values

String

An application-specific string that represents the command. Examples:

- Key user modify:

  modifyPassword
  modifyAccountValid
  modifyPasswordValidKey
- Policy modify:

  modifyPolicyMaxLoginFailures
  modifyPolicyMaxAccountAge
  modifyPolicyMaxPasswordAge
  modifyPolicyTimeOfDayAccess
- ACL modify:

  modifyACLSetAttribute
  modifyACLDelAttribute
- POP modify:

  modifyPOPSetAttribute
  modifyPOPDelAttribute
- protectedObject modify:

  modifyObjectDelAttribute
  modifyObjectSetAttribute

XPath

```
CommonBaseEvent/extendedDataElements[@name='mgmtInfo']/children
[@name='command']/values
```

mgmtInfo.targetInfo

Reference information about the mgmtInfo.targetInfo element.

Description

Information about the target resource of this operation.

Values

targetInfoType

XPath

Refer to "targetInfoType" on page 339 for details.

originalRealm

Reference information about the originalRealm element.

Description

Indicate the realm before mapping.

Values

Any arbitrary string

XPath

```
CommonBaseEvent/extendedDataElements[@name='originalRealm']/values
```

originalSecurityRealm

Reference information about the originalSecurityRealm element.

Description

Indicate the security domain before mapping.

Values

Any arbitrary string

XPath

```
CommonBaseEvent/extendedDataElements[@name='originalSecurityRealm']/values
```

originalUserName

Reference information about the originalUserName element.

Description

Indicate the user name before mapping.

Values

Any arbitrary string

XPath

```
CommonBaseEvent/extendedDataElements[@name='originalUserName']/values
```

outcome

Reference information about the outcome element.

Description

A container for the outcome of the action for which the audit record is generated.

This element type identifies a component that is the source of the event or reports an event, and defines the outcome of the event being audited.

This element uses the children of auditOutcomeType:
- outcome.failureReason
- outcome.majorStatus
- outcome.minorStatus
- outcome.result

XPath

No valid XPath for the shredder configuration file. A valid XPath requires a `values` declaration.

outcome.failureReason

Reference information about the outcome.failureReason element.

Description

Additional information about the outcome.

Values

Any arbitrary string.

The outcome element contains the failureReason element. The values for the failureReason elements are event-specific. The following strings are some of the suggested values:

accountDisabled
User's account has been disabled.

accountDisabledRetryViolation
Retry maximum has been violated for authentications that are not valid. The account has been disabled in the registry.

accountExpired
User account has expired.

accountLockedOutMaxLoginFail
User account has been temporarily locked out due to too many failed login attempts. Lock time interval has not elapsed.

accountLockedOutRetryViolation
Invalid authentication retry maximum has been violated. The account has been temporarily locked out.

accountMaxInactiveElapsed
Maximum inactive days has elapsed for the account.

accountUnlocked
User account was unlocked because lock time interval has elapsed.

authenticationFailure
Authentication failed. Use this value when you do not have a more specific value for this audit element.

certificateFailure
A client certificate could not be authenticated.

invalidUserName
The supplied user name does not exist in the registry.

invalidUserPassword
The password associated with the given user name is incorrect.

mappingFailure
The login data entered could not be mapped to an application-specific user.

nextToken
Next token required for authentication.

passwordChangeMaxIntervalElapsed
Maximum time interval since last password change has elapsed.

passwordChangeMinIntervalUnexpired
Minimum time interval required between password changes has not elapsed.

passwordContainOld
Password contains the old password or is contained in the old password.

passwordExpired
The user's password has expired and no further grace logins remain.

passwordFirstLastNumeric
Password contains a numeric first or last character.

passwordMaxCharOld
Password exceeds the allowed number of consecutive characters that are common with the previous password.

passwordMaxRepeated
Password exceeds the maximum allowed number of repeated characters.

passwordMinAlphabetic
Password does not contain the required minimum number of alphabetic characters.

passwordMinAlphabeticLower
Password does not contain the required minimum number of lowercase characters.

passwordMinAlphabeticUpper
Password does not contain the required minimum number of uppercase characters.

passwordMinAlphanumeric
Password does not contain the required minimum number of alphanumeric characters

passwordMinNumeric
Password does not contain the required minimum number of numeric characters.

passwordMinSpecial
Password does not contain the required minimum number of special characters.

passwordNumCharViolation
Password does not contain the required number of characters.

passwordOldReused
Password is a recently used old password.

passwordUserName
Password contains the user name or is contained in the user name.

pinRequired
A PIN must be assigned to enable account.

policyAllowedAccess
All login policy checks permitted access.

policyViolation
Login rejected due to policy violation.

policyViolationMaxLoginsReached
Login rejected because maximum number of concurrent logins reached.

policyViolationTOD
Authentication denied at this time of the day.

tokenExpired
The lifetime for the token has expired.

tokenNotSupported
The given token is not a supported type.

tokenNotInValidFormat
The given token was not in the expected format or was corrupted.

tokenNotValidYet
The token is not valid yet.

tokenSignatureValidationFailed
The signature for the token was not valid.

usernameMismatch
In the case of `reauthentication` or `stepUp` authentication, the given user name does not match the current user name.

When a suggested value is not available, use the string “Unknown Failure Reason”.

XPath

```
CommonBaseEvent/extendedDataElements[@name='outcome']/children
[@name='failureReason']/values
```

outcome.majorStatus

Reference information about the outcome.majorStatus element.

Description

Major status code. Typically, majorStatus will be zero when result is SUCCESSFUL, and some nonzero value when it is not.

Values

Any integer

XPath

```
CommonBaseEvent/extendedDataElements[@name='outcome']/children
[@name='majorStatus']/values
```

outcome.minorStatus

Reference information about the outcome.minorStatus element.

Description

Minor status code. Typically, minorStatus will be zero when result is SUCCESSFUL, and some non-zero value when it is not.

Values

Any integer

XPath

```
CommonBaseEvent/extendedDataElements[@name='outcome']/children
[@name='minorStatus']/values
```

outcome.result

Reference information about the outcome.result element.

Description

Overall status of the event commonly used for filtering. Use UNSUCCESSFUL when an error condition arose which prevented normal processing, and SUCCESSFUL for normal processing.

Values

Same as the successDisposition field in the Situation types in the Common Base Event specification.
- SUCCESSFUL
- UNSUCCESSFUL

XPath

```
CommonBaseEvent/extendedDataElements[@name='outcome']/children
[@name='result']/values
```

partner

Reference information about the partner element.

Description

End time of the operation.

Values

xsd:DateTime

XPath

```
CommonBaseEvent/extendedDataElements[@name='partner']/values
```

perfInfo

Reference information about the perfInfo element.

Description

A container that represents performance and statistical data This information that can be helpful during capacity planning activities.

This element uses the children of perfInfoType:
- perfInfo.aggregate
- perfInfo.description
- perfInfo.name
- perfInfo.maxValue
- perfInfo.minValue
- perfInfo.numDataPoints
- perfInfo.unit
- perfInfo.value

XPath

No valid XPath for the shredder configuration file. A valid XPath requires a `values` declaration.

perfInfo.aggregate

Reference information about the perfInfo.aggregate element.

Description

Operation for combining with other statistic events.

Values

String

The following strings are suggested values:

addition
: When combining with another statistic that measures the same data, then the values of the data should be added together.

average
: When combining with another statistic that measures the same data, then the values of the data should be averaged.

XPath

```
CommonBaseEvent/extendedDataElements[@name='perfInfo']/children
[@name='aggregate']/values
```

perfInfo.description

Reference information about the perfInfo.description element.

Description

Description of the statistic.

Values

Any arbitrary string

XPath

```
CommonBaseEvent/extendedDataElements[@name='perfInfo']/children
[@name='description']/values
```

perfInfo.name

Reference information about the perfInfo.name element.

Description

Name of the statistic.

Values

Any arbitrary string

XPath

```
CommonBaseEvent/extendedDataElements[@name='perfInfo']/children
[@name='name']/values
```

perfInfo.maxValue

Reference information about the perfInfo.maxValue element.

Description

Maximum value among all data points.

Values

Long

XPath

```
CommonBaseEvent/extendedDataElements[@name='perfInfo']/children
[@name='maxValue']/values
```

perfInfo.minValue

Reference information about the perfInfo.minValue element.

Description

Minimum value among all data points.

Values

Long

XPath

```
CommonBaseEvent/extendedDataElements[@name='perfInfo']/children
[@name='minValue']/values
```

perfInfo.numDataPoints

Reference information about the perfInfo.numDataPoints element.

Description

Number of data points gathered.

Values

Integer

XPath

```
CommonBaseEvent/extendedDataElements[@name='perfInfo']/children
[@name='numDataPoints']/values
```

perfInfo.unit

Reference information about the perfInfo.unit element.

Description

Unit of measurement for the value.

Values

Any arbitrary string

XPath

```
CommonBaseEvent/extendedDataElements[@name='perfInfo']/children
[@name='unit']/values
```

perfInfo.value

Reference information about the perfInfo.value element.

Description

Value of the statistic.

Values

Long

XPath

```
CommonBaseEvent/extendedDataElements[@name='perfInfo']/children
[@name='value']/values
```

permissionInfo

Reference information about the permissionInfo element.

Description

A container represents information about access permissions.

This element uses the children of permissionInfoType:
- permissionInfoType.checked
- permissionInfoType.denied
- permissionInfoType.granted
- permissionInfoType.J2EERolesChecked
- permissionInfoType.J2EERolesGranted

XPath

No valid XPath for the shredder configuration file. A valid XPath requires a `values` declaration.

permissionInfo.checked

Reference information about the permissionInfo.checked element.

Description

Permission that are being checked during the authorization call.

Values

String[]

Any arbitrary string allowed by the application can be provided as an element of the String[].

XPath

The XPath accesses the first checked element from an array of checked elements.

```
CommonBaseEvent/extendedDataElements[@name='permissionInfo']/children
[@name='checked']/values[1]
```

permissionInfo.denied

Reference information about the permissionInfo.denied element.

Description

Permissions that are denied out of the ones requested.

Values

String[]

Any arbitrary string allowed by the application can be provided as an element of the String[].

XPath

The XPath accesses the first denied element from an array of denied elements.

```
CommonBaseEvent/extendedDataElements[@name='permissionInfo']/children
[@name='denied']/values[1]
```

permissionInfo.granted

Reference information about the permissionInfo.granted element.

Description

Permissions that are granted.

Values

String[]

Any arbitrary string allowed by the application can be provided as an element of the String[].

XPath

The XPath accesses the first granted element from an array of granted elements.

```
CommonBaseEvent/extendedDataElements[@name='permissionInfo']/children
[@name='granted']/values[1]
```

permissionInfo.J2EERolesChecked

Reference information about the permissionInfo.J2EERolesChecked element.

Description

J2EE roles being checked.

Values

String[]

Any arbitrary string allowed by the application can be provided as an element of the String[].

XPath

The XPath accesses the first J2EERolesChecked element from an array of J2EERolesChecked elements.

```
CommonBaseEvent/extendedDataElements[@name='permissionInfo']/children
[@name='J2EERolesChecked']/values[1]
```

permissionInfo.J2EERolesGranted

Reference information about the permissionInfo.J2EERolesGranted element.

Description

J2EE roles granted.

Values

String[]

Any arbitrary string allowed by the application can be provided as an element of the String[].

XPath

The XPath accesses the first J2EERolesGranted element from an array of J2EERolesGranted elements.

```
CommonBaseEvent/extendedDataElements[@name='permissionInfo']/children
[@name='J2EERolesGranted']/values[1]
```

policyDescription

Reference information about the policyDescription element.

Description

Description of the policy that contains violation specification.

Values

String

Any arbitrary string allowed by the application.

XPath

```
CommonBaseEvent/extendedDataElements[@name='policyDescription']/values
```

policyInfo

Reference information about the policyInfo element.

Description

A container for information about the policy object, which can includes policies that are attached to the resource or policies that are the container of a resource.

This element type represents a policy associated with an authorization resource or policy management event.

The element uses the children of policyInfoType:
- policyInfo.attributes
- policyInfo.branch
- policyInfo.description
- policyInfo.name
- policyInfo.type

XPath

No valid XPath for the shredder configuration file. A valid XPath requires a `values` declaration.

policyInfo.attributes

Reference information about the policyInfo.attributes element.

Description

Attributes associated with a policy.

Values

attributeType[]

See "attributes" on page 296 for details.

XPath

The XPath accesses the first source element from an array of attributes elements.

```
CommonBaseEvent/extendedDataElements
[@name='policyInfo']/children[5]/children
[@name='source']/values
```

Note: The index is `5`, for the `attributes` element must come after the`branch`, `description`, `name`, and `type` elements:

policyInfo.branch

Reference information about the policyInfo.branch element.

Description

Name of the branch to which the policy applies.

Values

String

For example: The product lets you group the policy for similar machines under user-defined policy branches.

XPath

```
CommonBaseEvent/extendedDataElements[@name='policyInfo']/children
[@name='branch']/values
```

policyInfo.description

Reference information about the policyInfo.description element.

Description

Description of the policy.

Values

String

XPath

```
CommonBaseEvent/extendedDataElements[@name='policyInfo']/children
[@name='description']/values
```

policyInfo.name

Reference information about the policyInfo.name element.

Description

Name of the policy.

Values

String

XPath

```
CommonBaseEvent/extendedDataElements[@name='policyInfo']/children
[@name='name']/values
```

policyInfo.type

Reference information about the policyInfo.type element.

Description

Type of the policy.

Values

String

The following strings are suggested values:

accountPolicy
Account policy:
- Account expiry date
- Maximum account age
- Time of day (TOD) access

acl Access control list.

action Represents a permission.

actionGroup
Represents a collection of permissions.

authzRule
Authorization rule.

federation
A collection of groups or organizations participating in a trust relationship.

identityPolicy
Specifies how identities, or user IDs, should be generated when provisioning one or more resource.

key A cryptographic key, either symmetric or asymmetric.

loginPolicy
Policy that controls login behavior:
- Login failure count
- Login disable time interval

partner
A group or organization participating in a federation.

passwordPolicy
A set of rules in which all passwords for one or more services must conform.

policy Generic policy value to be used for policies not defined in the other values.

pop Protected object policy controls:
- Audit level
- Additional attributes
- Quality of protection (QoP)

provisioningPolicy
Used to associate one or multiple groups of users with one or multiple entitlements. The group of users is usually identified by organization or organization role. The entitlement is a construct to define a set of permissions, or privileges, on a managed provisioning resource.

serviceSelectionPolicy
Used in situations where the instance of a provisioning resource, on which the provisioning of an account is to take place, will be determined dynamically based on account owner's attributes.

spsModule
A Single Sign-On (SSO) Protocol Service module (for example, the Liberty module).

stsChain
A grouping of Security Token Service (STS) module instances.

stsModule
Security Token Service (STS) module (for example, SAML module).

XPath

```
CommonBaseEvent/extendedDataElements[@name='policyInfo']/children
[@name='type']/values
```

policyName

Reference information about the policyName element.

Description

Name of policy. Example: “ITCS104AIX”.

Values

String

Any arbitrary string allowed by the application.

XPath

```
CommonBaseEvent/extendedDataElements[@name='policyName']/values
```

progName

Reference information about the progName element.

Description

Name of the program that is involved in the event.

Values

Any arbitrary string

XPath

```
CommonBaseEvent/extendedDataElements[@name='progName']/values
```

provisioningInfo

Reference information about the provisioningInfo element.

Description

A container for the information about a provisioned resource that is the target of the operation.

This element uses the children of provisioningInfoType:
- provisioningInfoType.accountId
- provisioningInfoType.resourceId
- provisioningInfoType.resourceType

XPath

No valid XPath for the shredder configuration file. A valid XPath requires a `values` declaration.

provisioningInfo.accountId

Reference information about the provisioningInfo.accountId element.

Description

Unique identifier of the target account.

Values

String

XPath

```
CommonBaseEvent/extendedDataElements[@name='provisioningInfo']/children
[@name='accountId']/values
```

provisioningInfo.resourceId

Reference information about the provisioningInfo.resourceId element.

Description

Unique identifier of the target resource.

Values

String

XPath

```
CommonBaseEvent/extendedDataElements[@name='provisioningInfo']/children
[@name='resourceId']/values
```

provisioningInfo.resourceType

Reference information about the provisioningInfo.resourceType element.

Description

Type of the target. For example, the type of the user, or the type of the provisioning resource.

Values

An arbitrary string.

See suggested values for “resourceInfo.type” on page 331 audit element.

XPath

```
CommonBaseEvent/extendedDataElements[@name='provisioningInfo']/children
[@name='resourceType']/values
```

provisioningTargetInfo

Reference information about the provisioningTargetInfo element.

Description

A container for target provisioning account.

This element uses the children of provisioningInfoType:
- provisioningInfoType.accountId
- provisioningInfoType.resourceId
- provisioningInfoType.resourceType

XPath

No valid XPath for the shredder configuration file. A valid XPath requires a `values` declaration.

recommendation

Reference information about the recommendation element.

Description

Provides information related to remedial actions to take to protect against the vulnerability.

Values

String

Any arbitrary string allowed by the application.

XPath

```
CommonBaseEvent/extendedDataElements[@name='recommendation']/values
```

registryInfo

Reference information about the registryInfo element.

Description

A container for information about the user registry that is involved in the operation.

This element uses the children of the registryInfoType element:
- registryInfo.serverLocation
- registryInfo.serverLocationType
- registryInfo.serverPort
- registryInfo.type

XPath

No valid XPath for the shredder configuration file. A valid XPath requires a `values` declaration.

registryInfo.serverLocation

Reference information about the registryInfo.serverLocation element.

Description

Location of the registry server.

Values

xsd:string

Refer to the location field in the Common Base Event specification.

XPath

```
CommonBaseEvent/extendedDataElements[@name='registryInfo']/children
[@name='serverLocation']/values
```

registryInfo.serverLocationType

Reference information about the registryInfo.serverLocationType element.

Description

Type of server location.

Values

xsd:Name

Refer to the locationType field in the Common Base Event specification.

XPath

```
CommonBaseEvent/extendedDataElements[@name='registryInfo']/children
[@name='serverLocationType']/values
```

registryInfo.serverPort

Reference information about the registryInfo.serverPort element.

Description

Port on which the registry server is listening.

Values

String

Port number

XPath

```
CommonBaseEvent/extendedDataElements[@name='registryInfo']/children
[@name='serverPort']/values
```

registryInfo.type

Reference information about the registryInfo.type element.

Description

Type of registry.

Values

String

The following strings are suggested values:

ActiveDir
Active Directory registry.

AIX AIX user registry.

LDAP LDAP registry.

Linux Linux user registry.

Solaris
Solaris user registry.

Windows
Windows user registry.

XPath

```
CommonBaseEvent/extendedDataElements[@name='registryInfo']/children
[@name='type']/values
```

registryObjectInfo

Reference information about the registryObjectInfo element.

Description

A container for information about the registry object that is being managed.

This container uses the children of the registryObjectInfoType element:

- registryObjectInfo.attributes
- registryObjectInfo.description
- registryObjectInfo.name
- registryObjectInfo.registryName
- registryObjectInfo.type

XPath

No valid XPath for the shredder configuration file. A valid XPath requires a `values` declaration.

registryObjectInfo.attributes

Reference information about the registryObjectInfo.attributes element.

Description

Attributes associated with a registry object.

Values

attributeType[]

See “attributes” on page 296 for details.

XPath

The XPath accesses the first name element from an array of attributes elements.

```
CommonBaseEvent/extendedDataElements
[@name='registryObjectInfo']/children[5]
[@name='name']/values
```

Note: The index is `5`, for the `attributes` element must come after the`description`, `name`, `registryName`, and `type` elements:

registryObjectInfo.description

Reference information about the registryObjectInfo.description element.

Description

Description of the policy.

Values

String

XPath

```
CommonBaseEvent/extendedDataElements[@name='registryObjectInfo']/children
[@name='description']/values
```

registryObjectInfo.name

Reference information about the registryObjectInfo.name element.

Description

Application name for the registry object.

Values

String

Any string allowed by the application.

XPath

```
CommonBaseEvent/extendedDataElements[@name='registryObjectInfo']/children
[@name='name']/values
```

registryObjectInfo.registryName

Reference information about the registryObjectInfo.registryName element.

Description

Registry name for the registry object.

Values

String

Any string allowed by the registry.

XPath

```
CommonBaseEvent/extendedDataElements[@name='registryObjectInfo']/children
[@name='registryName']/values
```

registryObjectInfo.type

Reference information about the registryObjectInfo.type element.

Description

Type of the registry object.

Values

String

The following strings are suggested values:

domain
A registry object that represents a domain.

group
A registry object that represents a group.

gsoResource
A registry object that represents a global sign-on (GSO) resource.

orgContainer
Identifies the organization hierarchy for the user.

user
A registry object that represents a user.

XPath

```
CommonBaseEvent/extendedDataElements[@name='registryObjectInfo']/children
[@name='type']/values
```

reporterComponentId

Reference information about the reporterComponentId element.

Description

A container for the reporter of the audit record on behalf of the source component. This container element is used when the reporting component is different from the source component.

When displayed in output, this element uses the children of the auditComponentIdType element:

- application
- component
- componentIdType
- componentType
- executionEnvironment
- instanceId
- location
- locationType

- processed
- subcomponent
- threadId

XPath

This element, nor its children, should be defined in the shredder configuration file. These elements are generated by the code.

resourceInfo

Reference information about the resourceInfo element.

Description

The container for information about the resource that is being accessed or that to which the policy applies.

This element uses the children of the resourceInfoType element:

- resourceInfo.attributes
- resourceInfo.nameInApp
- resourceInfo.nameInPolicy
- resourceInfo.type

XPath

No valid XPath for the shredder configuration file. A valid XPath requires a `values` declaration.

resourceInfo.attributes

Reference information about the resourceInfo.attributes element.

Description

Array of attributes for the resource.

Values

attributeType []

Refer to "attributes" on page 296 for details.

XPath

The XPath accesses the first name element from an array of attributes elements.

```
CommonBaseEvent/extendedDataElements
[@name='registryObjectInfo']/children[4]
[@name='name']/values
```

Note: The index is `4`, for the `attributes` element must come after the`nameInApp`, `nameInPolicy`, and `type` elements:

```
CommonBaseEvent/extendedDataElements[@name='resourceInfo']/children
[@name='attributes']/values
```

resourceInfo.nameInApp

Reference information about the resourceInfo.nameInApp element.

Description

Name of the resource in the context of the application.

Values

Any arbitrary string

User "Not Available" when not available.

XPath

```
CommonBaseEvent/extendedDataElements[@name='resourceInfo']/children
[@name='nameInApp']/values
```

resourceInfo.nameInPolicy

Reference information about the resourceInfo.nameInPolicy element.

Description

Name of the resource when applying a policy to it. For example, Security Access Manager protected object name.

Values

Any arbitrary string

User "Not Available" when not available.

XPath

```
CommonBaseEvent/extendedDataElements[@name='resourceInfo']/children
[@name='nameInPolicy']/values
```

resourceInfo.type

Reference information about the resourceInfo.type element.

Description

Type of the resource.

Values

String

The following strings are suggested values:

application
An application such as Security Access Manager server, Directory Server, Identity Manager server or any executable process.

file File system resource. Example: /OSSEAL/policy-branch/File/filespec

group Used to group users for Role Based Access Control.

identityPolicy
Identify policy specifies how user identities should be generated when provisioning one or more resource.

junction
Describes a WebSEAL junction.

login
Policies related to login. For example, password expiry, account suspension due to failed login attempts, or account lockouts due to account inactivity.

management
Authorization of a management command. The specific management object type is contained in the resourceName.

messageQueue
A message queue.

netIncoming
Incoming network accesses are controlled by network resources: NetIncoming resource:/OSSEAL/policy-branch/NetIncoming/protocol[/service[/host]]

netOutgoing
Outgoing network accesses are controlled by the following network resource. NetOutgoing resource:/OSSEAL/policy-branch/NetOutgoing/[/hostspec[/protocol[/service]]]

orgContainer
The organization container defines the organization hierarchy for the managed resources.

passwordPolicy
Specifies a set of rules in which all passwords for one or more services must conform. For example, password strength and password aging.

policyUpdate
Indicates a policy update. Example: The product has received a policy update (downloaded from the policy database).

protectedResource
A generic value for a protected resource. For example, Security Access Manager protected object or Security Access Manager protected object space.

provisioningAccount
Represents a user's identity on the target provisioning resource.

provisioningPolicy
Used to associate one or multiple groups of users with one or multiple entitlements. The group of users is usually identified by organization or organization role. The entitlement is a construct to define a set of permissions, or privileges, on a managed provisioning resource.

provisioningResource
A resource for which Identity Provisioning is enabled.

serviceSelectionPolicy
Used in situations where the instance of a provisioning resource, on which the provisioning of an account is to take place, will be determined dynamically based on account owner's attributes.

sudo
Describe commands that require more stringent access control than whether or not a particular program can be run. Sudo commands allow access control based not only on a command but also on the parameters passed to that command.

You can use Sudo commands to remove the requirements for a user to become the root user on a system in order to perform administrative tasks.

Sudo resources are identified in the Security Access Manager namespace in the following way: /OSSEAL/policy-branch/Sudo/*sudo-command*[/*sudo-orglass*]

surrogate
Surrogate resources. Operations that can change the user identity or group identity of a process are referred to as surrogate operations and are controlled by resources of type surrogate. Surrogate resource names follow the form: /OSSEAL/policy-branch/Surrogate/User/*user-name*.

tcb Trusted Computing Base resources.

workflowTemplate
Defines the flow of a business workflow process.

url An absolute URL identifying the resource accessed. Use the File resource type for file:// URLs.

user The user entity that application manages in the registry.

XPath

```
CommonBaseEvent/extendedDataElements[@name='resourceInfo']/children
[@name='type']/values
```

sequenceNumber

Reference information about the sequenceNumber element.

Description

An internal identifier for an audit event as shown in the XML output.

This information is not related to the following line in the `CARSShredder.conf` file:

```
cars_t_event, cars_seq_number, #RECORD_ID
```

severity

Reference information about the severity element.

Description

Identifies severity of the violation.

Values

String

The following strings are suggested values:

high Violation of high severity.

low Violation of low severity.

medium
Violation of medium severity.

XPath

```
CommonBaseEvent/extendedDataElements[@name='severity']/values
```

sourceComponentId

Reference information about the sourceComponentId element.

Description

A container for the information about what originated the audit record.

When displayed in output, this element uses the children of the auditComponentIdType element:

- sourceComponentId/@application
- sourceComponentId/@component
- sourceComponentId/@componentIdType
- sourceComponentId/@componentType
- sourceComponentId/@executionEnvironment
- sourceComponentId/@instanceId
- sourceComponentId/@location
- sourceComponentId/@locationType
- sourceComponentId/@processed
- sourceComponentId/@subComponent
- sourceComponentId/@threadId

XPath

No valid XPath for the shredder configuration file. A valid XPath requires a `values` declaration.

sourceComponentId/@application

Reference information about the sourceComponentId/@application element.

Description

Refer to the Common Base Event specification.

Values

xsd:string

Refer to same field in the ComponentIdentification in the Common Base Event specification. For example: WebSEAL is an application within the component IBM Security Access Manager for Web.

XPath

```
CommonBaseEvent/sourceComponentId/@application
```

sourceComponentId/@component

Reference information about the sourceComponentId/@component element.

Description

Product name, version, and fix pack level.

Values

String

For example, WebSEAL is an application within the component IBM Security Access Manager for Web, version 7.0, FixPack *x*.

Refer to same field in the ComponentIdentification in the Common Base Event specification.

XPath

```
CommonBaseEvent/sourceComponentId/@component
```

sourceComponentId/@componentIdType

Reference information about the sourceComponentId/@componentIdType element.

Description

Specifies the format and meaning of the component identified by this componentIdentification.

Values

xsd:string

Refer to same field in the ComponentIdentification in the Common Base Event specification.

XPath

```
CommonBaseEvent/sourceComponentId/@componentIdType
```

sourceComponentId/@componentType

Reference information about the sourceComponentId/@componentType element.

Description

A well-defined name that is used to characterize all instances of a given kind of component.

Values

xsd:string

Refer to same field in the ComponentType in the Common Base Event specification.

XPath

```
CommonBaseEvent/sourceComponentId/@componentType
```

sourceComponentId/@executionEnvironment

Reference information about the sourceComponentId/@executionEnvironment element.

Description

The immediate environment that an application is running in.

Values

xsd:string

Refer to same field in the ComponentIdentification in the Common Base Event specification.

XPath

```
CommonBaseEvent/sourceComponentId/@executionEnvironment
```

sourceComponentId/@instanceId

Reference information about the sourceComponentId/@instanceId element.

Description

Module instance information, for example, port number.

Values

String

Refer to same field in the ComponentIdentification in the Common Base Event specification.

XPath

```
CommonBaseEvent/sourceComponentId/@instanceId
```

sourceComponentId/@location

Reference information about the sourceComponentId/@location element.

Description

Physical location of the reporting component.

Values

xsd:string

Refer to same field in the ComponentIdentification in the Common Base Event specification.

XPath

```
CommonBaseEvent/sourceComponentId/@location
```

sourceComponentId/@locationType

Reference information about the sourceComponentId/@locationType element.

Description

Type of location.

Values

xsd:string

Refer to same field in the ComponentIdentification in the Common Base Event specification.

XPath

```
CommonBaseEvent/sourceComponentId/@locationType
```

sourceComponentId/@processId

Reference information about the sourceComponentId/@processId element.

Description

Process ID.

Values

String

Refer to same field in the ComponentIdentification in the Common Base Event specification.

XPath

```
CommonBaseEvent/sourceComponentId/@processId
```

sourceComponentId/@subComponent

Reference information about the sourceComponentId/@subComponent element.

Description

Module name.

Values

String

Refer to same field in the ComponentIdentification in the Common Base Event specification.

XPath

```
CommonBaseEvent/sourceComponentId/@subComponent
```

sourceComponentId/@threadId

Reference information about the sourceComponentId/@threadId element.

Description

Thread ID.

Values

String

Refer to same field in the ComponentIdentification in the Common Base Event specification.

XPath

```
CommonBaseEvent/sourceComponentId/@threadId
```

startTime

Reference information about the startTime element.

Description

Start time of the operation.

Values

xsd:DateTime

Refer to the creationTime field in the Common Base Event specification.

XPath

```
CommonBaseEvent/extendedDataElements[@name='startTime'][@type='dateTime']/values
```

suppressed

Reference information about the suppressed element.

Description

Identifies if the violation was suppressed.

Values

String

Use one of the following strings:

- `yes`
- `no`

XPath

```
CommonBaseEvent/extendedDataElements[@name='suppressed']/values
```

targetAccount

Reference information about the targetAccount element.

Description

Name of the user account.

Values

String

Any string allowed by targetResource.

XPath

```
CommonBaseEvent/extendedDataElements[@name='targetAccount']/values
```

targetInfoType

Reference information about the targetInfoType element.

Description

This element type represents information about the target of a management action, such as associating an access control list with a protected resource.

When displayed in output, this element uses the children of the targetInfoType element:
- targetInfoType.attributes
- targetInfoType.targetNames

XPath

No valid XPath for the shredder configuration file. A valid XPath requires a `values` declaration.

targetInfo.attributes

Reference information about the targetInfo.attributes element.

Description

Array of attributes for the values for the target.

targetInfo.targetNames

Reference information about the targetInfo.targetNames element.

Description

Object this operation is targeted against.

String

String allowed for the target object name by the application.

Examples:
- For group associate, target is a list of users added to a group.
- For ACL associate, target is a resource name associated with an ACL.
- For ACL disassociate, target is a resource name disassociated with the ACL.

XPath

```
CommonBaseEvent/extendedDataElements[@name='mgmtInfo']/children
[@name='targetInfo']/children
[@name='targetNames']/values[1]
```

Note: This XPath assumes that the targetInfo is part of mgmtInfo.

targetResource

Reference information about the targetResource element.

Description

Name of the resource on which the account exists.

Values

String

Any string allowed by the application.

XPath

```
CommonBaseEvent/extendedDataElements[@name='targetResource']/values
```

targetUser

Reference information about the targetUser element.

Description

Name of the user.

Values

String

Any string allowed by the application.

XPath

```
CommonBaseEvent/extendedDataElements[@name='targetUser']/values
```

targetUserInfo (1)

Reference information about the targetUserInfo element when used with the AUDIT_WORKFLOW event type.

Description

A container for information about the target users when used with the AUDIT_WORKFLOW event type.

This element uses the children of userInfoType:

- userInfo.appUserName
- userInfo.attributes
- userInfo.callerList
- userInfo.domain
- userInfo.location
- userInfo.locationType
- userInfo.realm
- userInfo.registryUserName
- userInfo.sessionId
- userInfo.uniqueId

XPath

No valid XPath for the shredder configuration file. A valid XPath requires a `values` declaration.

targetUserInfo (2)

Reference information about the targetUserInfo element when used with the AUDIT_MGMT_PROVISIONING event type.

Description

A container for information about the target users when used with the AUDIT_MGMT_PROVISIONING event type.

For AUDIT_MGMT_PROVISIONING events, `registryObjectInfo.type` must be `User`.

This element uses the children of registryObjectInfoType:
- registryObjectInfo.attributes
- registryObjectInfo.description
- registryObjectInfo.name
- registryObjectInfo.registryName
- registryObjectInfo.type

XPath

No valid XPath for the shredder configuration file. A valid XPath requires a `values` declaration.

targetUserRegistryInfo

Reference information about the targetUserRegistryInfo element.

Description

A container for information about the registry to which the target user belongs.

This element uses the children of the registryInfoType element:
- registryInfo.serverLocation
- registryInfo.serverLocationType
- registryInfo.serverPort
- registryInfo.type

XPath

No valid XPath for the shredder configuration file. A valid XPath requires a `values` declaration.

terminateReason

Reference information about the terminateReason element.

Description

The reason for the termination.

Values

String

The following strings are suggested values:

idleTimeout
: The session was terminated because it was inactive for too long.

sessionExpired
: The session was terminated because its maximum lifetime was exceeded.

sessionDisplaced
: The session was terminated because the session's user created a new session displacing this one.

sessionTerminatedByAdmin
: The session was terminated by an administrative action.

userLoggedOut
: The session was terminated at the user's request.

XPath

```
CommonBaseEvent/extendedDataElements[@name='terminateReason']/values
```

timestamp

Reference information about the timestamp element.

Description

End time of the operation.

Values

xsd:DateTime

If not specified, it is generated automatically. The timestamp is used in reports to determine when the audit event occurred. If the caller specifies the timestamp, it is the caller's responsibility to ensure that the timestamp provided is not spoofed.

Refer to the creationTime field in the Common Base Event specification.

XPath

```
CommonBaseEvent/@creationTime
```

type

Reference information about the type element.

Description

The type of command.

Values

String

The following strings suggested values:

config Configuration object.

server Object that represents an application server.

XPath

```
CommonBaseEvent/extendedDataElements[@name='type']/values
```

userInfo

Reference information about the userInfo element.

Description

The container for information about the user.

This element uses the children of userInfoType:

- userInfo.appUserName
- userInfo.attributes
- userInfo.callerList
- userInfo.domain
- userInfo.location
- userInfo.locationType
- userInfo.realm
- userInfo.registryUserName
- userInfo.sessionId
- userInfo.uniqueId

XPath

No valid XPath for the shredder configuration file. A valid XPath requires a `values` declaration.

userInfo.appUserName

Reference information about the userInfo.appUserName element.

Description

User's name within a given application.

Values

String

Any arbitrary string allowed by the application. For example, a Security Access Manager user name.

The following strings are suggested values:

unauthenticated
An unauthenticated user

XPath

```
CommonBaseEvent/extendedDataElements[@name='userInfo']/children
[@name='appUserName']/values
```

userInfo.attributes

Reference information about the userInfo.attributes element.

Description

Array of attributes in the user's credential.

Values

attributeType

Refer to “attributes” on page 296 for details.

XPath

The XPath is the first name element from an array of attributes elements.

```
CommonBaseEvent/extendedDataElements
[@name='userInfo']/children[10]/children
[@name='name']/values
```

Note: The index is `10`, for the `attributes` element must come after the `appUserName`, `callerList`, `domain`, `location`, `locationType`, `realm`, `registryUserName`, `sessionId`, and `uniqueId` elements

userInfo.callerList

Reference information about the userInfo.callerList element.

Description

A list of names representing the user's identities.

Values

String[]

Any arbitrary string allowed by the application can be used in the String[].

XPath

The XPath is the first callerList element from an array of callerList elements.

```
CommonBaseEvent/extendedDataElements[@name='userInfo']/children
[@name='callerList']/values[1]
```

userInfo.domain

Reference information about the userInfo.domain element.

Description

Domain in which user belongs.

Values

String

Any arbitrary string allowed by the application.

XPath

```
CommonBaseEvent/extendedDataElements[@name='userInfo']/children
[@name='domain']/values
```

userInfo.location

Reference information about the userInfo.location element.

Description

Location of the user. Example: In the case of WebSEAL, where the user authenticated from.

Values

xsd:string

Refer to the location field in the Common Base Event specification.

XPath

```
CommonBaseEvent/extendedDataElements[@name='userInfo']/children
[@name='location']/values
```

userInfo.locationType

Reference information about the userInfo.locationType element.

Description

Type of location.

Values

xsd:Name

Refer to the locationType field in the Common Base Event specification.

XPath

```
CommonBaseEvent/extendedDataElements[@name='userInfo']/children
[@name='locationType']/values
```

userInfo.realm

Reference information about the userInfo.realm element.

Description

The registry partition to which the user belongs.

Values

String

Any arbitrary string allowed by the application.

XPath

```
CommonBaseEvent/extendedDataElements[@name='userInfo']/children
[@name='realm']/values
```

userInfo.registryUserName

Reference information about the userInfo.registryUserName element.

Description

The registry partition to which the user belongs.

Values

String

Any arbitrary string allowed by the application.

Use “Not Available” when not available.

XPath

```
CommonBaseEvent/extendedDataElements[@name='userInfo']/children
[@name='registryUserName']/values
```

userInfo.sessionId

Reference information about the userInfo.sessionId element.

Description

ID for the user's session.

Values

Any arbitrary string

XPath

```
CommonBaseEvent/extendedDataElements[@name='userInfo']/children
[@name='sessionId']/values
```

userInfo.uniqueId

Reference information about the userInfo.uniqueId element.

Description

User's unique identifier.

Values

Integer UUID

A value of `-99999` means that a unique ID is not available.

For events generated by Security Access Manager, the unique ID is not available and is always set to `0`. When using the Session Management Server component of

Security Access Manager, the unique ID is always set to -99999.

XPath

```
CommonBaseEvent/extendedDataElements[@name='userInfo']/children
[@name='uniqueId']/values
```

userInputs

Reference information about the userInputs element.

Description

A container for information about the user inputs that are related to the work item. The inputs are collected as a list of attributes. For example, for approval and reject, one attribute could be the comment.

This element uses the children of the attributeType element:
- attributeType.name
- attributeType.source
- attributeType.value

XPath

No valid XPath for the shredder configuration file. A valid XPath requires a `values` declaration.

violationClassification

Reference information about the violationClassification element.

Description

Identifies the type of violation.

Values

String

The following strings suggested values:

account
Generic classification for policy violations related to an account, or attributes associated with an account, that does not fit in one of the specific account violation classifications.

accountDisallowed
Account was disallowed. Example: Guest accounts could be forbidden.

aclRestriction
The authorization settings on a protected resource violate the policy. Example: The ACL settings on the executables for a Web server might be improperly set.

antiVirus
The proper antivirus protection is not in place. Example: Version*x.y* of antivirus product ABC may be required, or the antivirus scan must be configured to run at least once per week.

audit The audit settings on a system may not comply with the policy. Example: The policy may require that all failed authentication attempts be audited. If audit settings do not comply, a violation is logged.

netConfig
Network configuration settings are not set as required by the policy. Example: The -s option must be specified when using the netlsd daemon in AIX.

password
The password policy is not being adhered to. Example: All passwords must be 8 characters or longer.

prohibitedService
Certain services might be prohibited. Example: Policy may require that TFTP never be active on a system.

softwareVersion
Policy may require that specific versions of software be installed. Example: A down-level version of Microsoft IIS or a version that requires a patch might be installed on a production server.

sysConfig
System configuration settings are not set as required by the policy. Example: Certain system log files may be required to exist.

XPath

```
CommonBaseEvent/extendedDataElements[@name='violationClassification']/values
```

violationDescription

Reference information about the violationDescription element.

Description

Predefined description of the particular violation.

Values

String

Any string allowed by the application.

XPath

```
CommonBaseEvent/extendedDataElements[@name='violationDescription']/values
```

violationName

Reference information about the violationName element.

Description

Name of specific policy violation. Example: “Win2K Guest Account Restriction”.

Values

String

Any string allowed by the application.

XPath

```
CommonBaseEvent/extendedDataElements[@name='violationName']/values
```

workItemInfo

Reference information about the workItemInfo element.

Description

An element type that represents information about a work item used in events related to workflow operations.

This container uses the children of workItemInfoType:
- workItemInfoType.id
- workItemInfoType.type

XPath

No valid XPath for the shredder configuration file. A valid XPath requires a `values` declaration.

workItemInfoType.id

Reference information about the workItemInfoType.id element.

Description

Unique identifier of the work item.

Values

String

XPath

```
CommonBaseEvent/extendedDataElements[@name='workItemInfoType']/children
[@name='id']/values
```

workItemInfoType.type

Reference information about the workItemInfoType.type element.

Description

Type of the work item.

Values

String

The following strings are suggested values:

approval
This type of work item allows a user to either approve or reject a specific request.

requestForInfo
This type of work item allows a user to provide additional information for a specific request.

workOrder
This type of work item is used to request manual operations for the user. For example, a work order to manually create a specific account on a resource.

XPath

```
CommonBaseEvent/extendedDataElements[@name='workItemInfoType']/children
[@name='type']/values
```

Part 7. Troubleshooting

Chapter 25. Problem determination

This topic provides information for troubleshooting the installation of the Common Auditing Service components.

The topic provides the following information:

- Log files
- Installation problems
- Configuration problems
- Upgrade problems
- Uninstallation problems
- Staging utility problems
- Trace level for XML data store
- Debug trace log

Error messages and descriptions are in the *IBM Security Access Manager for Web Error Message Reference*.

Log files

Log files are produced for the following Common Auditing Service components:

- ISMP installer of Common Auditing Service audit server
- Common Auditing Service configuration utility
- Common Auditing Service audit server
- Common Auditing Service C client
- Common Auditing Service WebService emitter
- Common Auditing Service server utilities

Installation log files

The installation and uninstallation procedures generate a set of log files. These files are available in the locations specified in the table.

Table 66. Installation log files

Type	Default message log location
Server installation	**Windows**: • *CARS_HOME*\serverInstall.log • *CARS_HOME*\server\logs\sharedLibCreation.log (only for console feature) **AIX, Linux, or Solaris**: • *CARS_HOME*/serverInstall.log • *CARS_HOME*/server/logs/sharedLibCreation.log (only for console feature)
Server uninstallation	**Windows**: *CARS_HOME*\serverUninstall.log **AIX, Linux, or Solaris**: *CARS_HOME*/serverUninstall.log
Note: *CARS_HOME* is the installation directory of Common Auditing Service.	

Runtime log files

The Common Auditing Service produces runtime log files. The files are described in this section.

C client configuration

The location of the C client runtime log file is specified by the errorFilePath parameter in the `[cars-client]` stanza. See the configuration stanzas appendix for information about the errorFilePath parameter.

Server utilities log files

The Common Auditing Service server utilities produce log files.

Most errors in the server utilities are handled by generating exceptions. When an error occurs, it is logged in to message and trace logs. In addition, the error message is reported on the console (standard error). Trace and message log file locations and filtering are controlled by properties in the ibmcars.properties file. See "The ibmcars.properties file" on page 69 for more information.

When run as a postarchive operation, the XmlStoreUtility captures the output from DB2 commands in the XmlStoreUtilitsScript_<yyyymmdd>_<hhmmss>.log file. Review the contents of this file if XmlStoreUtility fails during a postarchive operation.

WebSphere Application Server log files

While running, Common Auditing Service components create entries in the WebSphere Application Server log files.

The following activities are logged in to the WebSphere Application Server logs:

- Installation of Common Auditing Service features (Common Auditing Service Server and Common Auditing Service Configuration Console)
- Configuration of Common Auditing Service components that use the configuration utility
- Common Auditing Service WebService emitter
- Security event factory utilities
- Common Auditing Service Web service (Web module)
- Common Auditing Service XML data store (EJB module)

By default, the WebSphere Application Server logs for a stand-alone server are at:

- **AIX, Linux, or Solaris**: `/opt/IBM/WebSphere/AppServer/profiles/`*`profile`*`/logs/`*`servername`*
- **Windows**: `C:\Program Files\IBM\WebSphere\AppServer\profiles\`*`profile`*`\logs\`*`servername`*

By default, the WebSphere Application Server logs on a Deployment Manager node are at:

- **AIX, Linux, or Solaris**: `/opt/IBM/WebSphere/AppServer/profiles/Dmgr01/logs/`*`dmgr`*
- **Windows**: `C:\Program Files\IBM\WebSphere\AppServer\profiles\Dmgr01\logs\`*`dmgr`*

Considerations for setting the trace file path, trace level, and error file path during problem determination

To help determine the source of errors, consider the use of the traceFilePath, traceLevel, and errorFilePath entries, which are specified in the [cars-client] stanza.

Purpose

When you troubleshoot the source of errors, consider the use of the traceFilePath and traceLevel values. Setting the trace level to the value 3 (traceLevel=3) writes events that result from error conditions and from all trace points in the code to the log. Output is written to the file specified by the traceFilePath parameter. The output includes the properties that are defined in this configuration file, and the values that are sent to the Common Audit Service audit server.

The errorFilePath entry specifies the name and location of the error log file to be used by the server or application.

Note: Tracing does not work if properties or values are specified incorrectly in the `[cars-client]` stanza. The names of the error file and trace log file must be unique between multiple instances of servers on a system. If more than one application or instance is configured to use the same file name, errors will result. To ensure uniqueness, specifyit is recommended that errorFilePath and traceFilePath specify the azn-server-name of the server.

Installation problems

This topic describes some installation problems that you might encounter.

Installer displays an error although the required DB2 software is installed

This topic describes the problem and workaround you can use if the Common Auditing Service installer does not detect correctly the DB2 software version.

Problem

During installation, the ISMP installer of the Common Auditing Service audit server component might display the following message because it does not accurately detect the installed version of DB2:

```
CBAIN0120E Prerequisite detection has not found an installation of IBM DB2.
The feature selected for installation requires either the IBM DB2 Server or the IBM
DB2 Client to operate. The version allowed is Version 9.7 and higher.
You must install an allowable version of the IBM DB2 product
either now or before attempting to use the selected product feature.
```

Workaround

Run the **db2level** command on the Common Auditing Service audit server host to verify that the required version of DB2 is installed. If the database server is remote to the audit server, run the command on the database server. If the database server and client are at the correct level, continue with the installation.

Example

Here is sample output from the db2level command:

```
DB21085I  Instance "ldapdb2" uses "32" bits and DB2 code release "SQL09012"
with level identifier "01030107".
Informational tokens are "DB2 v9.1.0.2", "s070210", "MI00183", and FixPack "2".
Product is installed at "/opt/IBM/db2/V9.1".
```

Silent installation does not fail when missing prerequisites

This topic describes the problem and workaround you can use when a silent installation does not fail, even though the prerequisites are not met.

Problem

During an interactive installation, the Common Auditing Service audit server installation checks for the existence of the appropriate versions of software prerequisites. You are notified if the appropriate versions of the prerequisites are not available on the machine where the audit server is being installed.

However, during a silent installation, the audit server installation continues even when the software prerequisites are not present, or are not at the required levels. Therefore, if you encounter a problem after you run the silent installation, the reason might be that the prerequisite products are not installed, or you did not complete the preinstallation checklist items.

Workaround

Ensure that the appropriate prerequisites are on the machine before you complete a silent installation. See “Pre-configuration checklist for all platforms” on page 37 for more information about setup before installation.

Installation does not continue when the target WebSphere Application Server is stopped

This topic describes the problem and workaround you can use when an installation does not continue because WebSphere Application Server is stopped.

Problem

During installation, Common Auditing Service checks if the target WebSphere Application Server is running. If WebSphere Application Server is stopped, you are notified with an error message that states that a connection was not made with the Deployment Manager or the stand-alone server in this profile.

Workaround

Ensure that the server of the specified WebSphere Application Server profile is running.

To check the status of the server, use the server status command that is in the WAS_profile/bin directory:

serverStatus.[bat | sh] *server_name*

If the server is stopped, issue the start server command that is in the WAS_profile/bin directory:

startServer.[bat | sh] *server_name*

Use the serverStatus command again to check the server status after you issue the startServer command.

Example

To check the status on a Windows system use:

```
cd D:\IBM\WebSphere\AppServer\profiles\AppSrv01\bin\
serverStatus.bat server1
```

Example

To start the server on a Windows system use:

```
cd D:\IBM\WebSphere\AppServer\profiles\AppSrv01\bin\
startServer.bat server1
```

Installation does not continue when JVM version 1.5 or 1.6 is not found

This topic describes the problem and workaround you can use if JVM version 1.5 or 1.6 is not located during the installation of Common Auditing Service.

Problem

During the installation of Common Auditing Service , if the installer cannot find JVM version 1.5 or above, you are notified with an error message that states that a suitable JVM was not found.

Workaround

Rerun the installation program and specify the following option:

`-is:javahome` *path_to_JVM1.6_home*

Example

The following example runs the Common Auditing Service installer and specifies the Java home path:

```
install_cars_audit_srv_win64.exe -is:javahome D:\IBM\Java
```

Installation displays an error when WebSphere Application Server software is not found

This topic describes the problem and workaround you can use if the required WebSphere Application Server information is not found during the installation of Common Auditing Service.

Problem

During Installation, if a wrong WebSphere Application Server profile path is passed, or if the prerequisite WebSphere Application Server software is not found, the installer returns an error:

```
Please enter a valid WebSphere profile path
```

Workaround

Specify a valid WebSphere Application Server profile path.

Common Audit Service configuration problems

This topic describes workarounds for some Common Audit Service configuration problems you might encounter.

Text displays incorrectly in some configuration panels

During configuration, text is displayed in reverse order in some windows, or when you use a bidirectional locale, such as Hebrew, the text displays incorrectly.

Problem

During the configuration of Common Auditing Service, some windows contain characters that are displayed in reverse order. For example, in the Common Auditing Service Status window, the Host and SOAP connector port are displayed as 0880:tsohlacol instead of localhost:0880. Also, if the browser is set to use a bidirectional locale, such as Hebrew, the text displays incorrectly.

Workaround

To correct this problem, specify "client.encoding.override=UTF-8" as a generic JVM argument in the Java Virtual Machine configuration window of the administrative console. After you set the encoding to UTF-8, the English text will display correctly left-to-right, and the Hebrew text will display correctly.

Use the following steps to set the use of UTF-8 character encoding:

1. In the administrative console, click **Servers-> Application servers** and select the server that you want to enable for UCS Transformation Format (UTF-8).
2. Under Server Infrastructure, click **Java and Process Management -> Process Definition- > Java Virtual Machine**.
3. Specify `-Dclient.encoding.override=UTF-8` for **Generic JVM Arguments** and click **OK**. When this argument is specified, UTF-8 character encoding is used instead of the character encoding that would be used if the autoRequestEncoding option was in effect.

SOAP connection fails when a Common Audit Service Configuration Console is deployed in an eWAS environment

This topic describes a SOAP connection problem that can occur when a Common Audit Service Configuration Console is deployed in an eWAS environment.

Problem

The eWAS console log displays the following error:

```
The Common Audit Service Console failed to connect to the
specified WebSphere Application Server process
The error is ADMC0009E: The system failed to make the SOAP RPC call: invoke
```

In the instance of WebSphere Application Server where the Common Audit Service audit server is deployed, the SystemOut.log displays the following error:

```
Caused by: [SOAPException: faultCode=SOAP-ENV:Client;
msg=com.ibm.cars.config.globalUtil.DeploymentObjectHandle
Server stack trace
JMXTransformException java.lang.ClassNotFoundException:
com.ibm.cars.config.globalUtil.DeploymentObjectHandle
```

Workaround

Restart the WebSphere Application Server where the audit server is deployed, and restart the eWAS where the Common Audit Service Configuration Console is deployed.

Problem deploying the Java stored procedure on a Linux platform

You might receive the following error message when you install the audit server:

```
An error occurred while installing the ibmcarsdd.jar JAR file.
```

The error message occurs while you deploy the Java stored procedure on a Linux platform. This cause of this error is that the appropriate symbolic links are not created in the `/usr/lib` directory.

Follow the instructions in "Setting up to run the Java stored procedures on Linux" on page 54.

C client cannot communicate with the Common Auditing Service server

The C client cannot communicate with the Common Auditing Service audit server when incorrectly configured.

Problem

The C client cannot communicate with the Common Auditing Service audit server when incorrectly configured.

Workaround

Correct mistakes in the [cars-client] stanza in the `pdaudit` configuration file, then restart the application. Correcting the configuration to enable auditing includes the following settings:

- Set the **`doAudit`** property to the value `yes`.
- Set the **`serverURL`** property to the correct value. To verify that the value is correct, specify the same value in the URL field of your browser to ensure that it resolves. For example, a URL value for a non-SSL server is similar to:

  `http://`*`hostname`*`:`*`WC_defaulthost_port_number`*`/CommonAuditService/services/Emitter`

  A URL value for an SSL-enabled server is similar to:

  `https://`*`hostname`*`:`*`WC_defaulthost_secure_port_number`*`/CommonAuditService/services/Emitter`

  A correct URL value results in the web browser displaying a page with contents that are similar to:

  ```
  {urn:ibm:cars:10}Emitter

  Hi there, this is a Web service!
  ```
- Set the **`diskCachePath`** property to a valid value if the **`useDiskPath`** property is set to `always` or to `auto`; `auto` is the default value. Both values enable caching to a cache file.

  A valid value for **`diskCachePath`** is a file path that exists and includes a valid cache file name.

Common Audit Service upgrade problems

Upgrading Common Audit Service from earlier versions to version 7.0 fails for various reasons

The upgrade of a lower-versioned audit database to the version 7.0 audit database might fail for the following reasons:

- Target DB2 server instance that is hosting the existing lower-versioned audit database was not started.
- Wrong credentials were specified for the DB2 instance owner in the Audit database window of the Common Audit Service Configuration Console during the upgrade.
- Target lower-versioned database became corrupted and is not a valid XML data store database.
- Remote DB2 server node that is hosting an existing lower-versioned audit database was not cataloged in the local DB2 client before the upgrade was started.
- Existing lower-versioned audit database that is present on the remote DB2 server node was not cataloged in the local DB2 client before the upgrade was started.

Use the above reasons for failure as a checklist to help prevent and resolve problems with the upgrading of the audit database for use with Common Audit Service Version 7.0.

Common Audit Service uninstallation problems

This topic describes uninstallation problems that you might encounter, and provides workarounds to help resolve the problems.

Uninstall.bin not available

When you try to use the installation wizard to uninstall the server on a 64-bit AMD machine, you might find that the file `/opt/IBM/Tivoli/CommonAuditService/_uninst/uninstall.bin` does not exist.

Instead of using the installation wizard, you can use the `uninstall.jar` file. To use the `uninstall.jar` file to uninstall, run the following command:

```
java -cp uninstall.jar run
```

The `uninstall.jar` file is in the following directory on AIX, Linux, and Solaris platforms: `/opt/IBM/Tivoli/CommonAuditService/_uninst/`.

CarsConfigUtil.jar is not removed during a successful uninstallation of Common Audit Service

Problem

After uninstalling Common Audit Service, the `CarsConfigUtility.jar` file is not removed from the *CARS_HOME*`/config/lib` folder.

Workaround

Remove the `CarsConfigUtility.jar` file manually from the *CARS_HOME*`/config/lib` folder, then restart the WebSphere Application Server.

Failed uninstallation workarounds

This topic describes the workarounds that are available to manually remove the audit server from the Deployment Manager or the managed nodes after a failed uninstallation.

Manually removing the audit server configuration components after a failed uninstallation

If an uninstallation of the audit server configuration components fails, use this procedure to clean up the system.

About this task

The uninstaller of the audit server can leave behind following entries in the target WebSphere Application Server or Network Deployment Manager in the event of an uninstallation failure:

- Common Audit Service Configuration Utility
- Common Audit Service Configuration Console
- Extension MBean Provider for Configuration Utility
- Shared Library for Configuration Console
- WebSphere Application Server *CARS_HOME* variable

Perform following steps if a server uninstallation fails:

1. Uninstall the Common Audit Service Configuration Utility, if this feature is not removed during a failed uninstallation:
   - For a stand-alone single server installation of Common Audit Service:
     a. In the target stand-alone WebSphere Application Server administrative console, select **Applications** > **Application Types** > **WebSphere enterprise applications**.
     b. Select the **CommonAuditServiceConfiguration** application, and click **Uninstall** to uninstall the Common Audit Service Configuration Utility from the target stand-alone WebSphere Application Server.
   - For a Network Deployment setup of Common Audit Service, uninstall the CommonAuditServiceConfiguration application from the target Deployment Manager by running the following command on the **wsadmin** command line of the Deployment Manager:
     ```
     wsadmin>$AdminApp uninstall CommonAuditServiceConfiguration
     ```
2. Uninstall the Common Audit Service Configuration Console, if this feature is not removed during a failed uninstallation. Run following command from the **wsadmin** command line of the target stand-alone single server or the Deployment Manager:
   ```
   $AdminApp update isclite modulefile { -operation delete  -contenturi CARS7.0.war}
   ```
3. Remove the Extension MBean Provider for the Configuration Utility if this component is not removed during a failed uninstallation:
   - For a stand-alone single server installation of Common Audit Service:
     a. In the target stand-alone WebSphere Application Server administrative console, select **Application servers** > **server1** > **Administration services** > **Extension MBean Providers**.
     b. Select **CarsConfigUtilProvider** and click **Delete**.
   - For a Network Deployment installation of Common Audit Service:

   a. In the deployment manager WebSphere Application Server administrative console, select **System Administration-> Deployment Manager-> Administration Services-> Extension MBean Providers**.
   b. Select **CarsConfigUtilProvider** and click **Delete**.
4. Remove the Shared Library for the Configuration Console if this library is not removed during a failed uninstallation:
   a. In the WebSphere Application Server administrative console, select **Environment-> Shared Libraries**.
   b. Select **All scopes** in the scope settings.
   c. Select **CarsConfigUtilSharedLib** and click **Delete**.
5. Remove the WebSphere CARS_HOME variable.
   a. Select **Environment**-> **WebSphere variables**.
   b. Select **All scopes** in the scope settings.
   c. Select **CARS_HOME** and click Delete.
6. Remove any directories and files from the Deployment Manager system (node), as well as any managed node systems (nodes), that are left behind by the failed uninstallation of the server. These might include:
   - *CARS_HOME* folder, for example, D:\IBM\Tivoli\CommonAuditService on Windows, or /opt/IBM/Tivoli/CommonAuditService on Linux or UNIX.
   - Log files under *WAS_HOME*\logs or *WAS_PROFILE_PATH*\logs. Also, on Linux or UNIX you might find logs in the /tmp folder.
7. Remove the _uninst folder from the /tmp directory on Linux and UNIX, or the %TEMP% directory on Windows.
8. Start the Deployment Manager.

Manually removing the audit server components after a failed uninstallation

You might need to manually remove audit server components if a server uninstallation fails and Common Audit Service components are not fully unconfigured.

About this task

The uninstaller of Common Audit Service checks whether components are not unconfigured before starting the uninstallation. In this situation, a warning message is displayed and prompts you to fully unconfigure the Common Audit Service components before you continue.

If you ignore the warning message and continue with the uninstallation, the configuration components might not uninstall, and you forfeit the ability to unconfigure Common Audit Service components with the configuration console. In this case, you must manually remove the Common Audit Service configuration components from the target WebSphere Application Server and DB2 UDB server before you install and configure Common Audit Service again.

Perform the following steps if a server uninstallation fails and Common Audit Service components are not fully unconfigured:

Procedure

1. Remove the deployed applications from WebSphere Application Server:
   a. Open the WebSphere Application Server administrative console of the Deployment Manager.

b. Manually uninstall the Common Audit Service Web Service (CommonAuditService) in the WebSphere Application Server administrative console of the network Deployment Manager. To uninstall enterprise applications from a cluster, click **Applications-> Enterprise Applications** in the WebSphere Application Server administrative console.

c. To uninstall an application from the cluster, select the application that you want to uninstall and click **Uninstall**.

2. Remove the JDBC resources from WebSphere Application Server.

   Ensure that the JDBC providers and WebSphere Application Server data sources for the Common Audit Service applications were removed during the uninstallation from the cluster and server scopes. If these resources were not removed during the uninstallation, remove them manually using the WebSphere Application Server administrative console of the network Deployment Manager.

   To remove the JDBC resources, click **Resources-> JDBC Providers**. For a clustered configuration of Common Audit Service, delete the entry for Event_DB2_XML_JDBC_Provider from the cluster. For a stand-alone single server configuration, delete the Event_DB2_XML_JDBC_Provider entry from Server Scopes. This will automatically remove data sources for the JDBC provider.

   To remove the J2C authentication entries for the data source:

   a. Click **JDBC Providers**, then click any of the existing JDBC providers.

   b. Click **Data sources**, then click the existing data source.

   c. Click **J2EE Connector Architecture (J2C) authentication data entries**. Here you might see residual entries of the J2C EventAuthDataAliasDB2Xml authentication alias. Delete this entry.

   Note: When you remove JDBC resources from the cluster scope, before you save the changes, ensure that you check **Synchronize changes with nodes** to synchronize the changes with other managed nodes.

3. Log off the WebSphere Application Server administrative console of the Network Deployment manager.

4. Remove the databases.

   If the XML data store database (normally eventxml) was not removed during the failed uninstallation, manually drop the database from the DB2 server. Open the DB2 command prompt from network Deployment Manager by using one of the following commands:

   - For Windows systems:

     ```
     db2cmd db2
     ```

   - For Linux or UNIX systems:

     ```
     db2
     ```

     If the DB2 server is local to the Deployment Manager system, run the following commands at the DB2 command prompt to remove the remaining databases:

     ```
     list database directory
     drop db database_name
     ```

     If the DB2 server is remote to the Deployment Manager system, run the following commands at the DB2 command prompt to remove the remaining databases:

```
list database directory
attach to node_name user db_user using password
drop db database_name
detach
```

where *database_name* is the name of the database that is listed with the list command.

5. Ensure that you remove entries of the databases from the DB2 administration clients on the Deployment Manager, and from all managed nodes. Remove entries of the database using the DB2 Control Center that is on these nodes.
6. Stop the cluster using the WebSphere Application Server administrative console, then stop the Deployment Manager.
7. Restart the cluster to start all managed node server instances. If any of the managed node server instances do not start, in the WebSphere Application Server administrative console of the network Deployment Manager:
   a. Click **System administration-> Nodes**.
   b. Select the node on which the server instance did not start.
   c. Click **Full Resynchronize** on the top menu.

Results

After you complete these steps, your system is ready for another Common Audit Service audit server installation. If residual entries still exist in the installation registry, the Common Audit Service audit server installation will fail to install. However, during the installation attempt, the registry entries will be removed when the rollback is completed.

Web service and emitter problems

This topic lists Web service and emitter problems that you might encounter.

Disregard message 0000004a

The following message that is generated by the Web service and emitter can be ignored:

- Message 0000004a

```
0000004a WSDDJAXRPCHan W
com.ibm.ws.webservices.engine.deployment.wsdd.WSDDJAXRPCHandlerInfoChain
 getHandlerChain WSWS3389E:
Error: JAXRPC Handler Class com.ibm.cars.webservice.DebugHandler not
found/loaded, ignored.
   java.lang.ClassNotFoundException: com.ibm.cars.webservice.DebugHandler
        at java.net.URLClassLoader.findClass(URLClassLoader.java
          (Compiled Code))
        at com.ibm.ws.bootstrap.ExtClassLoader.findClass
          (ExtClassLoader.java:106)
```

Web service emitter log displays event data

This topic describes the problem of the Web service emitter log containing audit data.

Problem

When the Web service emitter receives a server error, the event content is printed into the log file of the application. The audit data might contain sensitive information.

Workaround

The Web service emitter log might contain sensitive data. Access to the log file of the application must be protected.

Server utility problems

This topic lists some server utility problems you might encounter.

Exception occurs while the staging utility runs

While running the staging utility, the `javax.xml.transform.TransformerConfigurationException` might be thrown. This exception means that a syntactically incorrect XPath statement was passed to the staging utility.

Verify that the XPath statements in the CARSShredder.conf file to select attributes of events is syntactically correct. See the XPath documentation for detailed information about the correct XPath statement event attributes.

java.lang.NullPointer exception occurs while running the staging utility

This topic describes the problem of an error that occurs while using the staging utility.

Problem

While running the staging utility, a null pointer exception might be thrown. This failure can be caused by an error in the configuration file, an error that is caused by database failures, or network failures. Typically, database failures are caused by the transaction log filling up, free disk space that is running out, or when the DB2 server stops running.

Workaround

Verify that the error is repeatable by running the staging utility again. If the failure was caused by a temporary environment condition, such as a network failure, the staging utility runs to conclusion. If the error occurs again, rerun the staging utility with a batchsize set to 1. This causes the staging utility to print any underlying exception.

Identify the main exception. If the exception contains TransformerConfigurationException, the failure is caused by incorrect entries in the CARSShredder.conf file. In this case, examine the recent modifications to CARSShredder.conf, and correct any errors including mismatched quotations. The following example is an incorrect entry in CARSShredder.conf because the order of single and double quotation marks is inconsistent.

```
cars_t_event, eventType,     ' "AUDIT_AUTHN_CREDS_MODIFY' "
```

If the exception is SQLException, the failure might be caused by a database error or a staging error. See the staging utility error log to identify the SQL exception. Errors are frequently caused by a CARShredder.conf file that refers to a nonexisting table column, or by including multiple references to an existing target column, as shown in the following example:

```
cars_t_event, src_comp,     #sourceComponentId.component#
cars_t_event, src_comp,     #sourceComponentId.subComponent#
```

The following error log output identifies the cause of the error in the previous example of double references:

```
2006.09.11 19:21:55.730 ----- PROGRAM ERROR null null com.ibm.cars.staging.DBTable
update Thread-0 CBASU0125E A
database error occurred for the following SQL statement: INSERT into CARS_T_EVENT
(EVENT_ID,CARS_SEQ_NUMBER,EVENTTYPE,SRC_COMP,USR_SESSION_ID,SRC_SUB_COMP,
SRC_LOCATION,TIME_STAMP,OUTCOME_RESULT,S
TART_TIME,SRC_COMP,SRC_COMP,OUTCOME_FAIL_RSN,SRC_INSTANCE_ID,SRC_LOCATION,
USR_DOMAIN,USR_LOC_TYPE,APP_USR_NAME,EN
D_TIME,USR_LOC) VALUES(?,?,'AUDIT_AUTHN_CREDS_MODIFY',?,?,?,?,?,?,?,?,?,?,?,?,?,?)
The error occurred
during data insertion for: Table CARS_T_EVENT, column USR_LOC. Database exception:
The column "SRC_COMP" is
specified more than once in the INSERT, UPDATE or SET transition-variable statement.
CBASU0125E A database error occurred for the following SQL statement: INSERT
into CARS_T_EVENT
(EVENT_ID,CARS_SEQ_NUMBER,EVENTTYPE,SRC_COMP,USR_SESSION_ID,SRC_SUB_COMP,
SRC_LOCATION,TIME_STAMP,OUTCOME_RESULT,S
TART_TIME,SRC_COMP,SRC_COMP,OUTCOME_FAIL_RSN,SRC_INSTANCE_ID,SRC_LOCATION,
USR_DOMAIN,USR_LOC_TYPE,APP_USR_NAME,EN
D_TIME,USR_LOC) VALUES(?,?,'AUDIT_AUTHN_CREDS_MODIFY',?,?,?,?,?,?,?,?,?,?,?,?,?,?)
The error occurred
during data insertion for: Table CARS_T_EVENT, column USR_LOC. Database exception:
The column "SRC_COMP" is
specified more than once in the INSERT, UPDATE or SET transition-variable statement.
  com.ibm.db2.jcc.c.SqlException: The column "SRC_COMP" is specified more than once
in the INSERT, UPDATE or SET transition-variable statement.
```

If you cannot locate the source of error and the problem persists, consult with your database administrator.

Remote database access failure occurs when using staging utility or XML data store utilities

After you configure the Common Auditing Service server on a WebSphere Application Server Network Deployment cluster configuration, when the DB2 server is remote, the ibmcars.properties file might not be configured with the host name of the remote DB2 server.

Problem

If the remote DB2 host name is not configured, the server utility programs might not be able to connect to the remote database server, and the following errors would be encountered.

When running the XML store utilities:

```
CBAXU0216E An error occurred while establishing database connection.
The message returned by the database driver is: java.net.ConnectException :
Error opening socket to server localhost on port 50000 with message :
Connection refused
URL used for db connection is jdbc:db2://localhost:50000/eventxml.
```

When running the staging utility:

```
CBASU0101E Cannot connect to the database: java.net.ConnectException :
Error opening socket to server localhost on port 50000 with message :
Connection refused.
```

```
Wrapped Exception:
com.ibm.db2.jcc.c.DisconnectException: java.net.ConnectException :
Error opening socket to server localhost on port 50000 with message :
Connection refused
```

Workaround

Set the following property in the *CARS_HOME*/server/etc/ibmcars.properties file:

```
util.db.hostname=db2_server_hostname
```

WebSphere Application Server problems

This topic lists WebSphere Application Server problems that you might encounter.

Out of memory error

If millions of events are sent to the server, you might encounter a WebSphere out-of-memory error. This error might be caused by a stack overflow problem. For information about how to resolve the stack overflow problem, see the IBM Redbook *WebSphere Application Server: Application Server Crash Problem Determination*.

Debug tracing of installation or uninstallation of Common Audit Service

Run a debug trace with the -Dis.debug flag during installation and uninstallation to provide more information if there is a problem.

Purpose

Using the -Dis.debug flag causes installation program to display a detailed message about the installation process. This might indicate a problem with the installation program product itself or with the Common Auditing Service. This is a valuable tool in debugging problems you might encounter during silent installation. Start the installer with the following syntax as part of your command.

-Dis.debug=1 > *logging_file_directory*

Parameters

logging_file_directory
Specifies the file location on the target machine where the debug trace is recorded.

Sample

To use the debug parameter in a silent server installation and send the debug information to the file debug.txt, enter:

```
java -Dis.debug=1 -cp install_cars_srv.jar run -silent -options response_file >
  debug.txt
```

Notes

When you use the debug flag during installation, the XML database passwords are visible in the log file.

Part 8. Appendixes

Appendix A. Routing files

Routing files are ASCII files that you can use to customize the logging events for C language-based servers, daemons, and other C-language programs and applications. You can use the contents of routing files to control aspects of event logging, such as:

- Whether to enable logging for specific event classes
- Where to direct the output for each event class
- How many log files to use for each event class
- How large each log file can be for each event class

Locations of routing files

Table 67 lists the default locations for the routing files. The routing and routing.template files are in the same default directory. The routing files control the logging of events.

Table 67. Default locations of routing files

Component	Default name and location of routing file

Runtime environment	**Windows**
	`%PD_HOME%\etc\routing`
	AIX, Linux, and Solaris
	`/opt/PolicyDirector/etc/routing`

Policy server	**Windows**
	`%PD_HOME%\etc\pdmgrd_routing`
	AIX, Linux, and Solaris
	`/opt/PolicyDirector/etc/pdmgrd_routing`

Authorization server	**Windows**
	`%PD_HOME%\etc\pdacld_routing`
	AIX, Linux, and Solaris
	`/opt/PolicyDirector/etc/pdacld_routing`
	Note: `pdacld_routing` applies to the default authorization server. If there are multiple instances of the authorization server, the routing file name is prefixed with the given instance name, such as, *`instance1`*`-pdacld_routing`.

Policy proxy server	**Windows**
	`%PD_HOME%\etc\pdmgrproxyd_routing`
	AIX, Linux, and Solaris
	`/opt/PolicyDirector/etc/pdmgrproxyd_routing`

WebSEAL server	**Windows**
	`%PD_WEB%\etc\routing`
	AIX, Linux, and Solaris
	`/opt/pdweb/etc/routing`

Note:

- For WebSEAL, the `routing` file is created from the `routing.template` file during installation. The `routing` and `routing.template` file are in the same directory.
- The Plug-in for Web Servers component programmatically sets the information that is typically contained in a routing file. Therefore, Plug-in for Web Servers has no routing file of its own.

If you do not want to modify the default routing file (`/etc/routing`), you can use the PD_SVC_ROUTING_FILE environment variable to define an alternative routing file. If the file defined by this environment variable does not exist or is not accessible, the default routing file (`/etc/routing`) is used.

Routing file entries

Each routing file contains entries that control the logging of events. Use the following format (entered on a single line without spaces) when you define entries in routing files:

```
component:subcomponent.level[[,subcomponent.level]...]
:destination:location [[;destination:location]...] [;GOESTO:{other_severity
| other_component}]
```

Where:

component:*subcomponent* **[[,***subcomponent* **]...]**

Specifies the component, subcomponents, and reporting levels of events to log.

For the component portion, you can specify an asterisk (*) to log data for all components.

For the subcomponent portion, you can specify an asterisk (*) to log data for all subcomponents of the specified component.

destination

Specifies where to log the events. For each destination, you must specify a location. When you specify multiple destination-location pairs, separate each pair with a semicolon (;). The following destinations are valid:

DISCARD

Discards the events.

FILE Writes the events as ASCII text in the current code page and locale to the specified location.

When you use this destination, you must specify a location for the file. Optionally, you can follow the `FILE` destination by a period and two numbers that are separated by a period (for example, `FILE.10.100`).

The first value indicates the number of files to use. The second value indicates the number of events each file can contain. If you do not specify these values, there is only 1 log file that grows without limit.

The average size of an ASCII event is 200 bytes. Because the maximum size of a log file is 2 GB, the maximum number of events must be limited to approximately 10,000,000 events.

STDERR

Writes the events as ASCII text in the current code page and locale to the standard error device.

STDOUT

Writes the events as ASCII text in the current code page and locale to the standard output device.

TEXTFILE

Same a `FILE`.

UTF8FILE

Writes the events as UTF-8 text to the specified location.

When you use this destination, you must specify a location for the file. Optionally, you can follow the `UTF8FILE` destination by a period and two numbers that are separated by a period (for example, `UTF8FILE.10.100`).

The first value indicates the number of files to use. The second value indicates the number of events each file can contain. If you do not specify these values, there is only 1 log file that grows without limit.

The average size of a UTF-8 event is 200 bytes. Because the maximum size of a log file is 2 GB, the maximum number of events must be limited to approximately 10,000,000 events.

Note: When the operating system does not use a UTF-8 code page, the conversion to UTF-8 can result in data loss. When data loss occurs, the log file contains a series of question mark (?) characters at the location where the data conversion was problematic.

XMLFILE

Writes events to the specified location in the XML log format.

When you use this destination, you must specify a location for the file. Optionally, you can follow the `XMLFILE` destination by a period and two numbers that are separated by a period (for example, `XMLFILE.10.100`). The first value indicates the number of files to use. The second value indicates the number of events each file can contain.

If you do not specify these values, there is only 1 log file that grows without limit.

The maximum size of a log file is 2 GB.

XMLSTDERR

Writes events to the standard error device in the XML log format.

XMLSTDOUT

Writes events to the standard output device in the XML log format.

GOESTO:{*other_severity* | *other_component*}**]**

Specifies that events must additionally be routed to the same destination and location as events of the specified component.

location

Specifies the name and location of the log file. When the destination is `TEXT`, `TEXTFILE`, `UTF8FILE` or `XMLFILE`, you must specify a location. When the destination is `DISCARD`, `STDERR`, `STDOUT`, `XMLSTDERR`, or `XMLSTDOUT`, you must specify a hyphen (`-`).

When you specify a fully qualified file name, you can use the `%ld` character string to insert the process ID into the file name.

When the number of files is specified as part of the destination, a period and the file number are appended to the specified log file.

Note: On Windows operating systems, the file name must not end with a period. If the file name ends with a period, when the file number is appended, the file name contains two consecutive periods. File names with two consecutive periods are not valid.

On AIX, Linux, and Solaris operating systems, the file name must be followed by:

- File permissions.
- The user who owns the file.
- The group that owns the file.

Use the following format:

```
location:permissions:owner:group
```

Appendix B. Configuration stanzas

This appendix describes the guidelines for changing the following files:

- Configuration files.
- The location of the configuration files.
- The contents of the configuration files.

These files are used for auditing and statistic gathering purposes.

Guidelines for changing configuration files

These guidelines are provided to help you update the Security Access Manager configuration files. The guidelines are divided into the following categories:

- "General guidelines"
- "Default values"
- "Strings" on page 376
- "Defined strings" on page 376
- "File names" on page 376
- "Integers" on page 376
- "Boolean values" on page 377

General guidelines

Use the following general guidelines when you change the configuration settings:

- There is no order dependency or location dependency for stanzas in any configuration file.
- Stanza entries are marked as required or optional. When an entry is required, the entry must contain a valid key and value.
- Do not change the names of the keys in the configuration files. Changing the name of the key might cause unpredictable results for the servers.
- Stanza entries and key names are case-sensitive. For example, `usessl` and `UseSSL` are treated as different entries.
- Spaces are not allowed for names of keys.
- For the key value pair format of *key* = *value*, the spaces that surround the equal sign (=) are not required.
- Non-printable characters (such as tabs, carriage returns, and line feeds) that occur at the end of a stanza entry are ignored. Non-printable characters are ASCII characters with a decimal value less than 32.

Default values

Use the following guidelines when you change default configuration settings:

- Many values are created or modified only by using configuration programs. Do not manually edit these stanzas or values.
- Some values are added automatically during configuration. These values are needed for the initialization of the server after the configuration.
- The default values for a stanza entry might be different, depending on the server configuration. Some key value pairs are not applicable to certain servers and are omitted from the default configuration file for this server.

Strings

Some values accept a string value. When you manually edit the configuration file, use the following guidelines to change configuration settings that require a string:

- String values are expected to be characters that are part of the local code set.
- Additional or different restrictions on the set of allowable string characters might be imposed. For example, many strings are restricted to ASCII characters. Consult each stanza entry description for any restrictions.
- Double quotation marks are sometimes, but not always, required when you use spaces or more than one word for values. See the descriptions or examples for each stanza entry when in doubt.
- The minimum and maximum lengths of user registry-related string values, if there are limits, are imposed by the underlying registry. For example, for Active Directory the maximum length is 256 alphanumeric characters.

Defined strings

Some values accept a string value, but the value must be a set of defined strings. When you manually edit the configuration file, make sure that the string value you type matches one of the valid defined strings values.

For example, the `[aznapi-configuration]` stanza section contains the following entry:

```
mode = {local|remote}
```

The value for `mode` is expected to be `local` or `remote`. Any other value is invalid and results in an error.

File names

Some values are file names. For each stanza entry that expects a file name as a value, the description of the stanza entry specifies which of the following constructs are valid:

Filename
No directory path included.

Relative filename
A directory path is allowed but not mandatory.

These files typically are expected to be located relative to the location of a standard Security Access Manager directory. The stanza entry for each relative path name lists the root directory to which the file name is relative.

Fully qualified absolute path
An absolute directory path is required.

Some stanza entries allow more than one of the file name choices.

The set of characters that is permitted in a file name can be determined by the file system and by the local code set. For Windows operating systems, file names cannot have a backward slash (\), a colon (:), a question mark (?), or double quotation marks (").

Integers

Many stanza entries expect the value for the entry to be expressed as an integer. When you define an entry with an integer, consider the following guidelines:

- Stanza entries that take an integer value expect integer values within a valid range. The range is described in terms of a *minimum* value and a *maximum* value.

  For example, in the `[ivmgrd]` stanza, the `max-notifier-thread` stanza entry has a minimum value of 1 second and a maximum value of 128 threads.
- For some entries, the integer value must be positive, and the minimum value is 1. For other entries, a minimum integer value of 0 is allowed.

  Use caution when you set an integer value to 0. For example, an integer value of 0 might disable the function that is controlled by that stanza entry. For example, in the `[ivacld]` stanza, the entry `tcp-req-port = 0` disables the port number. Or, an integer value of 0 might indicate that the number is unlimited. For example, in the `[ldap]` stanza, the entry `max-search-size = 0` means that there is no limit to the maximum search size.
- For some entries that require integer values, Security Access Manager does not impose an upper limit for the maximum number allowed. For example, there is typically no maximum for timeout-related values, such as `timeout = *number*` in the `[ldap]` stanza.

  For this type of entry, the maximum number is limited only by the size of memory that is allocated for an integer data type. This number can vary, based on the type of operating system. For systems that allocate 4 bytes for an integer, this value is 2147483647.

  However, as the administrator, use a number that represents the value that is most logical for the value you are trying to set.

Boolean values

Many stanza entries represent a Boolean value. Security Access Manager recognizes the Boolean values `yes` and `no`.

Some of the entries in the configuration files are read by other servers and utilities. For example, many entries in the `[ldap]` stanza are read by the LDAP client. Some of these other programs recognize more Boolean characters:

- `yes` or `true`
- `no` or `false`

Anything other than `yes|true`, including a blank value, is interpreted as `no|false`.

The recognized Boolean entries are listed for each stanza entry. See the individual descriptions to determine when `true` or `false` are also recognized.

Configuration file reference

The operation of the Security Access Manager server is controlled by using configuration files. Each configuration file contains sections that are called *stanzas*.

Server configuration files are ASCII text-based and contain stanza entries. Configuration files are processed only when the servers start. The following configuration files are currently used by Security Access Manager when you use the Common Auditing Service:

pdaudit.*server_instance***.conf**

The configuration file that is used to configure the Common Auditing Service for each Security Access Manager server or server instance. For details about the stanzas that are contained in this template file, see “Common Auditing Service C client configuration files” on page 379.

Security Access Manager provides the following templates:

pdaudit.pdmgr.conf.template
The template configuration file that can be used as the base for configuring the Common Auditing Service for the Security Access Manager policy server.

pdaudit.pdproxy.conf.template
The template configuration file that can be used as the base for configuring the Common Auditing Service for a Security Access Manager policy proxy server.

pdaudit.pdacld.conf.template
The template configuration file that can be used as the base for configuring the Common Auditing Service for the Security Access Manager authorization server.

pdaudit.pdweb.conf.template
The template configuration file that can be used as the base for configuring the Common Auditing Service for a Security Access Manager WebSEAL server.

pdaudit.pdwebpi.conf.template
The template configuration file that can be used as the base for configuring the Common Auditing Service for a Security Access Manager Web server plug-in.

pdaudit.appsvr.conf.template
The template configuration file that can be used as the base for configuring the Common Auditing Service for another Security Access Manager application server.

Location of configuration files

This section provides information about the server-specific location of the configuration files.

Common Auditing Service C client

Consider that you installed Common Auditing Service in the default directories. Then, the configuration file templates for the Common Auditing Service C client are in one of the following platform-specific directories:

AIX, Linux, and Solaris operating systems
`/opt/PolicyDirector/etc/audit`

Windows operating systems
`C:\Program Files\Tivoli\Policy Director\etc\audit`

Security Access Manager base servers

If you installed Security Access Manager in the default directories, the configuration files for the base servers are in one of the following platform-specific directories:

AIX, Linux, and Solaris operating systems
`/opt/PolicyDirector/etc`

Windows operating systems
`C:\Program Files\Tivoli\Policy Director\etc`

Security Access Manager WebSEAL servers

If you did not change the installation directories during the installation of WebSEAL, its configuration files are in one of the following platform-specific directories:

AIX, Linux, and Solaris operating systems
`/opt/pdweb/etc`

Windows operating systems
`C:\Program Files\Tivoli\pdweb\etc`

Security Access Manager Plug-in for Web Servers

If you installed the Plug-in for Web Server in the default directory, its configuration files are in one of the following platform-specific directories:

AIX, Linux, and Solaris operating systems
`/opt/pdwebpi/etc`

Windows operating systems
`C:\Program Files\Tivoli\pdwebpi\etc`

Contents of configuration files

This section provides information about the stanzas and stanza entries in the available configuration files. The configuration files are used for auditing and statistic gathering purposes.

Security Access Manager configuration files

Within the configuration files for the Security Access Manager servers, you can define auditing and statistics characteristics. All C-based servers have the `[aznapi-configuration]` stanza, and WebSEAL has an additional `[logging]` stanza.

Common Auditing Service C client configuration files

To use the Common Auditing Service to create Security Access Manager audit reports, you must have a server-specific audit configuration file. Use this configuration file to customize auditing operations for the Security Access Manager server.

This configuration file can include the following stanza:
- `[cars-client]`
- `[cars-filter]`
- `[pdaudit-filter]`

Configuration file stanza reference

Within configuration files, stanza labels are shown within brackets, such as [*stanza-name*]. For example, the `[ssl]` stanza in the `ivmgrd.conf` configuration file defines the Secure Sockets Layer (SSL) configuration settings for the policy server. The `[ldap]` stanza defines the configuration settings that are required by the policy server to communicate with an LDAP-based user registry.

Each stanza in a Security Access Manager configuration file contains one or more key value pairs, which contain information that is expressed as a paired set of parameters. Each stanza entry is a key-value pair in the following format:

```
key = value
```

You must not change the names of the keys in the configuration files. Changing the name of the key might cause unpredictable results in the servers. The spaces that surround the equal sign (=) are not required.

The initial installation of Security Access Manager establishes many of the default values. Some values are static and never change; other values can be modified to customize server functionality and performance.

The following stanza descriptions provide a list of the valid stanza entries. Each stanza entry consists of key value pairs. Each stanza entry includes a description of its default behavior, when applicable.

[aznapi-configuration] stanza

The stanza entries for native Security Access Manager auditing and statistics gathering are in the `[aznapi-configuration]` stanza of the server-specific configuration file. The `[aznapi-configuration]` stanza contains more entries than the ones that are listed. For a complete list of entries that can be used in the server-specific configuration files, see the administration guide for that server or plug-in.

logcfg

Syntax

```
logcfg = category:[log-agent][[parameter[=value]] ...]
```

Description

Enables logging and auditing for the application. Category, destination, and other parameters are used to capture Security Access Manager auditing and logging events.

Each server provides its own event logging setting in its corresponding configuration file.

Options

category:*log-agent*

The category of the auditing event and the destination. *log-agent* is one of the following agents:

- `stdout`
- `stderr`
- `file path=`
- `pipe`
- `remote`

parameter=*value*

Allowable parameters. The parameters vary, depending on the category, the destination of events, and the type of auditing you want to perform.

See Chapter 19, “Audit event logging,” on page 173 for information about the log agents and the configuration parameters. Each log agent supports different parameters.

Usage

Optional

Default value

Remove the pound signs (#) at the beginning of the configuration file lines to enable authentication or authorization auditing (or both) for the application.

Example

```
logcfg = audit.azn:file path=audit.log,flush_interval=20,log_id=audit_log
```

[cars-client] stanza

The [cars-client] stanza contains the configuration of the client for the Common Auditing Service. The entries in this stanza specify the characteristics of the connection to the Common Auditing Service audit server and how the client processes audit events. You must specify the doAudit and serverURL entries. If these entries are not specified, the Common Auditing Service is not configured for use by Security Access Manager.

If secure communication is required between the client and audit server, specify the keyFilePath and stashFilePath entries.

Exercise care when you change entry values. Behavior is undefined if entries are # set to values that are not documented. Behavior is also undefined if numeric values are # specified larger than the ones supported by the architecture.

The stanza entry for filtering events is in the [cars-filter] stanza.

diskCachePath

Syntax

diskCachePath = *fully_qualified_path*

Description

Specifies the name and location of the file to be used to cache events. The file must exist at the specified location.

When events are written to the disk cache file, a cache manager thread periodically checks to determine whether the audit server can accept events. The thread uses the setting of the rebindInterval entry. When the service is available, the cache manager sends the events from the disk cache file.

The name of the disk cache file must be unique. If more than one server or server instance is configured to use the same disk cache file, errors occur.

Options

fully_qualified_path

Represents an alphanumeric string. String values are expected to be characters that are part of the local code set. The set of characters that is permitted in a file name can be determined by the file system and by the local code set. For Windows operating systems, file names cannot have a backward slash (\), a colon (:), a question mark (?), or double quotation marks ("). For AIX, Linux, and Solaris operating systems, path names and file names are case-sensitive.

Usage

Conditional. This entry is used only when the `useDiskCache` entry is set to `auto` or `always`. This entry must be specified if the `useDiskCache` entry is set to `auto` or `always`.

Default value

There is no default value.

doAudit

Syntax

```
doAudit = {yes|no}
```

Description

Specifies whether auditing with the Common Auditing Service is enabled or disabled. When auditing is disabled, events are not forwarded to the audit server.

Note: Ensure that you specify either `yes` or `no` as the value. If either of these values is not specified, or the value is specified incorrectly, events are not forwarded to the Common Auditing Service audit server.

After you configure the Common Auditing Service, you can start auditing by using the following steps:

1. Enter the following commands:
   ```
   > pdadmin login -l
   pdadmin local> config modify keyvalue set config_file cars-client doAudit yes
   ```
2. Restart the server.

To stop auditing, complete the following steps:

1. Enter the following commands:
   ```
   > pdadmin login -l
   pdadmin local> config modify keyvalue set config_file cars-client doAudit no
   ```
2. Restart the server.

Options

yes Enables auditing by using the Common Auditing Service.

no Disables auditing for the Common Auditing Service. *No* is the default value.

Usage

Required

Default value

The default value is `no`.

Example

```
doAudit = yes
```

clientPassword

Syntax

`clientPassword` = *password*

Description

Specifies the password for the WebSphere audit ID. This password is stored in the obfuscated version of the configuration file.

Usage

Conditional. This stanza entry is required only when you use secure communications with the Web service.

Default value

There is no default value.

clientUserName

Syntax

`clientUserName` = *user_id*

Description

Specifies the WebSphere audit ID used by the administrator. This ID is authenticated with HTTP basic authentication.

Usage

Conditional. This stanza entry is required only when you use secure communications with the Web service.

Default value

There is no default value.

errorFilePath

Syntax

`errorFilePath` = *fully_qualified_path*

Description

Specifies the name and location of the error log file. If the file does not exist at the specified location, the server identity creates the file.

The name of the log file must be unique. If more than one server or server instance is configured to use the same log file, an error occurs.

Options

fully_qualified_path
: Represents an alphanumeric string. String values are expected to be characters that are part of the local code set. The set of characters that is

permitted in a file name can be determined by the file system and by the local code set. For Windows operating systems, file names cannot have a backward slash (\), a colon (:), a question mark (?), or double quotation marks ("). For AIX, Linux, and Solaris operating systems, path names and file names are case-sensitive.

Usage

Optional

Default value

There is no default value.

flushInterval

Syntax

`flushInterval` = *interval*

Description

Limits the time an event waits in the queue before it is forwarded to the audit server. Use this entry to forward the events in the queue at the designated interval when:

- Events are generated at a slow rate.
- The queue does not reach the high water mark in a timely manner.

Options

interval
Specifies the number of seconds that an event waits in the queue.

Usage

Conditional. This entry is used when the `useDiskCache` entry is set to `auto` or `never`.

Default value

The default value is `2`.

Example

```
flushInterval = 600
```

keyFilePath

Syntax

`keyFilePath` = *fully_qualified_path*

Description

Specifies the SSL key file name and location. Use the SSL key file to handle certificates that are used to communicate with the common event Web service. The file extension can be anything, but the extension is usually `kdb`.

Options

fully_qualified_path

Represents an alphanumeric string. String values are expected to be characters that are part of the local code set.

The set of characters that is permitted in a file name can be determined by the file system and by the local code set.

For Windows operating systems, file names cannot have a backward slash (\), a colon (:), a question mark (?), or double quotation marks (").

For AIX, Linux, and Solaris operating systems, path names and file names are case-sensitive.

Usage

Conditional. This stanza entry is required only when you use secure communications with the Web service.

Default value

There is no default value.

lowWater

Syntax

`lowWater` = *number*

Description

Specifies the smallest number of events that can be in the queue before events are:

- No longer removed from the queue, and
- Written to the disk cache file.

When the audit server is slow and the event queue fills up, events are removed from the queue and written to the disk cache file. Events are written in this manner until the number of events in the queue is equal to or less than the low water mark. When this low water mark is reached, queued events are sent directly to the audit server.

Usage

Conditional. This entry is used when the `useDiskCache` entry is set to `auto`.

Default value

There is no default value.

hiWater

Syntax

`hiWater` = *number*

Description

Specifies the maximum number of events that can be in the queue. When this high water mark is reached, events are sent to the audit server.

Usage

Optional. This entry is used when the `useDiskCache` entry is set to `auto` or `never`.

Default value

The default value is `80`.

Example

```
hiWater = 30
```

maxCacheFiles

Syntax

`maxCacheFiles` = *number*

Description

Specifies the maximum number of disk cache files that can be created. Unlike error log and trace files, disk cache files can be used again.

After all the events in the disk cache file are sent to the audit server, the cache manager deletes that cache file.

Usage

Conditional. This entry is used when the `useDiskCache` entry is set to `auto` or `always`.

Default value

The default value is `50`.

maxCacheFileSize

Syntax

`maxCacheFileSize` = *size*

Description

Specifies the maximum size in bytes of the disk cache file. When this size is reached, the cache file rolls over and a new cache file is created. The maximum size is 1 GB (1,073,741,824 bytes).

Usage

Conditional. This entry is used when the `useDiskCache` entry is set to `auto` or `always`.

Default value

The default value is `10485760`.

maxErrorFiles

Syntax

maxErrorFiles = *number*

Description

Specifies the maximum number of error log files that can be created before the oldest log file is used again.

Usage

Optional

Default value

There is no default value.

maxErrorFileSize

Syntax

maxErrorFileSize = *size*

Description

Specifies the maximum size in bytes of the error log file. When this size is reached, the log file rolls over and a new error log file is created. For more information about how log files roll over, see the *IBM Security Access Manager for Web Troubleshooting Guide*.

Usage

Optional

Default value

There is no default value.

maxTraceFiles

Syntax

maxTraceFiles = *number*

Description

Specifies the maximum number of trace files that can be created before the oldest trace file is used again.

Usage

Optional

Default value

There is no default value.

maxTraceFileSize

Syntax

`maxTraceFileSize` = *size*

Description

Specifies the maximum size in bytes of the trace log file. When this size is reached, the log file rolls over and a new error log file is created. For more information about how log files roll over, see the *IBM Security Access Manager for Web Troubleshooting Guide*.

Usage

Optional

Default value

There is no default value.

numberCMThreads

Syntax

`numberCMThreads =` *`number_of_threads`*

Description

Specifies the number of threads to create for the cache manager. These threads read events from the disk cache files and send them to the audit server.

Options

number_of_threads
Represents a numeric value.

Usage

Optional. This entry is used when the `useDiskCache` entry is set to `auto` or `always`.

Default value

The default value is 8.

Example

```
numberCMThreads = 2
```

numberEQThreads

Syntax

`numberEQThreads =` *`number_of_threads`*

Description

Specifies the number of threads to create to service the event queue.

Options

number_of_threads
Represents a numeric value.

Usage

Optional. This entry is used when the `useDiskCache` entry is set to `auto` or `never`.

Default value

The default value is `8`.

Example

```
numberEQThreads = 2
```

numberRetries

Syntax

`numberRetries` = *number*

Description

When an error occurs during a network transfer, specifies the number of attempts before the data is transferred.

Usage

Optional

Default value

The default value is `3`.

queueSize

Syntax

`queueSize` = *size*

Description

Specifies the maximum number of audit events that can be queued.

Usage

Optional. This entry is used when the `useDiskCache` entry is set to `auto` or `never`.

Default value

The default value is `400`.

rebindInterval

Syntax

`rebindInterval` = *seconds*

Description

Specifies that number of seconds that the cache manager waits before attempting to establish a connection to the audit server.

Usage

Conditional. This entry is used when the `useDiskCache` entry is set to `auto` or `always`.

Default value

There is no default value.

retryInterval

Syntax

`retryInterval` = *seconds*

Description

When an error occurs during a network transfer, specifies the number of seconds to wait before another attempt is made to send the data.

Usage

Optional

Default value

The default value is `2`.

serverURL

Syntax

`serverURL` = *url*

Description

Specifies the URL of the Common Auditing Service. For secure communication, use the following URL:

> https://*hostname*:*WC_defaulthost_secure_port_number*/CommonAuditService/service/Emitter

For nonsecure communication, use the following URL:

> http://*hostname*:*WC_defaulthost_port_number*/CommonAuditService/service/Emitter

Note: Ensure that you specify the correct URL. If the value is specified incorrectly, events are not forwarded to the Common Auditing Service audit server. The page that displays in the browser must be similar to the following message:

```
{urn:ibm:cars:10}Emitter

Hi there, this is a Web service!
```

Options

url The URL of the Common Auditing Service.

Usage

Required

Default value

There is no default value.

stashFilePath

Syntax

`stashFilePath` = *fully_qualified_path*

Description

Specifies the SSL password stash file name and location. The password is used to protect private keys in the key file. The password might be stored encrypted in the stash file. The file extension can be anything, but it is typically `sth`.

Options

fully_qualified_path
: Represents an alphanumeric string. String values are expected to be characters that are part of the local code set. The set of characters that is permitted in a file name can be determined by the file system and by the local code set. For Windows operating systems, file names cannot have a backward slash (\), a colon (:), a question mark (?), or double quotation marks ("). For AIX, Linux, and Solaris operating systems, path names and file names are case-sensitive.

Usage

Conditional. This stanza entry is required only when you use secure communications with the Web service.

Default value

There is no default value.

tempStorageFullTimeout

Syntax

`tempStorageFullTimeout` = *wait_time*

Description

Specifies the number of seconds that the Common Auditing Service client waits before it discards cached events when temporary cache storage is filled.

This timeout is not intended to provide precise control. It takes effect when the event queue is full, and the disk cache cannot be written to. The reasons why the disk cache can become inaccessible include:

- The maximum value that was specified for the maxCacheFiles property was reached, and each cache file reached the maximum value that was specified for the maxCacheFileSize property.
- The cache file cannot be written to because a system error occurred.

You can also use this property to tune audit event processing in Common Auditing Service clients. The tuning is useful when the Common Auditing Service audit server is:

- Available, and
- Not keeping pace with the audit event processing of the Common Auditing Service clients.

The larger the value, the longer a Common Auditing Service client waits before it discards events.

Options

wait_time

Set *wait_time* to:

- Zero for no waiting.
- A positive integer to indicate the number of seconds to wait.
- -1 to indicate to wait forever.

Note: Ensure that you specify a valid value when you use this property. Using tempStorageFullTimeout without specifying a valid value can cause unpredictable behavior.

Usage

Conditional. This entry is used only when the `useDiskCache` entry is set to `auto` or `always`.

Default value

The default value is `0` for no waiting.

Example

```
tempStorageFullTimeout = -1
```

traceLevel

Syntax

`traceLevel` = *level*

Description

Specifies the level of trace events to write to the trace log. The following settings are valid:

1 Indicates that only events that result from error conditions are written to the log.

2 Indicates that only the following events are written to the log file:
- Error conditions
- Entry and exit trace points

3 Indicates that events that result from error conditions and from all trace points in the code are written to the log. Output is written to the file specified by the `traceFilePath` parameter. The output includes the properties that are defined in this configuration file, and the values that are sent to the Common Audit Service audit server.

Usage

Conditional. Required when `traceFilePath` is defined.

Default value

There is no default value.

traceFilePath

Syntax

`traceFilePath` = *fully_qualified_path*

Description

Specifies the name and location of the trace file. If the file does not exist at the specified location, then the server identity creates the file.

The name of the trace file must be unique. If more than one server or server instance is configured to use the same trace file, errors occur.

Options

fully_qualified_path
: Represents an alphanumeric string. String values are expected to be characters that are part of the local code set. The set of characters that is permitted in a file name can be determined by the file system and by the local code set. For Windows operating systems, file names cannot have a backward slash (\), a colon (:), a question mark (?), or double quotation marks ("). For AIX, Linux, and Solaris operating systems, path names and file names are case-sensitive.

Usage

Optional

Default value

There is no default value.

transferSize

Syntax

`transferSize` = *size*

Description

Number of audit events to send on each network transfer.

Usage

Optional

Default value

The default value is 40.

useDiskCache

Syntax

```
useDiskCache = {auto|always|never}
```

Description

Specifies whether to enable disk caching, and, when enabled, indicates how to handle disk caching.

Options

always Indicates that audit events are always written directly to the disk cache on the caller thread. There is no event queue.

never Indicates that audit events are written to the event queue. There is no disk cache.

auto Indicates that audit events are written to the event queue except when the server is down or the event queue is full. Under these conditions, the audit events are written to disk cache.

Usage

Optional

Default value

The default value is auto.

[cars-filter] stanza

The stanza entry for common audit filtering of the IBM Security Access Manager runtime is in the [cars-filter] stanza of the pdaudit.conf file.

auditevent

Syntax

```
auditevent = type, [outcome=outcome]
```

Description

Identifies the events to be captured for auditing. Events can be identified by event type, application name, and outcome. If an event logged by an application matches any configured filter entry (auditevent or outcome), it is forwarded to the Common Auditing Service audit server.

For each event type to capture, the configuration file must include a separate stanza entry.

To add event types to the event filter, use the **config modify** command with the **append** option.

To remove event types from the event filter, use the **config modify** command with the **remove** option.

Note: With the `auditevent` entry, do not use the config modify command with the **set** option. Using the **set** option overwrites the first `auditevent` entry in the configuration file.

Options

type
Specifies one of the following event types:

authn
Indicates authentication events. This event type can be used with all Security Access Manager servers.

authn_creds_modify
Indicates events that modify credentials for users. This event type can be used with all Security Access Manager servers.

authn_terminate
Indicates termination events. These types of events are the results of a timeout, an administrator who terminates a session, or a user-initiated log out. This event type can be used with all Security Access Manager servers.

authz
Indicates authorization events. This event type can be used with all Security Access Manager servers.

mgmt_config
Indicates configuration and other management events for a server. This event type can be used with the policy server.

mgmt_policy
Indicates events for security policy management, such as the creation of an ACL. This event type can be used with the policy server.

mgmt_registry
Indicates events for registry management, such as creating users and groups, administrator-initiated password changes, and modifying properties of users and groups. This event type can be used with the policy server.

mgmt_resource
Indicates event for resource events. This event type can be used with the policy server.

password_change
Indicates events for user-initiated password changes. This event type can be used with the policy server, WebSEAL server, or the plug-in for Web servers.

Administrator-initiated password changes are classified as registry management events.

resource_access
Indicates events that record all accesses to a resource, such as a file or HTTP request and response events outside of authorization events. This event type can be used with the WebSEAL server or the plug-in for Web servers.

runtime
Indicates runtime events, such as starting and stopping security servers. Events that were generated from administrator-initiated tasks that are classified as management tasks. This event type can

be used with all Security Access Manager servers. Additionally, this event type can be used for reporting WebSEAL statistics.

outcome=*outcome*
Specifies one of the following outcomes:

all Records all outcomes. The default value is *all*.

success
Records successful outcomes only.

unsuccessful
Records unsuccessful outcomes only.

unknown
Records outcomes where success can not be determined. This value applies to authz and resource_access event types only.

Usage

Required.

Default value

There is no default value.

Example

```
auditevent = authn, outcome=unsuccessful
auditevent = authz, outcome=unknown
```

[logging] stanza

The [logging] stanza contains the configuration details for logging HTTP audit events for WebSEAL servers. WebSEAL can be configured to maintain the following HTTP activities:

- agents
- referers
- requesters

The [logging] stanza is in the WebSEAL webseald.conf configuration file. Assume that the configuration file contains auditing entries in both the [aznapi-configuration] stanza and the [logging] stanza. Then, the logging details in the [aznapi-configuration] stanza take precedence over repeated details in the [logging] stanza.

For details about WebSEAL event processing, see “Process flow for logcfg logging” on page 195. For information about the [aznapi-configuration] stanza entries in the WebSEAL webseald.conf configuration file, see the *IBM Security Access Manager for Web WebSEAL Administration Guide*.

absolute-uri-in-request-log

Syntax

```
absolute-uri-in-request-log = {yes|no}
```

Description

Logs the absolute URI in the HTTP audit records. Adds protocol and host to the path.

Options

yes Log the absolute URI.

no Do not log the absolute URI.

Usage

This stanza entry is required.

Default value

```
no
```

Example

```
absolute-uri-in-request-log = no
```

agents

Syntax

```
agents = {yes|no}
```

Description

Enables or disables the agents log. This log records the contents of the `User_Agent:` header of each HTTP request.

Options

yes The value `yes` enables logging for the agents.

no The value `no` disables logging for the agents.

Usage

This stanza entry is required.

Default value

```
yes
```

Example

```
agents = yes
```

agents-file

Syntax

```
agents-file = fully_qualified_path
```

Description

Fully qualified path to the agents log file.

Options

fully_qualified_path
Fully qualified path to the agents log file.

Usage

This stanza entry is required.

Default value

The default location is `www-instance/log/agent.log`, located under the WebSEAL installation directory.

Example

Example on AIX, Linux, and Solaris:

```
agents-file = /var/pdweb/www-web1/log/agent.log
```

config-data-log

Syntax

```
config-data-log = fully_qualified_path
```

Description

Fully qualified path to the configuration data log file.

Options

fully_qualified_path
: Fully qualified path to the configuration data log file.

Usage

This stanza entry is required.

Default value

The default location is `log/config_data.log`, located under the WebSEAL installation directory.

Example

Example on AIX, Linux, and Solaris:

```
config-data-log = /var/pdweb/log/config_data.log
```

flush-time

Syntax

```
flush-time = number_of_seconds
```

Description

Integer value that indicates the frequency, in seconds, to force a flush of log buffers.

Options

number_of_seconds
: Integer value that indicates the frequency, in seconds, to force a flush of log buffers. The minimum value is `1` second. The maximum value is `600` seconds.

Usage

This stanza entry is optional.

Default value

```
20
```

Example

```
flush-time = 20
```

gmt-time

Syntax

```
gmt-time = {yes|no}
```

Description

Enables or disables logging requests in Greenwich Mean Time (GMT) instead of the local time zone.

Options

yes A value of `yes` means to use GMT.

no A value of `no` means to use the local time zone.

Usage

This stanza entry is required.

Default value

```
no
```

Example

```
gmt-time = no
```

host-header-in-request-log (deprecated)

Syntax

```
host-header-in-request-log = {yes|no}
```

Description

Log the `Host` header at the front of each line in the request log and the combined log.

Options

yes Log the `Host` header.

no Do not log the `Host` header.

Usage

This stanza entry is required.

Default value

```
no
```

Example

```
host-header-in-request-log = no
```

max-size

Syntax

max-size = *number_of_bytes*

Description

Integer value that indicates the size limit of the log files. This value applies to the request, referrer, and agent logs. The size limit is also known as the rollover threshold. When the log file reaches this threshold, the original log file is renamed, and a new log file with the original name is created.

Options

number_of_bytes

When the value is zero (`0`), no rollover log file is created.

When the value is a negative integer, the logs are rolled over daily, regardless of the size.

When the value is a positive integer, the value indicates the maximum size, in bytes, of the log file before the rollover occurs. The allowable range is from 1 byte to 2 MB.

Usage

This stanza entry is required.

Default value

```
2000000
```

Example

```
max-size = 2000000
```

referers

Syntax

```
referers = {yes|no}
```

Description

Enables or disables the referers log. This log records the `Referer:` header of each HTTP request.

Options

yes The value `yes` enables referers logging.

no The value `no` disables referers logging.

Usage

This stanza entry is required.

Default value

```
yes
```

Example

```
referers = yes
```

referers-file

Syntax

```
referers-file = fully_qualified_path
```

Description

Fully qualified path to the referers log file.

Options

fully_qualified_path
Fully qualified path to the referers log file.

Usage

This stanza entry is required.

Default value

The default location is `www-instance/log/referer.log`, located under the WebSEAL installation directory.

Example

Example on AIX, Linux, and Solaris:

```
referers-file = /var/pdweb/www-web1/log/referer.log
```

requests

Syntax

```
requests = {yes|no}
```

Description

Enables or disables the requests log. This log records standard logging of HTTP requests.

Options

yes The value `yes` enables requests logging.

no The value `no` disables requests logging.

Usage

This stanza entry is required.

Default value

```
yes
```

Example

```
requests = yes
```

requests-file

Syntax

```
requests-file = fully_qualified_path
```

Description

Fully qualified path to the request log file.

Options

fully_qualified_path
: Fully qualified path to the request log file.

Usage

This stanza entry is required.

Default value

The default location is `www-instance/log/request.log`, located under the WebSEAL installation directory.

Example

Example on AIX, Linux, and Solaris:

```
requests-file = /var/pdweb/www-web1/log/request.log
```

server-log

Syntax

```
server-log = fully_qualified_path
```

Description

Fully qualified path to the server error log file.

Options

fully_qualified_path
: Fully qualified path to the server error log file.

Usage

This stanza entry is required.

Default value

The default location is `log/webseald.log`, located under the WebSEAL installation directory.

Example

Example on AIX, Linux, and Solaris:

```
server-log = /var/pdweb/log/msg__webseald.log
```

[pdaudit-filter] stanza

The stanza entries for native Security Access Manager auditing are in the `[pdaudit-filter]` stanza of the server-specific `pdaudit.conf` configuration file. Use the `logcfg` entries in the `[pdaudit-filter]` stanza only if configured auditing for use with the Common Auditing Service.

logcfg

Syntax

```
logcfg = category:[log-agent][[parameter[=value]] ...]
```

Description

Enables logging and auditing for the application. Category, destination, and other parameters are used to capture Security Access Manager auditing and logging events.

Each server provides its own event log setting in its corresponding configuration file.

Options

category:*log-agent*

The category of the auditing event and the destination. *log-agent* is one of the following agents:

- `stdout`
- `stderr`
- `file path=`
- `pipe`
- `remote`

parameter=*value*

Allowable parameters. The parameters vary, depending on the category, the destination of events, and the type of auditing that you want to complete.

See Chapter 19, “Audit event logging,” on page 173 for information about the log agents and the configuration parameters. Each log agent supports different parameters.

Usage

Optional

Default value

Remove the number signs (#) at the beginning of the configuration file lines to enable authentication or authorization auditing (or both) for the application.

Example

```
logcfg = audit.azn:file path=audit.log,flush_interval=20,log_id=audit_log
```

Appendix C. Commands and utilities

This section provides reference information about the commands and utilities that are used for auditing, statistics gathering, and for viewing and changing entries in configuration files.

Reading syntax statements

The reference documentation uses the following special characters to define syntax:

[ ] Identifies optional options. Options that are not enclosed in brackets are required.

... Indicates that you can specify multiple values for the previous option.

| Indicates mutually exclusive information. You can use the option to the left of the separator or the option to the right of the separator. You cannot use both options in a single use of the command.

{ } Delimits a set of mutually exclusive options when one of the options is required. If the options are optional, they are enclosed in brackets ([ ]).

**** Indicates that the command line wraps to the next line. It is a continuation character.

The options for each command or utility are listed alphabetically in the Options section or in the Parameters section. When the order of the options or parameters must be used in a specific order, this order is shown in the syntax statements.

Commands

Table 68 lists the **pdadmin** commands that can be used during auditing and gathering of statistics activities.

Table 68. Auditing and statistics commands

Command	Description
"config modify"	Modifies a stanza entry in a configuration file or sets the password for the server user account.
"config show" on page 408	Shows the value that is associated with the specified stanza and key in a configuration file.
"login" on page 409	Establishes authentication credentials that are used during communication with the Security Access Manager policy server.
"server list" on page 411	Lists all registered Security Access Manager servers.
"server task stats" on page 412	Enables the gathering of statistical information for an installed Security Access Manager server or server instance.

config modify

Modifies a stanza entry in a configuration file or sets the password for the server user account.

Syntax

config modify keyvalue append [–obfuscate] *config_file stanza key value*

config modify keyvalue remove [–obfuscate] *config_file stanza key* [*value*]

config modify keyvalue set [–obfuscate] *config_file stanza key value*

config modify svrpassword *config_file password*

Description

The **config modify** command either modifies a stanza entry in a configuration file or sets the password for the application server user account. Depending on which configuration operation you want, you must either perform a local login or a remote login.

- To set the password for the server user account by using the svrpassword option, perform a remote login by using:
  - The **login** command.
  - The **login** command with the –d option.
  - The **login** command with the –m option.
- To modify the value of a stanza entry in a configuration file by using the keyvalue option, perform a local login. Use the **login** command with the –l option.

Note: If you attempt to run one of the configuration operations that requires a local login, an error is displayed.

```
Error: HPDMS4061ELocal authentication (local login) is required to perform
this operation (status 0x14c52fdd)
```

To use the svrpassword option, you must:

- Be defined in the ACL policy.
- Have the Password permission - **W** action bit.
- Have the necessary operating system permissions to modify the local configuration file.

To use the keyvalue options, you must have the necessary operating system permissions to read and modify the configuration file.

For key values that are not obfuscated, use the **config show** command to display modified values.

For information and guidelines about the stanzas and stanza entries in configuration files, see the *IBM Security Access Manager for Web Stanza Reference*.

Options

–obfuscate

Indicates that the stanza entry must be written to or removed from the obfuscated (.obf file) version of the configuration file, which is specified by *config_file*. (Optional)

config_file

Specifies the fully qualified name of the configuration file, unless the configuration file is in the current directory. When used with the –obfuscate option, do not specify the .obf extension as part of the configuration file name.

key Specifies the key portion of the stanza entry.

keyvalue append
Adds a value to a stanza entry in the configuration file stanza. If you attempt to append a duplicate value to a key, the duplicate value is ignored.

keyvalue remove
Removes a value from a stanza entry in the configuration file stanza. If you do not specify the *value* option, the key is removed from the configuration file.

keyvalue set
Defines a stanza entry (key value pair) or changes the value of a key in the configuration file stanza.

password
Specifies the password for the application server account.

stanza Specifies the name of stanza that contains the stanza entry.

svrpassword
Sets the password for the application server account. This password is updated in the registry and in the obfuscated version of the local configuration file.

value Specifies the configuration value for the key.

Authorization

No authentication is required, except for the svrpassword option. The svrpassword option requires authentication (administrator ID and password).

Return codes

0 The command completed successfully.

1 The command failed. When a command fails, the **pdadmin** command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2).

See the *IBM Security Access Manager for Web Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

Examples

- The following example provides a local login:
```
pdadmin> login -l
```
After a local login, the prompt changes from pdadmin> to pdadmin local>.
- After a local login, the following example changes the value of the version key in the [meta-info] stanza of the d:\temp\my.conf configuration file to 6798:
```
pdadmin local> config modify keyvalue set d:\temp\my.conf \
meta-info version 6798
```
- After a local login, the following example adds the new key mynewvalue to the [meta-info] stanza of the d:\temp\my.conf.obf configuration file. The example sets the value of the new key to 14 in a new obfuscated stanza entry:

```
pdadmin local> config modify keyvalue set -obfuscate d:\temp\my.conf \
meta-info mynewkey 14
```

Note: In the **config modify** command above, the name of the configuration file does not have the .obf file extension.

See also

"config show"
"login" on page 409

config show

Shows the value that is associated with the specified stanza and key in the Security Access Manager server configuration files or in customized server configuration files. The stanza and key must exist, or an error is displayed.

Requires a local login to use this command. No authentication is required.

Syntax

config show *config_file stanza key*

Options

config_file
Specifies the Security Access Manager or custom configuration file to use. Unless the configuration file is in the current directory, the configuration file name must be a fully qualified path name. The necessary operating system permissions are required to read and update the configuration file. The default names for the configuration files are documented in the Security Access Manager administration guides.

key
Specifies the configuration value to associate with the key in the specified configuration file stanza. Valid key-value pairs are documented in the Security Access Manager administration guides.

stanza
Specifies the name of a Security Access Manager or custom stanza that contains the input key. A valid stanza name is an alphanumeric string that is not case-sensitive. String values are expected to be characters that are part of the local code set. Valid stanzas are documented in the Security Access Manager administration guides.

Return codes

0
The command completed successfully.

1
The command failed. When a command fails, the **pdadmin** command provides a description of the error and an error status code in hexadecimal format (for example, 0x14c012f2). See the *IBM Security Access Manager for Web: Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

Examples

- The following example provides a local login and requests the value of the version key for the [meta-info] stanza. The value is 1296. The prompt changes to show that the login is local:

```
pdadmin> login -l

pdadmin local> config show "c:\Program Files\Tivoli\Policy
Directory\etc\activedir.conf" meta-info version
```

Provides output like:

```
1296
```

- The following example provides a local login and requests the value of the enabled key for the [ldap] stanza. The output provides a key value of yes. The prompt changes to show that the login is local:

  ```
  pdadmin>login -l

  pdadmin local> config show "C:\Program Files\Tivoli\Policy Director\etc\ldap.conf"
  ldap enabled
  ```

  Provides output like:

  ```
  yes
  ```

See also

"config modify" on page 405
"login"

login

Establishes authentication credentials that are used for communication with the Security Access Manager policy server. These credentials are used to determine access privileges for the user to policy server data. Most commands cannot be performed unless an explicit login is done.

This command does not require a login or authentication to use.

Syntax

login –a *admin_id* [–p *password*] [–d *domain*]

login –a *admin_id* [–p *password*] [–m]

login –l

Description

Credentials are used to determine user access privileges to policy server data. Except the **context, errtext**, **exit**, **help**, **login**, **logout**, and **quit** commands, and the local configuration commands, a user ID, and a password are needed for authentication.

Credentials are not accumulated or stacked. A **login** command completely replaces any existing credentials.

In interactive mode, the **pdadmin** prompt changes, depending on how the user logs in:

- Not interactive mode. This command starts the **pdadmin** utility. In interactive mode, the **login** commands are entered from the pdadmin> prompt.

  ```
  c:\> pdadmin
  pdadmin>
  ```

- A user local login that is performed for local configuration. No authentication is required.

  ```
  pdadmin> login -l
  pdadmin local>
  ```

- An administrator login that is performed to the local domain. In some cases, the local domain might be the management domain, which is named `Default`. Authentication is required.

  ```
  pdadmin> login -a sec_master -p secmstrpw
  pdadmin sec_master>
  ```

- A user login that is performed to the local domain. Authentication is required.

  ```
  pdadmin> login -a dlucas -p lucaspw
  pdadmin dlucas>
  ```

- A user login that is performed to another domain other than their local domain. Authentication is required.

  ```
  pdadmin> login -a dlucas -p lucaspw -d domain_a
  pdadmin dlucas@domain_a>
  ```

- A user login that is performed to the management domain. Authentication is required.

  ```
  pdadmin> login -a dlucas -p lucaspw -m
  pdadmin dlucas@Default>
  ```

Options

–a *admin_id*
: Specifies an administrator ID.

–d *domain*
: Specifies the Security Access Manager secure domain for the login. The *admin_id* user must exist in this domain.

–m
: Specifies that the login operation must be directed to the management domain. The *admin_id* user must exist in this domain.

  Note: Only one of the following domain options can be specified: –d *domain* or –m. If neither option is specified, the target domain is the local domain that is configured for the system. The *admin_id* user must exist in the target domain, whether it is explicitly specified.

–p *password*
: Specifies the password for the *admin_id* user. If this option is not specified, the user is prompted for the password. The password cannot be specified if the *admin_id* is not specified.

–l
: Specifies a local login operation. When modifications are made to local configuration files by using the **config** commands, a local login is required before you can run commands. The user can run the **context show** command to view more authentication information.

Return codes

0
: The command completed successfully.

1
: The command failed. When a command fails, the **pdadmin** command provides a description of the error and an error status code in hexadecimal format (for example, `0x14c012f2`). See the *IBM Security Access Manager for Web Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

Examples

- The following example logs the `sec_master` user in to the management domain and then displays the authentication context for the user:

  ```
  pdadmin> login -a sec_master -p pa55w0rd -m

  pdadmin sec_master> context show

  User: sec_master
  Domain: Default
  The user is logged in to the management domain.
  ```

- The following example logs in a user to the `domain1` domain and then displays the authentication context for the user:

  ```
  pdadmin> login -a domain1_admin -p d0main1pwd -d domain1

  pdadmin domain1_admin@domain1> context show

  User: domain1_admin
  Domain: domain1
  The user is not logged in to the management domain
  ```

- The following example interactively logs in the user to their local domain that is configured for the system. The domain name is `testdomain`. The example then displays the authentication context of the user:

  ```
  pdadmin> login
  Enter User ID: testdomain_admin
  Enter password: adminpwd

  pdadmin testdomain_admin> context show

  User: testdomain_admin
  Domain: testdomain
  The user is not logged in to the management domain
  ```

- The following example of a local login demonstrates how the prompt changes, depending on the type of interactive login:

  ```
  c:\> pdadmin login -l
  ```

  Provides this prompt:

  ```
  pdadmin local>
  ```

server list

Lists all registered Security Access Manager servers.

Requires authentication (administrator ID and password) to use this command.

Syntax

```
server list
```

Description

Lists all registered Security Access Manager servers. The name of the server for all server commands must be entered in the exact format as it is displayed in the output of this command. The **`server list`** command does not have such a requirement.

Options

None.

Return codes

0 The command completed successfully.

1 The command failed. When a command fails, the **pdadmin** command provides a description of the error and an error status code in hexadecimal format (for example, `0x14c012f2`). See the *IBM Security Access Manager for Web Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

Example

The following example lists registered servers:

```
pdadmin> server list
```

The output is as follows:

```
ivmgrd-master
ivacld-server1
ivacld-server2
```

where `ivmgrd-master` represents the Policy server; `ivacld-server2` and `ivacld-server1` represent Authorization server instances.

server task stats

Manages the gathering and reporting of statistics for Security Access Manager servers and server instances.

Requires authentication (administrator ID and password) to use this command.

Syntax

server task *server_name–host_name* stats get [*component*]

server task *server_name–host_name* stats list

server task *server_name–host_name* stats off [*component*]

server task *server_name–host_name* stats on *component* [*interval* [*count*]] [*destination*]

server task *server_name–host_name* stats reset [*component*]

server task *server_name–host_name* stats show [*component*]

Description

The **server task stats** command manages the gathering and reporting of statistics for Security Access Manager servers and server instances. You can use the **stats** commands with configuration setting that are defined by the stanza entries in the server configuration file to manage statistics.

Statistics gathering is enabled through:

- The **stats on** command.
- The defined configuration settings.

Then, you can use the **stats on** commands to modify the behavior for gathering and reporting statistics.

For example, statistics are enabled to create five statistics reports with each report generated each day. You can use the **stats on** command to change the frequency to every 12 hours. For this example, assume that the following command started statistics gathering:

```
pdadmin sec_master> server task PDWebPI-linuxweb.wasp.ibm.com stats on \
pdwebpi.stats 86400 5 file path=/tmp/stats.log
```

To modify the interval to 12 hours and create 10 reports, issue the following command:

```
pdadmin sec_master> server task PDWebPI-linuxweb.wasp.ibm.com stats on \
pdwebpi.stats 43200 10
```

Although the destination is not specified, the statistics infrastructure assumes any preexisting value. Entering the previous command does disable statistics from being written to the previously defined log file. However, if you specified a different destination, statistics reports would be written to the new destination only. You cannot use the **stats on** command to write statistics reports to more than one destination.

For more information about gathering statistics, see the *IBM Security Access Manager for Web Auditing Guide*.

Options

component
Specifies the component about which to gather or report statistics.

count
Specifies the number of reports to send to a log file. When you use the *count* option, you must specify the *interval* option. If you specify the *interval* option without the *count* option, the duration of reporting is indefinite.

After the count value is reached, reporting to a log file stops. Although statistics are no longer sent to a log file, the statistic component is still enabled. You can obtain reports from memory by using the **stats get** command.

destination
Specifies where the gathered statistics are written, where *destination* can be one of the following options:

file path=*file_name*
Specifies the fully qualified name of the log file.

log_agent
Specifies a directory where statistics information is gathered. For more information about logging events, see the *IBM Security Access Manager for Web Troubleshooting Guide*.

get
Displays the current report for a specific component or for all enabled components. If you specify the *component* option, displays the current report for that component; otherwise, displays the current report for all enabled components.

interval
Specifies the interval in seconds when statistics are sent from memory to a log file. When this option is specified, statistics are sent, by default, to the

server-specific log file designated by the `logcfg` entry in the server configuration file. You can specify another location by using the *`destination`* option. If an interval is not specified, statistics are not sent to a log file, but remain in memory.

Although statistics are not sent to a log file, the statistic component is still enabled. You can obtain reports from memory by using the **`stats get`** command.

`list` Lists all components that are available to gather and report statistics.

`off` Disables gathering of statistics for a specific component or for all components. If you specify the *`component`* option, disables gathering of statistics for that component; otherwise, disables gathering of statistics for all components.

`on` Enables gathering of statistics for a specific component. When you enable gathering of statistics, you can also set the reporting frequency, count, and log file.

`reset` Resets gathering of statistics for a specific component or for all enabled components. If you specify the *`component`* option, resets gathering of statistics for that component; otherwise, resets gathering of statistics for all components.

`server_name—host_name`
Specifies the name of the server or server instance. You must specify the server name in the exact format as it is shown in the output of the **`server list`** command.

For example, if the configured name of a single WebSEAL server on host `example.dallas.ibm.com` is `default`, the *`server_name`* would be `default-webseald` and the *`host_name`* would be `example.dallas.ibm.com`. For this example, the name of the server would be `default-webseald-example.dallas.ibm.com`.

If multiple server instances are configured on the same computer, for example:

- The host is `example.dallas.ibm.com`.
- The configured name of the WebSEAL server instance is `webseal2-webseald`.

Then,

- The *`server_name`* is `webseal2-webseald`.
- The *`host_name`* is `example.dallas.ibm.com`.
- The name of the server instance is `webseal2-webseald-example.dallas.ibm.com`.

`show` Lists all enabled components or indicates whether a specific component is enabled. If you specify the *`component`* option and the component is enabled, the output lists that component; otherwise, no output is displayed. If you do not specify the *`component`* option, the output lists all enabled components.

Return codes

0 The command completed successfully.

1 The command failed. See the *IBM Security Access Manager for Web Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

Examples

- The following example uses the **stats list** command to lists all enabled components on the ivacld-mogman.admogman.com authorization server:

  ```
  #pdadmin sec_master> server task ivacld-mogman.admogman.com stats list

  pd.ras.stats.monitor
  pd.log.EventPool.queue
  ```
- The following example:
  - Uses the **status on** command to enable gathering of statistics for the pd.log.EventPool.queue component on the ivacld-mogman.admogman.com authorization server.
  - Sets the reporting frequency to 30 days, that is, 2592000 seconds.
  - Sets the destination to the c:\myEPstats.log log file.

  ```
  #pdadmin sec_master> server task ivacld-mogman.admogman.com stats on \
  pd.log.EventPool.queue 2592000 file path=c:\myEPstats.log
  ```

See also

"server list" on page 411

Utilities

Table 69 lists the auditing utilities.

Table 69. Auditing utilities

Utility	Description
"amauditcfg"	Configures the Common Auditing Service client.

amauditcfg

Configures or unconfigures the Common Auditing Service client.

Syntax

amauditcfg –action config –srv_cfg_file *configuration_file* –audit_srv_url *url* –enable_ssl no –disk_cache_mode never

amauditcfg –action config –srv_cfg_file *configuration_file* –audit_srv_url *url* –enable_ssl no –disk_cache_mode {always|auto} –disk_cache_file *cache_file*

amauditcfg –action config –srv_cfg_file *configuration_file* –audit_srv_url *url* –enable_ssl yes –audit_key_file *key_file* –audit_stash_file *stash_file* –enable_pwd_auth no –disk_cache_mode never

amauditcfg –action config –srv_cfg_file *configuration_file* –audit_srv_url *url* –enable_ssl yes –audit_key_file *key_file* –audit_stash_file *stash_file* –enable_pwd_auth no –disk_cache_mode {always|auto} –disk_cache_file *cache_file*

amauditcfg –action –srv_cfg_file *configuration_file* –audit_srv_url *url* –enable_ssl yes config *key_file* –audit_stash_file *stash_file* –enable_pwd_auth yes –audit_id *audit_id* –audit_pwd audit_password –disk_cache_mode never

amauditcfg –action config –srv_cfg_file *configuration_file* –audit_srv_url *url* –enable_ssl yes –audit_key_file *key_file* –audit_stash_file *stash_file* –enable_pwd_auth yes –audit_id *audit_id* –audit_pwd *audit_password* –disk_cache_mode {always|auto}–disk_cache_file *cache_file* –temp_storage_full_timeout *number_of_seconds*

amauditcfg –action unconfig –srv_cfg_file *configuration_file*

amauditcfg –operations

amauditcfg –help [*options*]

amauditcfg –rspfile *response_file*

amauditcfg –usage

amauditcfg –?

Description

Use the **amauditcfg** utility to configure or unconfigure the Common Auditing Service client from the command line. The utility can be run in command-line mode or response file mode.

In command-line mode, all parameters must be specified from the command line.

In response file mode, the utility obtains the necessary parameters from the response file. You must manually create the response file, and the response file requires all parameters.

Parameters

–? Displays the syntax and an example for this utility.

–action {config|unconfig}
This parameter takes one of the following arguments:

config Configures the client.

unconfig
Unconfigures the client.

–audit_id *administrator_id*
Specifies the WebSphere administrator who has the EventSource role that is mapped to the CommonAuditService. This ID is authenticated through WebSphere by using HTTP basic authentication. This parameter is valid when the –enable_pwd_auth parameter is set to yes.

–audit_key_file *key_file*
Specifies the fully qualified name of the key file that is required for secure communication with the web service. This parameter is required when the –enable_ssl parameter is set to yes.

–audit_pwd *audit_id_password*
Specifies the password for the WebSphere administrator who has the EventSource role that is mapped to the CommonAuditService. This parameter is valid when the –enable_pwd_auth parameter is set to yes.

–audit_srv_url *url*

Specifies the URL of the web service. For secure communication, use the following URL:

```
https://hostname:9443/CommonAuditService/services/Emitter
```

For nonsecure communication, use the following URL:

```
http://hostname:9080/CommonAuditService/services/Emitter
```

–audit_stash_file *stash_file*

Specifies the fully qualified name of the stash file that is required for secure communication with the Common Audit web service. This parameter is required when the –enable_ssl parameter is set to yes.

–disk_cache_file *cache_file*

Specifies the fully qualified name of the disk cache file. This parameter is required when the –disk_cache_mode parameter is set to always or auto.

–disk_cache_mode {always|never|auto}

Specifies whether to enable disk caching, and, when enabled, indicates how to handle disk caching. The following values are valid:

always Indicates that audit events are always written directly to the disk cache.

never Indicates that audit events are written to the event queue. There is no disk cache.

auto Indicates that audit events are written to the event queue except when the server is down or the event queue is full. Under these conditions, the audit events are written to disk cache.

The default value is auto.

–temp_storage_full_timeout {0|-1| *number_of_seconds***}**

Specifies the number of seconds that the common auditing and reporting services client waits before cached events are discarded. The services client might discard cached events when the temporary disk cache storage is filled.

Valid values are -1, 0, number of seconds. A value of -1 indicates that cached events are not discarded.

A value of 0 indicates that cached events are discarded immediately. A specified number of seconds indicates that cached events are not discarded until the specified number of seconds passes. The default value is -1. (Optional)

This parameter takes effect only when –disk_cache_mode is set to always or auto.

–enable_pwd_auth {yes|no}

Specifies whether password authentication is used. Valid values are yes or no. This parameter is valid when the –enable_ssl parameter is set to yes. The default value is no. (Optional)

–enable_ssl {yes|no}

Specifies whether to enable SSL communication between the Common Audit client (the security server) and the Common Audit web service. Valid values are yes or no. The default value is no. (Optional)

–help [*parameters*]
Lists all parameters and their descriptions when specified without parameters. When one or more parameters are specified, lists the specified parameters and their descriptions.

–operations
Prints all the valid parameters.

–rspfile *response_file*
Specifies the fully qualified path and file name of the response file to use during silent configuration. A response file can be used for configuration. There is no default response file name. The response file contains stanzas and *key*=*value* pairs. For information about using response files, see the "Using response files" appendix in the *IBM Security Access Manager for Web Command Reference*. (Optional)

–srv_cfg_file *configuration_file*
Specifies the name of the configuration file that is associated with the security server (the Common Audit client). During configuration, entries are set to enable auditing. During an unconfiguration, the doAudit stanza entry is set to no in the [cars-client] stanza of the server-specific configuration file. For more information about entries in configuration files, see the *IBM Security Access Manager for Web Command Reference*.

–usage Displays the syntax and an example for this utility.

Availability

This utility is in one of the following default installation directories:

- On AIX, Linux, and Solaris operating systems:
  /opt/policyDirector/sbin/
- On Windows operating systems:
  c:\Program Files\Tivoli\PolicyDirector\sbin

When an installation directory other than the default is selected, this utility is in the /sbin directory under the installation directory (for example, *installation_directory*/sbin).

Return codes

0 The utility completed successfully.

1 The utility failed. When a utility fails, a description of the error and an error status code in hexadecimal format is provided (for example, 0x15c3a00c). See the *IBM Security Access Manager for Web Error Message Reference*. This reference provides a list of the Security Access Manager error messages by decimal or hexadecimal codes.

Examples

The following example configures an authorization server by using SSL and password authentication:

```
amauditcfg -action config \
-srv_cfg_file /opt/PolicyDirector/etc/pdaudit.pdacld.conf \
-srv_url https://hostname:9443/CommonAuditService/services/Emitter \
-enable_ssl yes -audit_key_file /certs/WSclient.kdb \
-audit_stash_file /certs/WSclient.sth -enable_pwd_auth yes \
-audit_id administrator_id -auditpwd password
```

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law :

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to

IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information in softcopy form, the photographs and color illustrations might not be displayed.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, Acrobat, PostScript and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Cell Broadband Engine and Cell/B.E. are trademarks of Sony Computer Entertainment, Inc., in the United States, other countries, or both and is used under license therefrom.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Index

Printed in USA

SC23-6511-02