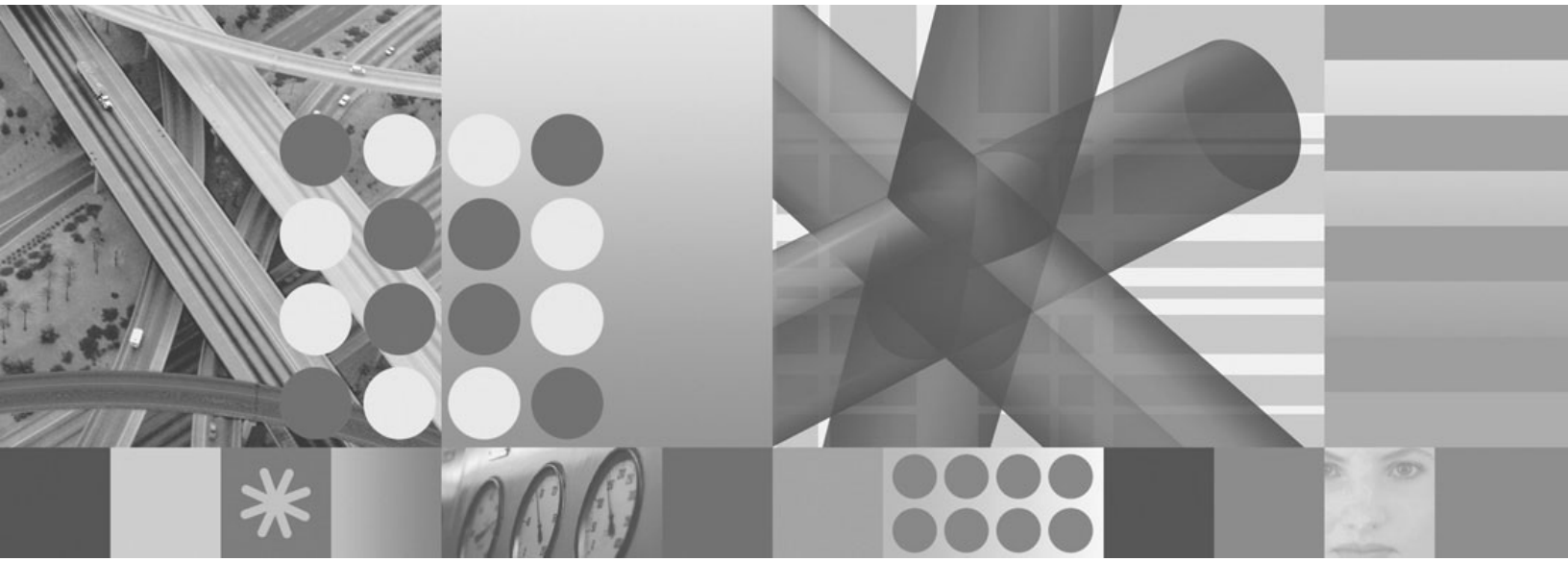
Tivoli® Federated Identity Manager
Version 6.2

IBM

Auditing Guide

GC32-2287-02

Tivoli® Federated Identity Manager
Version 6.2



Auditing Guide

GC32-2287-02

Note
Before using this information and the product it supports, read the information in "Notices" on page 197.

This edition applies to version 6, release 2, modification 0 of IBM Tivoli Federated Identity Manager (product number 5724-L73) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

About this publication vii
Intended audience vii
Publications vii
Accessibility ix
Tivoli technical training ix
Support information ix
Conventions used in this book x
Typeface conventions x
Operating system differences x

Chapter 1. Tivoli Federated Identity Manager auditing 1
Configuring Federated Identity Manager auditing settings to use an audit file 2
Configuring Federated Identity Manager auditing settings to use Common Audit Service 3
Selecting the events you want to audit 4

Chapter 2. IBM Tivoli Federated Identity Manager auditing events 7
Federated profiles - single sign-on (IBM_SECURITY_AUTHN) 10
Federated profiles - single logout (IBM_SECURITY_AUTHN_TERMINATE) 12
Federated profiles - name identifier management (IBM_SECURITY_FEDERATION) 14
Trust service modules - chains (IBM_SECURITY_TRUST) 18
Message security - encryption (IBM_SECURITY_ENCRYPTION) 22
Message security - signing (IBM_SECURITY_SIGNING) 24
Runtime configuration management (IBM_SECURITY_MGMT_POLICY) 25
Audit configuration management (IBM_SECURITY_MGMT_AUDIT) 60
Audit Provisioning (IBM_SECURITY_MGMT_PROVISIONING) . . . 62

Chapter 3. Overview of the Common Audit Service 75
Common Audit Service infrastructure 76
Scenario for collecting audit data 77

Chapter 4. Installing Common Audit Service 79
Installing prerequisite products 79
Installing the DB2 client on Windows systems . . 80
Installing the DB2 Administration Client on Linux and UNIX systems 81
Pre-installation checklist for all platforms 82
Interactive installation 84
Starting the installation wizard 84
Interactive installation using the GUI panels . . 85
Audit server installation options 86
Silent installation 88
Enabling language support 89
Installing language support packages 89
Customizing the XML store data definition language script 90

Chapter 5. Configuring the audit server 93
Pre-configuration checklist for all platforms . . . 93
Interactive configuration using the GUI panels . . 93
Common Audit Service configuration options . . . 95
Configuring JDBC resources in a clustered environment 96
Determining the type of cluster 97
Configuring JDBC resources against a heterogeneous cluster 97
Configuring JDBC resources against a homogeneous cluster 98
Configuring the compress property 98
Configuring a Web server for use in a clustered environment 98
Configuring a Web server that is installed on a cluster node 98
Configuring a Web server that is installed on a system outside the cluster 99
Enabling the IBM HTTP Server 101
Propagating the plug-in if the IBM HTTP Server is installed on a WebSphere Application Server node host 101
Propagating the plug-in if the IBM HTTP Server is installed on a remote host 102
Completing the Common Audit Service application to Web server mapping 102
Verifying your configuration settings for Common Audit Service 103
Verifying the configuration settings for the Common Audit Service application 103
Verifying the configuration settings for the Common Audit Service data source 105
Verifying the configuration settings for the Common Audit Service data store 106
Verifying the configuration settings for the Common Audit Service webservice component . 106
Deploying the Java stored procedure for an audit details report 108
Setting up to run the Java stored procedures on Linux 108
Setting the jdk_path parameter 109
Running ibmcarsddinst to deploy the Java stored procedure 109
Verifying the deployment of the IBMCARS_DD_REPORT Java stored procedure . 110
Verifying the deployment of the IBMCARS_EVENT_DETAIL Java stored procedure 110

Chapter 6. Upgrading the Common Audit Service audit server from earlier versions 113
Considerations for upgrading the Common Audit Service audit server 113
Upgrade goals 113
Preparing to upgrade the Common Audit Service audit server 114
Procedures for upgrading the Common Audit Service audit server from earlier versions 115
Installing Common Audit Service Version 6.1 when upgrading to use an existing database . . 115
Configuring Common Audit Service Version 6.1 to use an existing audit database 115
Post-upgrade steps: remove the old script, configure the clients to use the new port, uninstall the old version of the audit server . . 117

Chapter 7. Unconfiguring Common Audit Service 119

Chapter 8. Uninstalling Common Audit Service 121
Uninstallation checklist for all platforms 121
Interactive uninstallation 122
Starting the uninstallation wizard 122
Interactive uninstallation using the GUI windows 123
Silent uninstallation 123
Uninstalling language support packages 124

Chapter 9. Securing data flow in the operating environment 125
Securing Web service client events 125
Configuring the server 125
Securing the XML data store 134

Chapter 10. Running the server utilities 135
Preparing to run the server utilities 135
Running the staging utility command 136
Running the XML data store utilities 137
The ibmcars.properties file 139
Configuration parameters for the staging utility and XML data store utilities 142

Chapter 11. Federated Identity Manager reports 149
Required reporting setup tasks 150
Running Federated Identity Manager reports using the command line interface 153
Listing available reports using the command line interface 154
Sample response files for running a report using the command line interface 155
Running the out-of-box Federated Identity Manager reports using the console 156
Importing the Federated Identity Manager out-of-box reports into the Tivoli Common Reporting environment 156
Running the out-of-box Federated Identity Manager reports using the Tivoli Common Reporting console 158
Creating reports from existing designs 159
Federated Identity Manager reports 160
Creating custom reports 161
Requirements for creating new reporting tables 161
Steps to support custom reports 162
Working with the CARSShredder.conf configuration file 163
Sample custom report 168
Customizing reports for Tivoli Federated Identity Manager 169
Sample Tivoli Federated Identity Manager custom report 170
Federated Identity Manager DDL contents. . . 171
Creating a custom security event details report . . 172
Generating operational reports from archived data 173
Examples of XML-formatted security event data 173

Chapter 12. Archiving and restoring audit data 177
Archiving audit data 177
Restoring audit data 178

Chapter 13. Problem determination 179
Log files 179
Installation log files 179
Server utilities log files 180
WebSphere Application Server log files 180
Configuring log and trace settings 180
Considerations for setting the trace file path, trace level, and error file path during problem determination 181
Installation problems 181
Installer displays an error although the required DB2 software is installed 181
Silent installation does not fail when missing prerequisites. 182
Installation does not continue when the target WebSphere Application Server is stopped . . . 182
Installation does not continue when JVM version 1.5 is not found 183
Installation displays an error when WebSphere Application Server software is not found . . . 183
Debug tracing of installation or uninstallation of Common Audit Service 184
Common Audit Service configuration problems 184
(AIX) Audit Database configuration fails because operating systems SP2 is not applied . 184
Text displays incorrectly in some configuration panels 185
SOAP connection fails when a Common Audit Service Configuration Console is deployed in an eWAS environment 186
Problem deploying the Java stored procedure on a Linux platform 186

C client cannot communicate with the Common Audit Service server 186
Common Audit Service upgrade problems . . . 187
Common Audit Service uninstallation problems 187
Uninstall.bin not available 188
CarsConfigUtil.jar is not removed during a successful uninstallation of Common Audit Service 188
Failed uninstallation workarounds 188
Web service and emitter problems 191
Disregard message 0000004a 191
Web service emitter log displays event data . . 192
Server utility problems 192
Exception occurs while running the staging utility 192
java.lang.NullPointer exception occurs while running the staging utility 192
Remote database access failure occurs when using staging utility or XML data store utilities . 193
WebSphere Application Server problems 194
Problem sending events to Common Audit Service server when security is enabled . . . 194
Out of memory error 195

Notices 197
Trademarks 199

Index 201

About this publication

IBM Tivoli Federated Identity Manager Version 6.2 implements solutions for federated single sign-on, Web services security management, and provisioning that are based on open standards. IBM Tivoli Federated Identity Manager extends the authentication and authorization solutions provided by IBM Tivoli Access Manager to simplify the integration of multiple existing Web solutions.

This guide describes how to configure the IBM Tivoli Federated Identity Manager to audit security events. It also describes how to view IBM Tivoli Federated Identity Manager identity mapping and authorization events by creating customized reports. This guide also includes information about the IBM Tivoli Common Audit Service.

Intended audience

This guide is for network security architects, system administrators, network administrators, and system integrators. Readers of this book should have working knowledge of networking security issues, encryption technology, keys, and certificates. Readers should also be familiar with the implementation of authentication and authorization policies in a distributed environment.

Publications

Read the descriptions of the IBM® Tivoli® Federated Identity Manager library, the prerequisite publications, and the related publications to determine which publications you might find helpful. After you determine the publications you need, refer to the instructions for accessing publications online.

IBM Tivoli Federated Identity Manager library

The publications in the IBM Tivoli Federated Identity Manager library are:

- *IBM Tivoli Federated Identity Manager Quick Start Guide*

  Provides instructions for getting started with IBM Tivoli Federated Identity Manager.
- *IBM Tivoli Federated Identity Manager Installation and Configuration Guide*

  Provides instructions for installing and configuring IBM Tivoli Federated Identity Manager. Also provides instructions for configuring a demonstration application environment.
- *IBM Tivoli Federated Identity Manager for z/OS Program Directory*

  Provides instructions for installing IBM Tivoli Federated Identity Manager on z/OS.
- *IBM Tivoli Federated Identity Manager Administration Guide*

  Provides instructions for completing administration tasks that are required for all deployments.
- *IBM Tivoli Federated Identity Manager Single Sign-on Guide*

  Provides instructions for completing configuration tasks for federated single sign-on.
- *IBM Tivoli Federated Identity Manager Web Services Security Management Guide*

Provides instructions for completing configuration tasks for Web services security management.

- *IBM Tivoli Federated Identity Manager Auditing Guide*

  Provides instructions for auditing IBM Tivoli Federated Identity Manager events.
- *IBM Tivoli Federated Identity Manager Error Message Reference*

  Provides explanations for the IBM Tivoli Federated Identity Manager error messages.
- *IBM Tivoli Federated Identity Manager Problem Determination Guide*

  Provides troubleshooting information and instructions for problem solving.

You can obtain the publications from the IBM Tivoli Federated Identity Manager information center:

http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.tivoli.fim.doc_6.2/welcome.htm

Prerequisite publications

To use the information in this book effectively, you should have some knowledge of related software products, which you can obtain from the following sources:

- IBM Tivoli Access Manager for e-business Information Center:

  http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?toc=/com.ibm.itame.doc/toc.xml
- IBM WebSphere® Application Server Version 6.1 Information Center:

  http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp

  You can obtain PDF versions of the IBM WebSphere Application Server documentation at:

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

Related publications

You can obtain related publications from the following IBM Web sites:

- The IBM Tivoli Federated Identity Manager Business Gateway Information Center at http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.tivoli.fim_bg.doc_6.2/welcome.htm
- *Enterprise Security Architecture Using IBM Tivoli Security Solutions* (SG24-6014-04). This book is available in PDF (Portable Document Format) at http://www.redbooks.ibm.com/redbooks/pdfs/sg246014.pdf or in HTML (Hypertext Markup Language) at http://www.redbooks.ibm.com/redbooks/SG246014/.
- *Federated Identity Management and Web Services Security with IBM Tivoli Security Solutions* (SG24-6394-01). This book is available in PDF at http://www.redbooks.ibm.com/redbooks/pdfs/sg246394.pdf or in HTML at http://www.redbooks.ibm.com/redbooks/SG246394/
- The Tivoli Software Library provides a variety of Tivoli publications such as white papers, datasheets, demonstrations, redbooks, and announcement letters. The Tivoli Software Library is available on the Web at: http://publib.boulder.ibm.com/tividd/td/tdprodlist.html
- The *Tivoli Software Glossary* includes definitions for many of the technical terms related to Tivoli software. The *Tivoli Software Glossary* is available at http://publib.boulder.ibm.com/tividd/glossary/tivoliglossarymst.htm

Accessing publications online

The publications for this product are available online in Portable Document Format (PDF) or Hypertext Markup Language (HTML) format, or both in the Tivoli software library: http://publib.boulder.ibm.com/tividd/td/tdprodlist.html

To locate product publications in the library, click the first letter of the product name or scroll until you find the product name. Then, click the product name. Product publications include release notes, installation guides, user's guides, administrator's guides, and developer's references.

Note: To ensure proper printing of PDF publications, select the **Fit to page** check box in the Adobe Acrobat Print window (which is available when you click **File** → **Print**).

Ordering publications

You can order many Tivoli publications online at the following Web site:

http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi

You can also order by telephone by calling one of these numbers:

- In the United States: 800-879-2755
- In Canada: 800-426-4968

In other countries, see the Web site.

Accessibility

Accessibility features help a user who has 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 also can use the keyboard instead of the mouse to operate all features of the graphical user interface.

For additional information, see the "Accessibility" topic in the information center at http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.tivoli.fim.doc_6.2/welcome.htm.

Tivoli technical training

For Tivoli software training information, refer to the IBM Tivoli Education Web site: http://www.ibm.com/software/tivoli/education

Support information

If you have a problem with your IBM software, you want to resolve it quickly.

IBM provides the following ways for you to obtain the support you need:

Online

Go to the IBM Software Support site at http://www.ibm.com/software/support/probsub.html.

IBM Support Assistant
The IBM Support Assistant (ISA) is a free local software serviceability tool that helps you resolve questions and problems with IBM software products. The ISA provides quick access to support-related information and serviceability tools for problem determination. For information about the IBM Support Assistant, go to http://www.ibm.com/software/support/isa. For information about installing the ISA software and the product-specific plug-in, see the *IBM Tivoli Federated Identity Manager Installation and Configuration Guide*.

Problem Determination Guide
For more information about resolving problems, see the *IBM Tivoli Federated Identity Manager Problem Determination Guide*.

Conventions used in this book

This reference uses several conventions for special terms and actions and for operating system-dependent commands and paths.

Typeface conventions

The following typeface conventions are used in this guide.

Bold Lowercase commands or mixed case commands that are difficult to distinguish from surrounding text, keywords, parameters, options, names of Java™ classes, and objects are in **bold**.

Italic Variables, titles of publications, and special words or phrases that are emphasized are in *italic*.

`Monospace`
Code examples, command lines, screen output, file and directory names that are difficult to distinguish from surrounding text, system messages, text that the user must type, and values for arguments or command options are in `monospace`.

Operating system differences

This book uses the UNIX® convention for specifying environment variables and for directory notation. When you are using the Windows® command line, replace *$variable* with *%variable%* for environment variables and replace each forward slash (/) with a backslash (\) in directory paths. If you are using the bash shell on a Windows system, you can use the UNIX conventions.

Chapter 1. Tivoli Federated Identity Manager auditing

IBM Tivoli Federated Identity Manager uses the Common Audit Service (*previously named Common Auditing and Reporting Service*) client components to generate, format, and send audit events. All audit events are formatted as XML Common Base Events. You can configure Tivoli Federated Identity Manager to send audit events to a file-based log or to a Web service provided by Common Audit Service. The Common Audit Service audit server processes events that are sent to the Web service.

The Tivoli Federated Identity Manager administration console enables you to:

- Set audit logging on or off.
- Configure auditing settings, such as the location of the log files, file size, and number of log files.
- Specify which types of events to audit.
- Specify to send audit events to log files or to a Common Audit Service-provided Web service that enables more robust data management and reporting facilities.

When auditing configuration information is changed, a message is not automatically generated with the details of the changed auditing information. You must review the database of Common Audit Service events to determine the specific details of the changed information. When an auditing configuration change is made, an event is logged that contains *only* a listing of the current state of all auditing configuration settings. The specifically changed settings are not provided.

You can select to audit (or not audit) the following types of events:

Single sign-on
Contains information related to user login actions.

Single logout
Contains information related to user log-off actions.

Name identifier management
Contains information related to identity mapping actions (creation, deletion, update) that are associated with a user's consent to federate.

Message security
Contains information related to the signing and encryption of a message.

Trust service modules
Contains information related to trust server actions, such as, validation of a token, issuance of a token, mapping of an identity, or authorization of a Web service call.

Runtime configuration management
Contains information related to runtime management calls, such as, when federation, federation partner, or Web service partner changes are created, modified, or deleted.

Audit configuration management
Contains information related to auditing configuration, such as, when auditing is disabled, or when an auditing configuration setting is modified.

Audit provisioning
Contains information related to the provisioning of a cardspace to a user.

Configuring Federated Identity Manager auditing settings to use an audit file

This topic describes how to use the Federated Identity Manager administration console to store audit events in auditing files; instead of using Common Audit Service to store events. To enable auditing you must first configure the Tivoli Federated Identity Manage auditing settings.

You must have Tivoli Federated Identity Manager installed.

The Federated Identity Manage auditing service is configured using the Federated Identity Manager administration console, which runs as a plug-in to the Integrated Solutions Console.

1. Log in to the Integrated Solutions Console and expand **Tivoli Federated Identity Manager** in the navigation area.
2. In the Domain Management navigation area click **Auditing** to display the Audit Settings panel.
3. Select the **Enable audit** check box. You can select or deselect this check box at any time depending on whether you want to enable or disable auditing. The default setting is unchecked, that is, auditing is disabled.
4. Select the **Audit file** radio button to send the audit records to a local audit file. Each machine in the federation has its own audit file.
5. Specify the location for audit files in the **Audit log location** field:
   - You can accept the default, audit_location, which resolves to <WebSphere profile path>/logs/fimaudit.
   - Enter a relative path which resolves to <WebSphere profile path>/<your relative path>.
   - Enter an absolute path which is used as entered for the audit log file path.

   The specified audit log location serves as a path prefix for where audit files are created. The IBM Tivoli Federated Identity audit services generates an absolute path for runtime and management audit events as follows:

   For runtime audit events:
   <Your specified path prefix>/<domain name>/<cell name>/<node name>/<server name>

   For management audit events:
   <Your specified path prefix>/mgmt/<domain name>/<cell name>/<node name>/<server name>
6. In the **Maximum audit file size before rollover (MB)** field, specify the maximum audit file size. This value must be at least 1 MB. The default value is 10.
7. In the **Maximum number of audit files size before rollover** field, specify the maximum number of audit files that can be created before the oldest file is overwritten. This value must be at least 1. The default value is 100.
8. Do one of the following:
   - Click **OK** to save your changes and exit.
   - Click **Apply** to save your changes without exiting.
   - Click **Cancel** to exit without saving your changes.

You must stop and restart the WebSphere Application Server for your changes to take effect.

Configuring Federated Identity Manager auditing settings to use Common Audit Service

This topic describes how to use the Federated Identity Manager administration console to store audit events using Common Audit Service, instead of storing events directly into auditing files. To enable auditing you must first configure the Federated Identity Manage auditing settings.

You must have installed Tivoli Federated Identity Manager; and you must have installed and configured Common Audit Service before you can specify to direct events to the Common Audit Service.

Refer to Chapter 4, "Installing Common Audit Service," on page 79 and Chapter 5, "Configuring the audit server," on page 93 for information on installing and configuring Common Audit Service.

The Common Audit Service server root signer certificate must be imported to the Tivoli Federated Identity Manager keystore.

Tivoli Federated Identity Manage auditing service is configured using the Federated Identity Manager administration console, which runs as a plug-in to the Integrated Solutions Console.

1. Log in to the Integrated Solutions Console and expand **Tivoli Federated Identity Manager** in the navigation area.
2. In the Domain Management navigation area click **Auditing** to display the Audit Settings panel.
3. Select the **Enable audit** check box. You can select or deselect this check box at any time depending on whether you want to enable or disable auditing. The default setting is unchecked, that is, auditing is disabled.
4. Select the **Tivoli Common Auditing and Reporting Service** radio button to send the audit records to the Common Audit Service event server.
5. Type the address for the Common Audit Service Server in the **Web Service URL** field. The default value is `http://<audit_server_hostname>:9080/CommonAuditService/services/Emitter`. The server can be either a secure server (https) or an unsecure server (http).
6. Specify the location of the disk cache files in the **Disk cache location** field:
   - You can accept the default audit_location. This resolves to <WebSphere profile path>/logs/fimaudit.
   - Enter a relative path which resolves to <WebSphere profile path>/<your relative path>.
   - Enter an absolute path which is used as entered for the audit log file path.

   The disk cache location serves as a path prefix for where disk cache files are created. The Tivoli Federated Identity audit services generate an absolute path for runtime and management audit events as follows:

   For runtime audit events:
   Your_specified_path_prefix/domain_name/cell_name/node_name/server_name

   For management audit events:
   Your_specified_path_prefix/mgmt/*domain_name/cell_name/node_name/server_name*
7. Click **Web Service Security Settings**

- If you are using HTTP, go to Step 10 to select the appropriate authentication.

  Note: If you enter SSL settings, you receive an error message when you enable auditing.
- If you are using HTTPS continue with the next step to select your SSL settings.

8. From the **Keystore** drop-down menu select **DefaultKeyStore**:

   Note: You must import the Common Audit Service server root signer certificate to this keystore before performing this step. Common Audit Service supports only non-trusted keystores for security; you cannot select a trusted keystore, such as DefaultTrustedKeyStore, from the list of keystores. Use a non-trusted keystore, such as DefaultKeyStore.
9. Type in the password in the **Keystore Password** field.
10. Click **List Keys** and select the key you want to use.
11. Select the type of authentication:

Table 1.

Click:	Required actions:
None	No action required. This is the default setting, not to use authentication.
Use Basic Authentication	Type the user name and password in the **Basic Authentication Username** and **Basic Authentication Password** fields.

12. Do one of the following:
    - Click **Apply** to save your changes without exiting.
    - Click **Cancel** to exit without saving your changes.
13. Click **Load configuration changes to Tivoli Federated Identity Manager** in order for your changes to take effect. Click **Dismiss** to return to the previous window if you want to change a setting before loading your changes.

You must stop and restart the WebSphere Application Server for your changes to take effect.

Selecting the events you want to audit

Determine which events you want to audit. You can do this either before or after you have configured the Tivoli Federated Identity Manager auditing settings.

You must have installed Tivoli Federated Identity Manager and Common Audit Service to specify which events to audit.

IBM Tivoli Federated Identity Manager auditing service is configured using the Federated Identity Manager administration console, which runs as a plug-in to the Integrated Solutions Console.

1. Log in to the Integrated Solutions Console and expand **Tivoli Federated Identity Manager** in the navigation area.
2. Expand **Domain Management** and click **Auditing** to display the Audit Settings panel.
3. Under **Current Domain**, select the domain for which you want to select which events to audit.

4. Click **Audit Events** to display the Audit Events selections.
5. Select the events that you want to audit:
   - If you click **Federated Profiles**, the following types of events are also preselected:
     - Federated Profiles: Single-Sign-On
     - Federated Profiles: Single Logout
     - Federated Profiles: Name Identifier Management
     - Trust Service Modules
     - Message Security

     You can deselect any of these groups of event types. If you select the Federated Profiles group, at least one subgroup of events must also be selected. Up to two subgroups of Federated Profiles can be deselected.
   - If you click **Trust Service Modules**, all of the Trust Service Modules subgroups of events are automatically preselected and cannot be deselected.
   - The following groups of event types are independent of one another and can be selected or deselected:
     - Message Security
     - Audit Provisioning
     - Runtime Configuration Management
     - Audit Configuration Management
6. Do one of the following:
   - Click **Apply** to save your changes.
   - Click **Cancel** to exit without saving your changes.

You must stop and restart the WebSphere Application Server for your changes to take effect.

Chapter 2. IBM Tivoli Federated Identity Manager auditing events

This section lists the audit elements that are available for each Federated Identity Manager audit event type.

The IBM Tivoli Federated Identity Manager supports the following auditing events:

- Federated profiles - single sign-on
- Federated profiles - single logout
- Federated profiles - name identifier management
- Trust service modules - chains
- Message security - Signing
- Message security - Encryption
- Runtime configuration management
- Audit configuration management

This section describes the available elements for each event type.

Common elements for all events

The following elements are included with all security events:

- ContextDataElements
- SourceComponentIdelements
- Situation
- Outcome

ContextDataElements

The contextId value, specified on the type attribute, is included in the ContextDataElements element to correlate all events that are associated with a single transaction.

Table 2. Attributes and elements of the ContextDataElements element

Attribute	Value
name	Security Event Factory The XPath is: `CommonBaseEvent/contextDataElements/@name`
type	eventTrailId The XPath is: `CommonBaseEvent/contextDataElements/@type`
contextId	This element is a container element for the eventTrailId value; it does not have an XPath value.

Table 2. Attributes and elements of the ContextDataElements element (continued)

Attribute	Value
eventTrailId	The event trail identifier value, for example, FIM_116320b90110104ab7ce9df3453615a1+729829786 The XPath is: `CommonBaseEvent/contextDataElements/[@type='eventTrailId']/contextId`

Below are XML-formatted examples of CBE event headers containing entries for the ContextDataElements element. These entries illustrate how separate events are correlated for a single transaction.

```
<CommonBaseEvent
 creationTime="2007-01-31T20:59:57.625Z"
 extensionName="IBM_SECURITY_TRUST"
 globalInstanceId="CE4454A122E10AB044A1DBB16E020E1D80"
 sequenceNumber="1" version="1.0.1">
 <contextDataElements name="Security Event Factory"  type="eventTrailId">
  <contextId>FIM_79f4e4c801101db5aba48cd8e0212be7+656317861</contextId>
 </contextDataElements>
...
</CommonBaseEvent>
```

```
<CommonBaseEvent
 creationTime="2007-01-31T20:59:57.765Z"
 extensionName="IBM_SECURITY_TRUST"
 globalInstanceId="CE4454A122E10AB044A1DBB16E02213050"
 sequenceNumber="2" version="1.0.1">
 <contextDataElements name="Security Event Factory" type="eventTrailId">
  <contextId>FIM_79f4e4c801101db5aba48cd8e0212be7+656317861</contextId>
 </contextDataElements>
...
</CommonBaseEvent>
```

SourceComponentId element

The SourceComponentId is an identifier that represents the source that generates the event.

Table 3. Attributes for the SourceComponentId element

Attribute	Value
application	ITFIM#6.2 The XPath is: `CommonBaseEvent/sourceComponentId/ @application`
component	IBM Tivoli Federated Identity Manager The XPath is: `CommonBaseEvent/sourceComponentId/ @component`
componentIdType	ProductName The XPath is: `CommonBaseEvent/sourceComponentId/ @componentIdType`

Table 3. Attributes for the SourceComponentId element (continued)

Attribute	Value
componentType	http://www.ibm.com/namespaces/autonomic/Tivoli_componentTypes The XPath is: `CommonBaseEvent/sourceComponentId/@componentType`
executionEnvironment	<OS name>#<OS Architecture>#<OS.version> The XPath is: `CommonBaseEvent/sourceComponentId/@executionEnvironment`
location	<hostname> The XPath is: `CommonBaseEvent/extendedDataElements[@name='registryInfo']/children[@name='location']/values`
locationType	FQHostname The XPath is: `CommonBaseEvent/sourceComponentId/@locationType`
subComponent	<classname> The XPath is: `CommonBaseEvent/sourceComponentId/@subComponent`

Situation element

The Situation element describes the circumstance that caused the audit event.

Table 4. Attributes for the Situation element

Attribute	Value
categoryName	ReportSituation The XPath is: `CommonBaseEvent/situation/@categoryName`
reasoningScope	INTERNAL The XPath is: `CommonBaseEvent/situation/situationType/@reasoningScope`
reportCategory	SECURITY The XPath is: `CommonBaseEvent/situation/situationType/@reportCategory`

Outcome element

The Outcome element is the result of the action for which the security event is being generated.

Table 5. Attributes for the Outcome element

Attribute	Value
failureReason	The XPath is: `CommonBaseEvent/extendedDataElements [@name='outcome']/children [@name='failureReason']/values`
majorStatus	The XPath is: `CommonBaseEvent/extendedDataElements [@name='outcome']/children [@name='majorStatus']/values`
result	The XPath is: `CommonBaseEvent/extendedDataElements [@name='outcome']/children [@name='result']/values`

Note: Tivoli Federated Identity Manager does not utilize the ReporterComponentId field.

Federated profiles - single sign-on (IBM_SECURITY_AUTHN)

This event type is used when a user login is performed.

The following table lists the elements that can be displayed in the output of an IBM_SECURITY_AUTHN event.

Table 6. Elements for an IBM_SECURITY_AUTHN event

Element	Description
action	The operation that caused the authentication to occur. The XPath is: `CommonBaseEvent/extendedDataElements [@name='action']/values`
authnProvider	The provider of the authentication service. The default is webseal. The XPath is: `CommonBaseEvent/extendedDataElements [@name='authnProvider']/values`
authnScope	External - authentication is managed by an external entity. The session is established by the point-of-contact in response to externally provided credentials. The XPath is: `CommonBaseEvent/extendedDataElements [@name='authnScope']/values`
authnType	The type of authentication used - trustRelationship The XPath is: `CommonBaseEvent/extendedDataElements [@name='authnType']/values`

Table 6. Elements for an IBM_SECURITY_AUTHN event (continued)

Element	Description
partner	A group or organization participating in a federation. The XPath is: `CommonBaseEvent/extendedDataElements [@name='partner']/values`
progName	The name of the program that initiated the authentication, that is, the target Web address that user is trying to access. The XPath is: `CommonBaseEvent/extendedDataElements [@name='progName']/values`
tokenType	Defaults to xmlTokenType. The XPath is: `CommonBaseEvent/extendedDataElements [@name='tokenType']/values`
trustRelationship	The type of trustRelationship. The default is tokenAssertion. The XPath is: `CommonBaseEvent/extendedDataElements [@name='trustRelationship']/values`
userInfo.appUserName	Information about the user who is authenticating. The XPath is: `CommonBaseEvent/extendedDataElements [@name='userInfoList']/children[1]/children [@name='appUserName']/values`
xmlTokenType	The type of XML token. Supported types are: LibertyIDFFv11 LibertyIDFFv12 SAMLV10 SAMLV11 SAMLV20 WSFED The XPath is: `CommonBaseEvent/extendedDataElements [@name='xmlTokenType']/values`

Sample of a IBM_SECURITY_AUTHN event

The following is an example of an IBM_SECURITY_AUTHN event for a SAML2.0 SSO operation:

```
<CommonBaseEvent
 creationTime="2006-04-05T19:28:09.119Z"
 extensionName="IBM_SECURITY_AUTHN"
 globalInstanceId="CE11DAC4DA50628740DD4FC3FDDE8CF9A6"
 sequenceNumber="6"
 version="1.0.1">
 <extendedDataElements name="authnType" type="string">
  <values>trustRelationship</values>
 </extendedDataElements>
 <extendedDataElements name="partner" type="string">
```

```
 <values>https://ip/FIM/sps/saml20-ip/saml20</values>
</extendedDataElements>
<extendedDataElements name="authnScope" type="string">
 <values>external</values>
</extendedDataElements>
<extendedDataElements name="outcome" type="noValue">
 <children name="majorStatus" type="int"><values>0</values></children>
 <children name="result" type="string"><values>SUCCESSFUL</values></children>
</extendedDataElements>
<extendedDataElements name="tokenType" type="string">
 <values>xmltokentype</values>
</extendedDataElements>
<extendedDataElements name="xmlTokenType" type="string">
 <values>SAMLV20</values>
</extendedDataElements>
<extendedDataElements name="progName" type="string">
 <values>https://sp:444/</values>
</extendedDataElements>
<extendedDataElements name="trustRelationship" type="string">
 <values>tokenAssertion</values>
</extendedDataElements>
<extendedDataElements name="authnProvider" type="string">
 <values>webseal</values>
</extendedDataElements>
<extendedDataElements name="userInfoList" type="noValue">
 <children name="userInfo" type="noValue">
 <children name="appUserName" type="string">
 <values>me_elain</values></children>
 <children name="registryUserName" type="string">
 <values>Not Available</values></children></children>
</extendedDataElements>
<sourceComponentId
application="ITFIM#6.2"
 component="IBM Tivoli Federated Identity Manager"
 componentIdType="ProductName"
 executionEnvironment="Linux[x86]#2.4.21-4.EL"
 location="fimtest.au.ibm.com"
 locationType="FQHostname"
 subComponent=
 "com.tivoli.am.fim.saml20.protocol.actions.sso.SAML20LocalLoginAction"
 threadId="WebContainer : 0"
 componentType="http://www.ibm.com/namespaces/autonomic/Tivoli_componentTypes"/>
 <situation categoryName="ReportSituation">
  <situationType xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                 xsi:type="ReportSituation"
                 reasoningScope="INTERNAL"
                 reportCatagory="SECURITY"/>
 </situation>
</CommonBaseEvent>
```

Federated profiles - single logout (IBM_SECURITY_AUTHN_TERMINATE)

This audit event is generated when a user is logged out.

The following table lists the elements that can be displayed in the output of an IBM_SECURITY_AUTHN_TERMINATE event.

Table 7. Elements for an IBM_SECURITY_AUTHN_TERMINATE event

Element	Description
action	The operation that caused the termination of the authentication to occur - logout The XPath is: `CommonBaseEvent/extendedDataElements [@name='action']/values`
authnProvider	The provider of the authentication service. The default is webseal. The XPath is: `CommonBaseEvent/extendedDataElements [@name='authnProvider']/values`
authnType	The type of authentication used by the user - trustRelationship. The XPath is: `CommonBaseEvent/extendedDataElements [@name='authnType']/values`
terminateReason	The reason the session was terminated. For example the user logged out. The XPath is: `CommonBaseEvent/extendedDataElements [@name='terminateReason']/values`
userInfo.appUserName	Information about the user who is logging out. The XPath is: `CommonBaseEvent/extendedDataElements [@name='userInfoList']/children[1]/children [@name='appUserName']/values`

Sample of a IBM_SECURITY_AUTHN_TERMINATE event

The following is an example of a IBM_SECURITY_AUTHN_TERMINATE event:

```
<CommonBaseEvent
 creationTime="2006-04-19T18:13:15.916Z"
 extensionName="IBM_SECURITY_AUTHN_TERMINATE"
 globalInstanceId="CE11DACFD02C005F20EE33FA70BA750567"
 sequenceNumber="10"
 version="1.0.1">
 <extendedDataElements name="authnType" type="string">
  <values>trustRelationship</values>
 </extendedDataElements>
 <extendedDataElements name="action" type="string">
  <values>logout</values>
 </extendedDataElements>
 <extendedDataElements name="outcome" type="noValue">
  <children name="majorStatus" type="int">
   <values>0</values></children>
  <children name="result" type="string">
   <values>SUCCESSFUL</values></children>
 </extendedDataElements>
 <extendedDataElements name="terminateReason" type="string">
  <values>UserLoggedOut</values>
 </extendedDataElements>
```

```
 <extendedDataElements name="authnProvider" type="string">
  <values>webseal</values>
 </extendedDataElements>
 <extendedDataElements name="userInfoList" type="noValue">
  <children name="userInfo" type="noValue">
   <children name="appUserName" type="string">
    <values>me_elain</values></children>
   <children name="registryUserName" type="string">
    <values>Not Available</values></children>
  </children>
 </extendedDataElements>
<sourceComponentId
 application="ITFIM#6.2"
 component="IBM Tivoli Federated Identity Manager"
 componentIdType="ProductName"
 executionEnvironment="Linux[x86]#2.4.21-4.EL"
 location="fimtest.au.ibm.com"
 locationType="FQHostname"
 subComponent=
 "com.tivoli.am.fim.saml20.protocol.actions.SAML20LocalLogoutAction"
 threadId="WebContainer : 0"
 componentType="http://www.ibm.com/namespaces/autonomic/Tivoli_componentTypes"/>
 <situation categoryName="ReportSituation">
   <situationType xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                  xsi:type="ReportSituation"
                  reasoningScope="INTERNAL"
                  reportCatagory="SECURITY"/>
  </situation>
</CommonBaseEvent>
```

Federated profiles - name identifier management (IBM_SECURITY_FEDERATION)

This event type is generated when a federation event occurs.

A IBM_SECURITY_FEDERATION event is generated by the following actions:

- When a user identity mapping is created, that is, when a user is federated.
- When a user consents to federate.
- When a user identity mapping is deleted, that is, when a user is de-federated.
- When a user mapping is updated, for example, an RNI operation.

The following table lists the elements that can be displayed in the output of an IBM_SECURITY_FEDERATION event.

Table 8. Elements for an IBM_SECURITY_FEDERATION event

Element	Description
action	The type of federation action: CreateMapping ConsentToFederate DeleteMapping UpdateMapping The XPath is: `CommonBaseEvent/extendedDataElements [@name='action']/values`

Table 8. Elements for an IBM_SECURITY_FEDERATION event (continued)

Element	Description
messageAction	The type of action associated with the message. The XPath is: `CommonBaseEvent/extendedDataElements [@name='messageAction']/values`
partner	The partner that sends or receives the message. The XPath is: `CommonBaseEvent/extendedDataElements [@name='partner']/values`
profile	The profile within the federation. The XPath is: `CommonBaseEvent/extendedDataElements [@name='profile']/values`
protocolName	The type of federation protocol. The XPath is: `CommonBaseEvent/extendedDataElements [@name='protocolName']/values`
role	The role in which the audit generating component is acting. The XPath is: `CommonBaseEvent/extendedDataElements [@name='role']/values`
userInfo.appUserName	Information about the user who is performing this operation. The XPath is: `CommonBaseEvent/extendedDataElements [@name='userInfoList']/children[1]/children[ @name='appUserName']/values`

Action-dependent additional attributes

Depending on the type of federation event action, the following attributes are available:

Table 9. IBM_SECURITY_FEDERATION action-dependent additional attributes

Action	Additional attributes	Description

CreateMapping	selfAlias	If a self alias is set for the user, then this attribute displays that value.
		
		The XPath for the attribute name is:
		`CommonBaseEvent/extendedDataElements [@name='attributes']/ children [@name='attribute']/children [@name='name']/values [contains(.,'consentToFederate')] /../../children [@name='value']/values`
		
		The XPath for the attribute value is:
		`CommonBaseEvent/extendedDataElements [@name='attributes']/ children [@name='attribute']/children [@name='name']/values [contains(.,'selfAlias')] /../../children [@name='value']/values`

	partnerAlias	If a partner alias is set for the user, then this attribute displays that value.
		
		The XPath for the attribute name is:
		`CommonBaseEvent/extendedDataElements [@name='attributes']/ children [@name='attribute']/children [@name='name']/values [contains(.,'partnerAlias')]`
		
		The XPath for the attribute value is:
		`CommonBaseEvent/extendedDataElements [@name='attributes']/ children [@name='attribute']/children [@name='name']/values [contains(.,'partnerAlias')] /../../children [@name='value']/values`

ConsentToFederate	ConsentToFederate	This attribute specifies whether the user consented to federate. This event applies to Liberty and SAML20 protocol flows.
		
		The XPath for the attribute name is:
		`CommonBaseEvent/extendedDataElements [@name='attributes']/ children [@name='attribute'/children [@name='name']/values [contains(.,'consentToFederate')]`
		
		The XPath for the attribute value is:
		`CommonBaseEvent/extendedDataElements [@name='attributes']/ children [@name='attribute']/children [@name='name']/values [contains(.,'consentToFederate')] /../../children [@name='value']/values`

DeleteMapping	None	None

Table 9. IBM_SECURITY_FEDERATION action-dependent additional attributes (continued)

Action	Additional attributes	Description

UpdateMapping	selfAlias	If a self alias is set for the user, then this attribute displays the updated value.
		
		The XPath for the attribute name is:
		`CommonBaseEvent/extendedDataElements [@name='attributes']/ children [@name='attribute']/children [@name='name']/values [contains(.,'consentToFederate')] /../../children [@name='value']/values`
		
		The XPath for the attribute value is:
		`CommonBaseEvent/extendedDataElements [@name='attributes']/ children [@name='attribute']/children [@name='name']/values [contains(.,'selfAlias')] /../../children [@name='value']/values`

	partnerAlias	If a partner alias is set for the user, then this attribute displays the updated value.
		
		The XPath for the attribute name is:
		`CommonBaseEvent/extendedDataElements [@name='attributes']/ children [@name='attribute']/children [@name='name']/values [contains(.,'partnerAlias')]`
		
		The XPath for the attribute value is:
		`CommonBaseEvent/extendedDataElements [@name='attributes']/ children [@name='attribute']/children [@name='name']/values [contains(.,'partnerAlias')] /../../children [@name='value']/values`

Sample of a IBM_SECURITY_FEDERATION event

The following is an example of a IBM_SECURITY_FEDERATION event:

```
<CommonBaseEvent
 creationTime="2006-04-05T20:09:41.983Z"
 extensionName="IBM_SECURITY_FEDERATION"
 globalInstanceId="CE11DAC4E01E4BBF50E69681063F1AA1AF"
 sequenceNumber="7"
 version="1.0.1">
 <extendedDataElements name="action" type="string">
  <values>DeleteMapping</values>
 </extendedDataElements>
 <extendedDataElements name="partner" type="string">
  <values>https://sp:444/FIM/sps/saml20-sp/saml20</values>
 </extendedDataElements>
 <extendedDataElements name="relayState" type="string">
  <values>Not Available</values>
 </extendedDataElements>
 <extendedDataElements name="outcome" type="noValue">
```

```
 <children name="majorStatus" type="int"><values>0</values></children>
 <children name="result" type="string"><values>SUCCESSFUL</values></children>
</extendedDataElements>
<extendedDataElements name="clientInfo" type="boolean">
 <values>false</values>
</extendedDataElements>
<extendedDataElements name="role" type="string">
 <values>IP</values>
</extendedDataElements>
<extendedDataElements name="messageAction" type="string">
 <values>RECEIVED</values>
</extendedDataElements>
<extendedDataElements name="profile" type="string">
 <values>urn:oasis:names:tc:SAML:2.0:profiles:SSO:nameid-mgmt</values>
</extendedDataElements>
<extendedDataElements name="protocolName" type="string">
 <values>urn:oasis:names:tc:SAML:2.0:protocol</values>
</extendedDataElements>
<extendedDataElements name="userInfoList" type="noValue">
 <children name="userInfo" type="noValue">
  <children name="appUserName" type="string"><values>Elain</values></children>
  <children name="registryUserName" type="string">
   <values>Not Available</values></children>
 </children>
</extendedDataElements>
<sourceComponentId
 application="ITFIM#6.2"
 component="IBM Tivoli Federated Identity Manager"
 componentIdType="ProductName"
 executionEnvironment="Linux[x86]#2.4.21-4.EL"
 location="fimtest.au.ibm.com"
 locationType="FQHostname"
 subComponent=
"com.tivoli.am.fim.saml20.protocol.actions.nimgmt.
 SAML20ProcessManageNameIDMessageAction"
 threadId="WebContainer : 1"
 componentType="http://www.ibm.com/namespaces/autonomic/Tivoli_componentTypes"/>
 <situation categoryName="ReportSituation">
   <situationType xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                  xsi:type="ReportSituation"
                  reasoningScope="INTERNAL"
                  reportCatagory="SECURITY"/>
  </situation>
</CommonBaseEvent>
```

Trust service modules - chains (IBM_SECURITY_TRUST)

This event type is generated by the trust server when it validates a token, issues a token, maps an identity, or authorizes a Web service call.

Trust events are generated by the trust server when it validates a token, issues a token, maps an identity, and authorizes a Web service call.

The following table lists the elements that can be displayed in the output of an IBM_SECURITY_TRUST event.

Table 10. Elements for an IBM_SECURITY_TRUST event

Element	Description
accessDecision	For the authorization module, it is the result of the authorization decision. This element is filled in only when the action is authorized. The XPath is: `CommonBaseEvent/extendedDataElements [@name='accessDecision']/values`
action	The action being performed. Possible actions are: authorize issue map validate The XPath is: `CommonBaseEvent/extendedDataElements [@name='action']/values`
appliesTo	The destination or resource to which the request or token applies. The XPath is: `CommonBaseEvent/extendedDataElements [@name='appliesTo']/values`
issuer	The party responsible for issuing the token. The XPath is: `CommonBaseEvent/extendedDataElements [@name='issuer']/values`
moduleName	The module within the STS module chain that is being acted upon. The XPath is: `CommonBaseEvent/extendedDataElements [@name='moduleName']/values`
ruleName	The rule name used for the mapping module. This element is filled in only when specified action is map. The XPath is: `CommonBaseEvent/extendedDataElements [@name='ruleName']/values`
token	The incoming token being acted upon. Only the first 1024 characters of the token are set. When the action is map, this element represents the incoming principal. The XPath is: `CommonBaseEvent/extendedDataElements [@name='token']/values`
tokenInfo	The internal representation of the user information *after* changes are made by the module. Only the first 1024 characters of the token are set. When action is map, this element represents the outgoing principal. When the action is authorize, this element represents the principal for whom the access decision was made. The XPath is: `CommonBaseEvent/extendedDataElements [@name='tokenInfo']/values`

Table 10. Elements for an IBM_SECURITY_TRUST event (continued)

Element	Description
tokenType	The type of token the module is using. The XPath is: `CommonBaseEvent/extendedDataElements [@name='tokenType']/values`

Samples of a IBM_SECURITY_TRUST events

The following examples show two events generated by a Trust request. The first event is a validation action on the received token, and the second event is a mapping action that is generated when the token is mapped to the STSUniversalUser.

```
<CommonBaseEvent
 creationTime="2007-01-31T20:59:57.625Z"
 extensionName="IBM_SECURITY_TRUST"
 globalInstanceId="CE4454A122E10AB044A1DBB16E020E1D80"
 sequenceNumber="1"
 version="1.0.1">
 <contextDataElements
  name="Security Event Factory"
  type="eventTrailId">
  <contextId>FIM_79f4e4c801101db5aba48cd8e0212be7+656317861</contextId>
 </contextDataElements>
 <extendedDataElements name="token" type="string">
  <values>
   <saml:Assertion
    xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:wss="http://docs.oasis-open.org/wss/2004/01/
   oasis-200401-wss-wssecurity-secext-1.0.xsd"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    AssertionID="Assertion-uuid79b3e60c-0110-1536-9aa6-9a781eece361"
    IssueInstant="2007-01-31T19:48:57Z"
    Issuer="http://issuer/saml11-2"
    MajorVersion="1"
    MinorVersion="1">
   <saml:Conditions
    NotBefore="2007-01-31T19:38:57Z"
    NotOnOrAfter="2007-02-01T12:28:57Z">
    <saml:AudienceRestrictionCondition>
     <saml:Audience>http://appliesto/saml11-2</saml:Audience>
    </saml:AudienceRestrictionCondition>
   </saml:Conditions>
   <saml:AuthenticationStatement
    AuthenticationInstant="2007-01-31T19:48:57Z"
    AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password">
   <saml:Subject>
  </values>
 </extendedDataElements>
 <extendedDataElements name="issuer" type="string">
  <values>http://issuer/saml11</values>
 </extendedDataElements>
 <extendedDataElements name="moduleName" type="string">
  <values>com.tivoli.am.fim.trustserver.sts.modules.SAMLTokenSTSModuleBase</values>
 </extendedDataElements>
 <extendedDataElements name="ruleName" type="string">
  <values>Not Available</values>
 </extendedDataElements>
 <extendedDataElements name="tokenInfo" type="string">
 <values>
  <?xml version="1.0" encoding="UTF-8"?>
```

```
 <stsuuser:STSUniversalUser xmlns:stsuuser="urn:ibm:names:ITFIM:1.0:stsuuser">
 <stsuuser:Principal>
  <stsuuser:Attribute name="NameQualifier"
  type="urn:oasis:names:tc:SAML:1.0:assertion"/>
   <stsuuser:Attribute name="name"
  type="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">
   <stsuuser:Value>BigDummy</stsuuser:Value>
  </stsuuser:Attribute>
 </stsuuser:Principal>
 <stsuuser:AttributeList>
  <stsuuser:Attribute name="ssn" type="http://example.com/federation/v1/namevalue">
   <stsuuser:Value>
    <stsuuser:Value xmlns:stsuuser=
  "urn:ibm:names:ITFIM:1.0:stsuuser">555-55-555</stsuuser:Value>
   </stsuuser:Value>
  </stsuuser:Attribute>
  <stsuuser:Attribute name="MinorVersion"
  type="urn:oasis:names:tc:SAML:1.0:assertion">
   <stsuuser:Value>1</stsuuser:Value>
  </stsuuser:Attribute>
  <stsuuser:Attribute name="email" type="http://example.com/federation/v1/email">
   <stsuuser:Value>elain@hotmail.com</stsuuser:Value>
  </stsuuser:Attribute>
  <stsuuser:Attribute </values>
</extendedDataElements>
<extendedDataElements name="appliesTo" type="string">
 <values>http://appliesto/saml11</values>
</extendedDataElements>
<extendedDataElements name="action" type="string">
 <values>Validate</values>
</extendedDataElements>
<extendedDataElements="">
 <values>Not Available</values>
<extendedDataElements>
</extendedDataElements>
<extendedDataElements name="outcome" type="noValue">
 <children="">
  <values>SUCCESSFUL</values>
 <children></children>
 <children name="majorStatus" type="int">
  <values>0</values>
 </children>
</extendedDataElements>
<sourceComponentId
 application="ITFIM#6.1"
 component="IBM Tivoli Federated Identity Manager"
 componentIdType="ProductName"
 executionEnvironment="Windows XP[x86]#5.1 build 2600 Service Pack 2"
 location="jarocke.austin.ibm.com"
 locationType="FQHostname"
 subComponent="com.tivoli.am.fim.trustserver.sts.modules.SAMLTokenSTSModuleBase"
 threadId="WebContainer : 0"
 componentType="http://www.ibm.com/namespaces/autonomic/Tivoli_componentTypes"/>
<situation categoryName="ReportSituation">
 <situationType
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:type="ReportSituation"
  reasoningScope="INTERNAL"
  reportCategory="SECURITY"/>
</situation>
</CommonBaseEvent>

<CommonBaseEvent
creationTime="2007-01-31T20:59:57.765Z"
extensionName="IBM_SECURITY_TRUST"
globalInstanceId="CE4454A122E10AB044A1DBB16E02213050"
sequenceNumber="2" version="1.0.1">
<contextDataElements name="Security Event Factory" type="eventTrailId">
 <contextId>FIM_79f4e4c801101db5aba48cd8e0212be7+656317861</contextId>
</contextDataElements>
<extendedDataElements
 name="token"
```

```
 type="string">
 <values>BigDummy [ Attribute 1 NameQualifier urn:oasis:names:tc:SAML:1.0:
        assertion ]
                  [ Attribute 2 name urn:oasis:names:tc:SAML:1.1:nameid-format:
        emailAddress
                  [ value 1 BigDummy ] ]
 </values>
</extendedDataElements>
<extendedDataElements name="issuer" type="string">
 <values>http://issuer/saml11</values>
</extendedDataElements>
<extendedDataElements name="moduleName" type="string">
 <values>com.tivoli.am.fim.trustserver.sts.modules.STSMapDefault</values>
</extendedDataElements>
<extendedDataElements name="ruleName" type="string">
 <values>rule1</values>
</extendedDataElements>
<extendedDataElements name="tokenInfo" type="string">
 <values>me_guest [ Attribute 1 NameQualifier urn:oasis:names:tc:SAML:1.0:
        assertion ]
                  [ Attribute 2 name urn:ibm:names:ITFIM:5.1:accessmanager
                  [ value 1 me_guest ] ]
 </values>
</extendedDataElements>
<extendedDataElements name="appliesTo" type="string">
 <values>http://appliesto/saml11</values>
</extendedDataElements>
<extendedDataElements name="action" type="string">
 <values>Map</values>
</extendedDataElements>
<extendedDataElements name="tokenType" type="string">
 <values>Not Available</values>
</extendedDataElements>
<extendedDataElements name="outcome" type="noValue">
 <children name="result" type="string">
  <values>SUCCESSFUL</values>
 </children>
 <children name="majorStatus" type="int">
  <values>0</values>
 </children>
</extendedDataElements>
<sourceComponentId
 application="ITFIM#6.2"
 component="IBM Tivoli Federated Identity Manager"
 componentIdType="ProductName"
 executionEnvironment="Windows XP[x86]#5.1 build 2600 Service Pack 2"
 location="jarocke.austin.ibm.com"
 locationType="FQHostname"
 subComponent="com.tivoli.am.fim.trustserver.sts.modules.STSMapDefault"
 threadId="WebContainer : 0"
 componentType="http://www.ibm.com/namespaces/autonomic/Tivoli_componentTypes"/>
<situation categoryName="ReportSituation">
 <situationType xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:type="ReportSituation"
  reasoningScope="INTERNAL"
  reportCategory="SECURITY"/>
 </situation>
</CommonBaseEvent>
```

Message security - encryption (IBM_SECURITY_ENCRYPTION)

This event is generated whenever data is encrypted.

The following table lists the elements that can be displayed in the output of an IBM_SECURITY_ENCRYPTION event.

Table 11. Elements for an IBM_SECURITY_ENCRYPTION event

Element	Description
action	The operation that is being performed, either encryption or decryption. The XPath is: `CommonBaseEvent/extendedDataElements [@name='action']/values`
keyInfo	The key used to perform the action. The XPath is: `CommonBaseEvent/extendedDataElements [@name='keyInfo']/values`
msgInfo	The pertinent parts of the SOAP messages. The XPath is: `CommonBaseEvent/extendedDataElements [@name='msgInfo']/values`

Sample of a IBM_SECURITY_ENCRYPTION event

The following is an example of a IBM_SECURITY_ENCRYPTION event:

```
<CommonBaseEvent
 creationTime="2006-04-18T18:02:09.824Z"
 extensionName="IBM_SECURITY_ENCRYPTION"
 globalInstanceId="CE11DECF0574918190EA65C3F4A1F4E637"
 sequenceNumber="23"
 version="1.0.1">
 <extendedDataElements name="keyInfo" type="string">
  <values>DefaultKeyStore_testkey</values>
 </extendedDataElements>
 <extendedDataElements name="action" type="string">
  <values>Encrypt</values>
 </extendedDataElements>
 <extendedDataElements name="outcome" type="noValue">
  <children name="majorStatus" type="int">
   <values>0</values></children>
  <children name="result" type="string">
   <values>SUCCESSFUL</values></children>
 </extendedDataElements>
 <extendedDataElements name="msgInfo" type="string">
  <values>[{urn:oasis:names:tc:SAML:2.0:protocol}Response[0]
   {http://www.w3.org/2000/09/xmldsig#}Signature[0]]</values>
 </extendedDataElements>
 <extendedDataElements name="userInfo" type="noValue">
  <children name="appUserName" type="string">
   <values>Not Available</values></children>
  <children name="registryUserName" type="string">
   <values>Not Available</values></children>
 </extendedDataElements>
<sourceComponentId
 application="ITFIM#6.2"
 component="IBM Tivoli Federated Identity Manager"
 componentIdType="ProductName"
 executionEnvironment="Linux[x86]#2.4.21-4.EL"
 location="fimtest.au.ibm.com"
 locationType="FQHostname"
 subComponent=
```

```
"com.tivoli.am.fim.kess.service.jks.worker.impl.KessServiceJksWorkerImpl"
 threadId="WebContainer : 1"
 componentType="http://www.ibm.com/namespaces/autonomic/Tivoli_componentTypes"/>
 <situation categoryName="ReportSituation">
   <situationType xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                  xsi:type="ReportSituation"
                  reasoningScope="INTERNAL"
                  reportCatagory="SECURITY"/>
  </situation>
</CommonBaseEvent>
```

Message security - signing (IBM_SECURITY_SIGNING)

This auditing event is generated when the IBM Tivoli Federated Identity Manager signs data.

The following table lists the elements that can be displayed in the output of an IBM_SECURITY_SIGNING event.

Table 12. Elements for an IBM_SECURITY_SIGNING event

Element	Description
action	The operation that is being performed, either signature or validation. The XPath is: `CommonBaseEvent/extendedDataElements [@name='action']/values`
keyInfo	The key identifier used to perform the action. The XPath is: `CommonBaseEvent/extendedDataElements[ @name='keyInfo']/values`
msgInfo	The pertinent parts of the SOAP messages. The XPath is: `CommonBaseEvent/extendedDataElements [@name='msgInfo']/values`

Sample of a IBM_SECURITY_SIGNING event

The following is an example of a IBM_SECURITY_SIGNING event:

```
<CommonBaseEvent
 creationTime="2006-04-18T18:02:09.824Z"
 extensionName="IBM_SECURITY_SIGNING"
 globalInstanceId="CE11DACF0574918190EA65C3F4A1F4E637"
 sequenceNumber="2"
 version="1.0.1">
 <extendedDataElements name="keyInfo" type="string">
  <values>DefaultKeyStore_testkey</values>
 </extendedDataElements>
 <extendedDataElements name="action" type="string">
  <values>Signature</values>
 </extendedDataElements>
 <extendedDataElements name="outcome" type="noValue">
  <children name="majorStatus" type="int">
   <values>0</values></children>
  <children name="result" type="string">
   <values>SUCCESSFUL</values></children>
 </extendedDataElements>
 <extendedDataElements name="tokenType" type="string">
```

```
  <values>x509</values>
 </extendedDataElements>
 <extendedDataElements name="msgInfo" type="string">
  <values>[{urn:oasis:names:tc:SAML:2.0:protocol}Response[0],
   {http://www.w3.org/2000/09/xmldsig#}Signature[0]]</values>
 </extendedDataElements>
 <extendedDataElements name="userInfo" type="noValue">
  <children name="appUserName" type="string">
   <values>Not Available</values></children>
  <children name="registryUserName" type="string">
   <values>Not Available</values></children>
 </extendedDataElements>
<sourceComponentId
 application="ITFIM#6.2"
 component="IBM Tivoli Federated Identity Manager"
 componentIdType="ProductName"
 executionEnvironment="Linux[x86]#2.4.21-4.EL"
 location="fimtest.au.ibm.com"
 locationType="FQHostname"
 subComponent=
"com.tivoli.am.fim.kess.service.jks.worker.impl.KessServiceJksWorkerImpl"
 threadId="WebContainer : 1"
 componentType="http://www.ibm.com/namespaces/autonomic/Tivoli_componentTypes"/>
 <situation categoryName="ReportSituation">
   <situationType xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                  xsi:type="ReportSituation"
                  reasoningScope="INTERNAL"
                  reportCatagory="SECURITY"/>
  </situation>
</CommonBaseEvent>
```

Runtime configuration management (IBM_SECURITY_MGMT_POLICY)

This event type generated by IBM Tivoli Federation Identity Manager management calls.

A IBM_SECURITY_MGMT_POLICY event is generated by the following actions:

- When a new Federation is created.
- When an existing federation is modified.
- When a federation is deleted.
- When a partner is added to a federation.
- When a partner is deleted from a federation.
- When the properties of a partner are modified.
- When a Web Service partner is created.
- When a Web Service partner is modified.

The following table lists the elements that can be displayed in the output of an IBM_SECURITY_MGMT_POLICY event.

Table 13. Elements for an IBM_SECURITY_MGMT_POLICY event

Element	Description
action	The type of operation that is being performed. The supported operations are: create delete modify The XPath is: `CommonBaseEvent/extendedDataElements [@name='action']/values`
mgmtInfo.command	Information about the management operation. The supported management operations are: CreateFederation ModifyFederation DeleteFederation CreateFederationPartner ModifyFederationPartner DeleteFederationPartner CreateWebServicePartner ModifyWebServicePartner **Note:** Modifying or deleting a Web service partner generates a ModifyWebServicePartner operation. The XPath is: `CommonBaseEvent/extendedDataElements [@name='mgmtInfo']/children [@name='command']/values`

policyInfo.attributes	The different attributes for this policyInfo object. See the tables in "Attributes determined by policy profile type" on page 27 for attributes that might be present in the event. Each attribute consists of a name and a value.
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values`
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='value']/values`

policyInfo.name	The name of the federation, the name of the partner, or the name of the Web service partner. The XPath is: `CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='name']/values`

Table 13. Elements for an IBM_SECURITY_MGMT_POLICY event (continued)

Element	Description
policyInfo.type	Information about the policy object. The type can be either federation or partner. The XPath is: `CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='type']/values`
userInfo.appUserName	Information about the user who is performing this operation. The XPath is: `CommonBaseEvent/extendedDataElements [@name='userInfoList']/children[1]/children [@name='appUserName']/values`

Attributes determined by policy profile type

Depending on the type of profile used, policyInfo contains different attributes. These attributes can be shredded or extracted for custom reports. See "Sample Tivoli Federated Identity Manager custom report" on page 170 for additional information.

Note: Different partner attributes are specified as *partner id_attribute name*, where *partner id* is the uuid assigned to a partner and *attribute name* is an attribute from the following tables.

Shredding and staging attributes

This example shows how the data can be shredded by using the contains keyword. It requires an XPath for each attribute.

To stage the following name-value pairs for FederationName, FederationId and SAML1.SigningKey Identifier from the attributes fields of a policyInfo, use the following XPaths:

Table 14. XPaths for hredding and staging attributes

Field	XPath
policyInfo attributes FederationId	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'FederationId')]`
policyInfo attributes FederationId value	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'FederationId')] /../../children [@name='value']/values`

Table 14. XPaths for hredding and staging attributes (continued)

Field	XPath
policyInfo attributes FederationName	CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'FederationName')]
policyInfo attributes FederationName value	CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'FederationName')] /../../children [@name='value']/values
policyInfo attribue SAML1.SigningKeyIdentifier	CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML1.SigningKeyIdentifier')]
policyInfo attribue SAML1.SigningKeyIdentifier value	CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML1.SigningKeyIdentifier')] /../../children [@name='value']/values

SAML10 and SAML11 attributes

The following table lists the SAML10 and SAML11 configuration attributes that are audited in profiles for service providers and identity providers.

Table 15. Policy information attributes for SAML10 and SAML11 configuration profiles.

Common attributes for service providers and identity providers	**Descriptions**

SAML1.SoapRequestClientBasicAuth	The indicator for whether client basic authentication is used for the SOAP backchannels.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML1.SoapRequestClientBasicAuth')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML1.SoapRequestClientBasicAuth')] /../../children [@name='value']/values`

SAML1.SoapRequestClientCertAuth	The indicator for whether client certificate authentication is used for the SOAP backchannels.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML1.SoapRequestClientCertAuth')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML1.SoapRequestClientCertAuth')] /../../children [@name='value']/values`

Table 15. Policy information attributes for SAML10 and SAML11 configuration profiles. (continued)

Common attributes for service providers and identity providers	**Descriptions**

SAML1.SoapRequestServerCertAuth	The indicator for whether server certificate authentication is used for the SOAP backchannels.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML1.SoapRequestServerCertAuth')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML1.SoapRequestServerCertAuth')] /../../children [@name='value']/values`

SAML1. SoapRequestServerCertAuthKeyIdentifier	The identifier for the key used when using server certificate authentication.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML1. SoapRequestServerCertAuthKeyIdentifier')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML1. SoapRequestServerCertAuthKeyIdentifier')] /../../children [@name='value']/values`

Table 15. Policy information attributes for SAML10 and SAML11 configuration profiles. (continued)

Common attributes for service providers and identity providers	**Descriptions**

SAML1.
SoapRequestClientCertAuthKeyIdentifier
	The identifier for the key used when using client certificate authentication.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements`
	`[@name='policyInfo']/children`
	`[@name='attributes']/children`
	`[@name='attribute']/children`
	`[@name='name']/values`
	`[contains`
	`(.,'SAML1.`
	`SoapRequestClientCertAuthKeyIdentifier')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements`
	`[@name='policyInfo']/children`
	`[@name='attributes']/children`
	`[@name='attribute']/children`
	`[@name='name']/values`
	`[contains`
	`(.,'SAML1.`
	`SoapRequestClientCertAuthKeyIdentifier')]`
	`/../../children`
	`[@name='value']/values`

SAML1.SigningKeyIdentifier	The identifier for the key used to sign outgoing messages.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements`
	`[@name='policyInfo']/children`
	`[@name='attributes']/children`
	`[@name='attribute']/children`
	`[@name='name']/values`
	`[contains`
	`(.,'SAML1.SigningKeyIdentifier')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements`
	`[@name='policyInfo']/children`
	`[@name='attributes']/children`
	`[@name='attribute']/children`
	`[@name='name']/values`
	`[contains`
	`(.,'SAML1.SigningKeyIdentifier')]`
	`/../../children`
	`[@name='value']/values`

Table 15. Policy information attributes for SAML10 and SAML11 configuration profiles. (continued)

Common attributes for service providers and identity providers	**Descriptions**

SAML1.SignArtifactResponse	The indicator for whether the provider signs outgoing artifact responses.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML1.SignArtifactResponse')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML1.SignArtifactResponse')] /../../children [@name='value']/values`

SAML1.ValidateArtifactRequest	The indicator for whether the provider validates incoming artifact requests.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML1.ValidateArtifactRequest')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML1.ValidateArtifactRequest')] /../../children [@name='value']/values`

Table 15. Policy information attributes for SAML10 and SAML11 configuration profiles. (continued)

Common attributes for service providers and identity providers	**Descriptions**

SAML1.ValidateKeyIdentifier	The identifier for the key used to validate signatures on incoming messages from a partner. This is the signing public key of the partner.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML1.ValidateKeyIdentifier')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML1.ValidateKeyIdentifier')] /../../children [@name='value']/values`

SAML1.SignArtifactRequest	The indicator for whether the provider signs outgoing artifact requests.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML1.SignArtifactRequest')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML1.SignArtifactRequest')] /../../children [@name='value']/values`

Table 15. Policy information attributes for SAML10 and SAML11 configuration profiles. (continued)

Common attributes for service providers and identity providers	Descriptions

SAML1.ValidateArtifactResponse	The indicator for whether the provider validates incoming artifact responses.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML1.ValidateArtifactResponse')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML1.ValidateArtifactResponse')] /../../children [@name='value']/values`

SAML1.UseArtifactProfileForSSO	The indicator for whether an artifact profile is used for single sign on.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML1.UseArtifactProfileForSSO')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML1.UseArtifactProfileForSSO')] /../../children [@name='value']/values`

Note: SAML10 and SAML11 do not have any additional specific service provider or identity provider attributes.

SAML20 self attributes

The following table lists the SAML20 self attributes that are audited in profiles for service providers and identity providers.

Table 16. Policy information attributes for a SAML20 self profile.

Common attributes for service providers and identity providers	**Definitions**

SAML2.SigningKeyIdentifier	The identifier for the key used to sign outgoing messages.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.SigningKeyIdentifier')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.SigningKeyIdentifier')] /../../children [@name='value']/values`

SAML2.DecryptionKeyIdentifier	The pointer to the private key used to decrypt the symmetric encryption key in encrypted messages from a partner.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.DecryptionKeyIdentifier')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.DecryptionKeyIdentifier')] /../../children [@name='value']/values`

Table 16. Policy information attributes for a SAML20 self profile. (continued)

Common attributes for service providers and identity providers	**Definitions**

SAML2.EncryptionKeyTransportAlgorithm	The algorithm used to encrypt the symmetric encryption key.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2. EncryptionKeyTransportAlgorithm)]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2. EncryptionKeyTransportAlgorithm')] /../../children [@name='value']/values`

SAML2.SignArtifactRequest	The indicator for whether the provider signs outgoing artifact requests.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.SignArtifactRequest')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.SignArtifactRequest')] /../../children [@name='value']/values`

Table 16. Policy information attributes for a SAML20 self profile. (continued)

Common attributes for service providers and identity providers	**Definitions**

SAML2.SignArtifactResponse	The indicator for whether the provider signs outgoing artifact responses.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.SignArtifactResponse')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.SignArtifactResponse')] /../../children [@name='value']/values`

SAML2.SignLogoutRequest	The indicator for whether the provider signs outgoing logout requests.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.SignLogoutRequest')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.SignLogoutRequest')] /../../children [@name='value']/values`

Table 16. Policy information attributes for a SAML20 self profile. (continued)

Common attributes for service providers and identity providers	**Definitions**

SAML2.SignLogoutResponse	The indicator for whether the provider signs outgoing logout responses.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.SignLogoutResponse')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.SignLogoutResponse')] /../../children [@name='value']/values`

SAML2.SignNameIDManagementRequest	The indicator for whether the provider signs outgoing name identifier management requests.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.SignNameIDManagementRequest')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.SignNameIDManagementRequest')] /../../children [@name='value']/values`

Table 16. Policy information attributes for a SAML20 self profile. (continued)

Common attributes for service providers and identity providers	**Definitions**

SAML2.SignNameIDManagementResponse	The indicator for whether the provider signs outgoing name identifier management responses.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.SignNameIDManagementResponse')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.SignNameIDManagementResponse')] /../../children [@name='value']/values`

SAML2.PresentFederationConsent	The indicator for whether the identity provider presents a consent to federate page when the federation occurs.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.PresentFederationConsent')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.PresentFederationConsent')] /../../children [@name='value']/values`

Additional self attributes for service providers only	

Table 16. Policy information attributes for a SAML20 self profile. (continued)

Common attributes for service providers and identity providers	**Definitions**

SAML2.SignAuthnRequest	The indicator for whether the provider signs outgoing authentication requests.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.SignAuthnRequest')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.SignAuthnRequest')] /../../children [@name='value']/values`

SAML2.WantAssertionsSigned	The indicator for whether the provider wants to receive signed assertions.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.WantAssertionsSigned')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.WantAssertionsSigned')] /../../children [@name='value']/values`

Additional self attributes for identity providers only	

Table 16. Policy information attributes for a SAML20 self profile. (continued)

Common attributes for service providers and identity providers	**Definitions**

SAML2.ValidateAuthnRequest	The indicator for whether the provider validates incoming authentication requests.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.ValidateAuthnRequest')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.ValidateAuthnRequest')] /../../children [@name='value']/values`

SAML2.SignAuthnResponse	The indicator for whether the provider signs authentication responses.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.SignAuthnResponse')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.SignAuthnResponse')] /../../children [@name='value']/values`

SAML20 partner attributes

The following table lists the SAML20 partner attributes that are audited in profiles for service providers and identity providers.

Table 17. Policy information attributes for a SAML20 partner profile.

Common attributes for service providers and identity providers	**Definitions**

SAML2.SoapRequestClientBasicAuth	The indicator for whether client basic authentication is used for the SOAP backchannels.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.SoapRequestClientBasicAuth')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.SoapRequestClientBasicAuth')] /../../children [@name='value']/values`

SAML2.SoapRequestClientCertAuth	The indicator for whether client certificate authentication is used for the SOAP backchannels.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.SoapRequestClientCertAuth')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.SoapRequestClientCertAuth')] /../../children [@name='value']/values`

Table 17. Policy information attributes for a SAML20 partner profile. (continued)

Common attributes for service providers and identity providers	**Definitions**

SAML2.SoapRequestServerCertAuth	The indicator for whether server certificate authentication is used for the SOAP backchannels.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.SoapRequestServerCertAuth')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.SoapRequestServerCertAuth')] /../../children [@name='value']/values`

SAML2. SoapRequestServerCertAuthKeyIdentifier	The identifier for the key used when using server certificate authentication.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2. SoapRequestServerCertAuthKeyIdentifier')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2. SoapRequestServerCertAuthKeyIdentifier')] /../../children [@name='value']/values`

Table 17. Policy information attributes for a SAML20 partner profile. (continued)

Common attributes for service providers and identity providers	**Definitions**

SAML2. SoapRequestClientCertAuthKeyIdentifier	The identifier for the key used when using client certificate authentication.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2. SoapRequestClientCertAuthKeyIdentifier')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2. SoapRequestClientCertAuthKeyIdentifier')] /../../children [@name='value']/values`

SAML2.ValidateKeyIdentifier	The identifier for the key used to validate signatures on incoming messages from a partner. This is the signing public key of the partner.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.ValidateKeyIdentifier')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.ValidateKeyIdentifier')] /../../children [@name='value']/values`

Table 17. Policy information attributes for a SAML20 partner profile. (continued)

Common attributes for service providers and identity providers	**Definitions**

SAML2.EncryptionKeyIdentifier	The identifier for the key used to encrypt outgoing messages to a partner. This is the encrypting public key of the partner.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.EncryptionKeyIdentifier')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.EncryptionKeyIdentifier')] /../../children [@name='value']/values`

SAML2.ValidateArtifactRequest	The indicator for whether the provider validates incoming artifact requests.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.ValidateArtifactRequest')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.ValidateArtifactRequest')] /../../children [@name='value']/values`

Table 17. Policy information attributes for a SAML20 partner profile. (continued)

Common attributes for service providers and identity providers	**Definitions**

SAML2.ValidateArtifactResponse	The indicator for whether the provider validates incoming artifact responses.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.ValidateArtifactResponse')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.ValidateArtifactResponse')] /../../children [@name='value']/values`

SAML2.ValidateLogoutRequest	The indicator for whether the provider validates incoming logout requests.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.ValidateLogoutRequest')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.ValidateLogoutRequest')] /../../children [@name='value']/values`

Table 17. Policy information attributes for a SAML20 partner profile. (continued)

Common attributes for service providers and identity providers	**Definitions**

SAML2.ValidateLogoutResponse	The indicator for whether the provider validates incoming logout responses.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.ValidateLogoutResponse')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.ValidateLogoutResponse')] /../../children [@name='value']/values`

SAML2. ValidateNameIDManagementRequest	The indicator for whether the provider validates incoming name identifier management requests.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2. ValidateNameIDManagementRequest')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2. ValidateNameIDManagementRequest')] /../../children [@name='value']/values`

Table 17. Policy information attributes for a SAML20 partner profile. (continued)

Common attributes for service providers and identity providers	**Definitions**

SAML2. ValidateNameIDManagementResponse	The indicator for whether the provider validates incoming name identifier management responses.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2. ValidateNameIDManagementResponse')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2. ValidateNameIDManagementResponse')] /../../children [@name='value']/values`

SAML2.EncryptNameIdentifiers	The indicator for whether name identifiers need to be encrypted for the partner.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.EncryptNameIdentifiers')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.EncryptNameIdentifiers)] /../../children [@name='value']/values`

Table 17. Policy information attributes for a SAML20 partner profile. (continued)

Common attributes for service providers and identity providers	**Definitions**

SAML2.BlockEncryptionAlgorithm	The algorithm used to encrypt the data.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.BlockEncryptionAlgorithm')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.BlockEncryptionAlgorithm')] /../../children [@name='value']/values`

Additional partner attributes for service providers only	

SAML2.WantAssertionsSigned	The indicator for whether the provider wants to receive signed assertions.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.WantAssertionsSigned')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.WantAssertionsSigned')] /../../children [@name='value']/values`

Additional partner attributes for identity providers only	

Table 17. Policy information attributes for a SAML20 partner profile. (continued)

Common attributes for service providers and identity providers	Definitions

SAML2.SignAssertions	The indicator for whether the provider signs assertions.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.SignAssertions')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.SignAssertions')] /../../children [@name='value']/values`

SAML2.EncryptAssertions	The indicator for whether the provider encrypts assertions.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.EncryptAssertions')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML2.EncryptAssertions')] /../../children [@name='value']/values`

LibertyIDFFv11 and LibertyIDFFv12 attributes

The following table lists the LibertyIDFFv11 and LibertyIDFFv12 configuration attributes that are audited in profiles for service providers and identity providers.

Table 18. Policy information attributes for LibertyIDFFv11 and LibertyIDFFv12 configuration profiles.

Common attributes for identity providers and service providers	**Definitions**

SoapRequestClientBasicAuth	The indicator for whether client basic authentication is used for the SOAP backchannels.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SoapRequestClientBasicAuth ')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SoapRequestClientBasicAuth ')] /../../children [@name='value']/values`

SoapRequestClientCertAuth	The indicator for whether client certificate authentication is used for the SOAP backchannels.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SoapRequestClientCertAuth')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SoapRequestClientCertAuth')] /../../children [@name='value']/values`

Table 18. Policy information attributes for LibertyIDFFv11 and LibertyIDFFv12 configuration profiles. (continued)

Common attributes for identity providers and service providers	**Definitions**

SoapRequestServerCertAuth	The indicator for whether server certificate authentication is used for the SOAP backchannels.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SoapRequestServerCertAuth')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SoapRequestServerCertAuth')] /../../children [@name='value']/values`

SoapRequestServerCertAuthKeyIdentifier	The identifier for the key used when using server certificate authentication.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (., 'SoapRequestServerCertAuthKeyIdentifier')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (., 'SoapRequestServerCertAuthKeyIdentifier')] /../../children [@name='value']/values`

Table 18. Policy information attributes for LibertyIDFFv11 and LibertyIDFFv12 configuration profiles. (continued)

Common attributes for identity providers and service providers	**Definitions**

SoapRequestClientCertAuthKeyIdentifier	The identifier for the key used when using client certificate authentication.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements`
	`[@name='policyInfo']/children`
	`[@name='attributes']/children`
	`[@name='attribute']/children`
	`[@name='name']/values`
	`[contains`
	`(.,`
	`'SoapRequestClientCertAuthKeyIdentifier')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements`
	`[@name='policyInfo']/children`
	`[@name='attributes']/children`
	`[@name='attribute']/children`
	`[@name='name']/values`
	`[contains`
	`(.,`
	`'SoapRequestClientCertAuthKeyIdentifier')]`
	`/../../children`
	`[@name='value']/values`

SignLibertyMessages	The identifier for the key used to sign outgoing messages.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements`
	`[@name='policyInfo']/children`
	`[@name='attributes']/children`
	`[@name='attribute']/children`
	`[@name='name']/values`
	`[contains`
	`(.,'SignLibertyMessages')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements`
	`[@name='policyInfo']/children`
	`[@name='attributes']/children`
	`[@name='attribute']/children`
	`[@name='name']/values`
	`[contains`
	`(.,'SignLibertyMessages')]`
	`/../../children`
	`[@name='value']/values`

Table 18. Policy information attributes for LibertyIDFFv11 and LibertyIDFFv12 configuration profiles. (continued)

Common attributes for identity providers and service providers	**Definitions**

LibertyMessageSignatureRequired	The indicator for whether the provider signs outgoing artifact responses.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'LibertyMessageSignatureRequired')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'LibertyMessageSignatureRequired')] /../../children [@name='value']/values`

SigningKeyIdentifier	The indicator for whether the provider signs outgoing artifact requests.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SigningKeyIdentifier')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SigningKeyIdentifier')] /../../children [@name='value']/values`

Table 18. Policy information attributes for LibertyIDFFv11 and LibertyIDFFv12 configuration profiles. (continued)

Common attributes for identity providers and service providers	Definitions

ValidateKeyIdentifier	The identifier for the key used to validate signatures on incoming messages from a partner. This is the signing public key of the partner.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'ValidateKeyIdentifier')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'ValidateKeyIdentifier')] /../../children [@name='value']/values`

UseArtifactProfileForSSO	The indicator for whether an artifact profile is used for single sign on.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'UseArtifactProfileForSSO')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'UseArtifactProfileForSSO')] /../../children [@name='value']/values`

Note: LibertyIDFFv11 and LibertyIDFFv12 do not have any additional specific service provider or identity provider attributes.

WS-Federation attributes

The following table lists the WS-Federation configuration attributes that are audited in profiles for service providers and identity providers.

Table 19. Policy information attributes for WS-Federation configuration profiles.

Common attributes for identity providers and service providers	**Definitions**

com.tivoli.am.fim.sts.saml.1.1.assertion.sign	The identifier for the key used when signing assertions.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'com.tivoli.am.fim.sts.saml.1.1. assertion.sign')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'com.tivoli.am.fim.sts.saml.1.1. assertion.sign')] /../../children [@name='value']/values`

com.tivoli.am.fim.sts.saml.1.1.assertion. verify.signatures	The identifier for the key used when verifying assertions.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'com.tivoli.am.fim.sts.saml.1.1. assertion.verify.signatures')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'com.tivoli.am.fim.sts.saml.1.1. assertion.verify.signatures')] /../../children [@name='value']/values`

Note: WS-Federation does not have any additional specific service provider or identity provider attributes.

Web service partner attributes

The following table lists the Web service partner attributes that are audited in profiles for service providers and identity providers.

Table 20. Web service partner profile attributes

Common attributes for identity providers and service providers	**Definitions**

PartnerProviderId	The Web service URL.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'PartnerProviderId')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'PartnerProviderId')] /../../children [@name='value']/values`

FederationPartnerDisplayName	The company name.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'FederationPartnerDisplayName')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'FederationPartnerDisplayName')] /../../children [@name='value']/values`

Table 20. Web service partner profile attributes (continued)

Common attributes for identity providers and service providers	Definitions

State	The indicator for whether the partner is enabled or disabled.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'State')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'State')] /../../children [@name='value']/values`

TokenType	The incoming token type for the partner.
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'TokenType')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'TokenType')] /../../children [@name='value']/values`

Note: Web service partner does not have any additional specific service provider or identity provider attributes.

Sample of a IBM_SECURITY_MGMT_POLICY event

The following is an example of a IBM_SECURITY_MGMT_POLICY event:

```
<CommonBaseEvent
 creationTime="2006-04-26T12:22:25.874Z"
 extensionName="IBM_SECURITY_MGMT_POLICY"
 globalInstanceId="CE11DAD51F526D53D0E30FDAA2C9637F07"
 sequenceNumber="1"
 version="1.0.1">
 <extendedDataElements name="action" type="string">
```

```
 <values>Create</values>
</extendedDataElements>
<extendedDataElements name="outcome" type="noValue">
 <children name="majorStatus" type="int">
  <values>0</values></children>
 <children name="result" type="string">
  <values>SUCCESSFUL</values></children>
</extendedDataElements>
<extendedDataElements name="policyInfo" type="noValue">
 <children name="attributes" type="noValue">
  <children name="attribute" type="noValue">
  <children name="value" type="string">
   <values>saml11-ip</values></children>
  <children name="name" type="string">
   <values>FederationName</values></children>
 </children>
 <children name="attribute" type="noValue">
  <children name="value" type="string">
   <values>enabled</values></children>
  <children name="name" type="string">
   <values>State</values></children>
 </children>
 <children name="attribute" type="noValue">
  <children name="value" type="string">
   <values>saml11-ip</values></children>
  <children name="name" type="string">
   <values>FederationId</values></children>
 </children>
 <children name="attribute" type="noValue">
  <children name="value" type="string">
   <values>DefaultKeyStore_testkey</values></children>
  <children name="name" type="string">
   <values>SAML1.SigningKeyIdentifier</values></children>
 </children>
 <children name="attribute" type="noValue">
  <children name="value" type="string">
   <values>true</values></children>
  <children name="name" type="string">
   <values>SAML1.SignArtifactResponse</values></children>
 </children>
 <children name="attribute" type="noValue">
  <children name="value" type="string">
   <values>SAML1_1</values></children>
  <children name="name" type="string">
   <values>FederationProtocol</values></children>
 </children>
</children>
<children name="type" type="string">
 <values>federation</values></children>
<children name="name" type="string">
 <values>saml11-ip</values></children>
</extendedDataElements>
<extendedDataElements name="mgmtInfo" type="noValue">
 <children name="command" type="string">
  <values>CreateFederation</values></children>
</extendedDataElements>
<extendedDataElements name="userInfo" type="noValue">
 <children name="appUserName" type="string">
  <values>Not Available</values></children>
 <children name="registryUserName" type="string">
  <values>Not Available</values></children>
</extendedDataElements>
<sourceComponentId
 application="ITFIM#6.2"
 component="IBM Tivoli Federated Identity Manager"
 componentIdType="ProductName"
 executionEnvironment="Linux[x86]#2.4.21-4.EL"
```

```
  location="localhost.localdomain"
  locationType="FQHostname"
  subComponent="com.tivoli.am.fim.mgmt.fim.FIMManagementImpl"
  threadId="SoapConnectorThreadPool : 1"
  componentType=
  "http://www.ibm.com/namespaces/autonomic/Tivoli_componentTypes"/>
  <situation categoryName="ReportSituation">
   <situationType xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                  xsi:type="ReportSituation"
                  reasoningScope="INTERNAL"
                  reportCatagory="SECURITY"/>
  </situation>
</CommonBaseEvent>
```

Audit configuration management (IBM_SECURITY_MGMT_AUDIT)

This event type provides information about changes to the auditing settings, for example, if auditing is enabled or disabled, or if auditing is set for specific transactions.

IBM_SECURITY_MGMT_AUDIT events are generated when the audit configuration is modified. Changes to the following data are audited:

- User name
- Action
- Domain
- Audit configuration properties:
  - Enable auditing
  - Enable auditing for specific audit event types. Event types are shown in the table below under the mgmtInfo element.
  - Audit log location
  - Maximum number of audit files
  - Maximum audit file size
  - Disk cache location
  - Web service SSL keystore
  - Enable Web service basic authentication
- Disable auditing:
  - User name
  - Action

The following table lists the elements that can be displayed in the output of an IBM_SECURITY_MGMT_AUDIT event.

Table 21. Elements used in IBM_SECURITY_MGMT_AUDIT events

Element	Description
action	The type of action that occurred against the audit settings. Possible values are `Modify` and `Disable`. The XPath is: `CommonBaseEvent/extendedDataElements [@name='action']/values`

Table 21. Elements used in IBM_SECURITY_MGMT_AUDIT events (continued)

Element	Description
mgmtInfo	Information about the auditing operation. The supported items and values are: EnableAudit=true | false Domain=*domain_ name* AuditLogLocation=*path* CacheLocation=*path* WebServiceBasicAuthUsername=*username* WebServiceBasicAuthPassword=*password* WebServiceKeyIdentifier=*keyname* WebServiceURL=*URL* MaxAuditFiles=*number* AuditFileSize=*number* UseWebServiceBasicAuth=true | false WebServiceKeystore=*keystore_name* AuditSecurityAuthnEvents=true | false AuditSecurityAuthnTerminateEvents=true | false AuditSecurityEncryptionEvents=true | false AuditSecuritySigningEvents=true | false AuditSecurityFederationEvents=true | false AuditSecurityTrustEvents=true | false AuditSecurityMgmtPolicyEvents=true | false AuditSecurityMgmtAuditEvents=true | false The XPath is: `CommonBaseEvent/extendedDataElements [@name='mgmtInfo']/children [@name='command']/values`
userInfo	Information about the user who is performing the operation. The XPath is: `CommonBaseEvent/extendedDataElements [@name='userInfo']/children [@name='appUserName']/children [@name='registryUserName']/values`
type	Always set to the `audit` value. The XPath is: `CommonBaseEvent/extendedDataElements [@name='type']/values`

Sample of an IBM_SECURITY_MGMT_AUDIT event

The following is an example of an IBM_SECURITY_MGMT_AUDIT event:

```
<CommonBaseEvent
 creationTime="2007-04-25T07:01:51.726Z"
 extensionName="IBM_SECURITY_MGMT_AUDIT"
```

```
globalInstanceId="CEFA81F627EBCFC5DFA1DBF2FAD8573020"
sequenceNumber="1"
version="1.0.1">
<contextDataElements name="Security Event Factory" type="eventTrailId">
 <contextId>FIM_278bcbef011213a9865f8a816f9717a6+1969112872</contextId>
</contextDataElements>
<extendedDataElements name="mgmtInfo" type="noValue">
 <children name="command" type="string">
  <values>EnableAudit=true;
   Domain=mydomain-server1;
   AuditLogLocation=audit_location;
   AuditFileSize=10;
   MaxAuditFiles=100;AuditAuthnEvents=true;
   AuditAuthnTerminateEvents=true;
   AuditFederationEvents=true;
   AuditTrustEvents=true;
   AuditSigningEvents=true;
   AuditEncryptionEvents=true;
   AuditMgmtPolicyEvents=true;
   AuditMgmtAuditEvents=true;
  </values>
 </children>
</extendedDataElements>
<extendedDataElements name="type" type="string">
 <values>audit</values>
</extendedDataElements>
<extendedDataElements name="userInfo" type="noValue">
 <children name="appUserName" type="string">
  <values>unauthenticatedUser</values>
 </children>
 <children name="registryUserName" type="string">
  <values>Not Available</values>
 </children>
</extendedDataElements>
<extendedDataElements name="action" type="string">
 <values>Modify</values>
</extendedDataElements>
<extendedDataElements name="outcome" type="noValue">
 <children name="result" type="string">
  <values>SUCCESSFUL</values>
 </children>
 <children name="majorStatus" type="int">
  <values>0</values>
 </children>
</extendedDataElements>
<sourceComponentId
 application="ITFIM#6.2"
 component="IBM Tivoli Federated Identity Manager"
 componentIdType="ProductName"
 executionEnvironment="Linux[x86]#2.6.9-34.ELsmp"
 location="fimfun2.austin.ibm.com"
 locationType="FQHostname"
 subComponent="com.tivoli.am.fim.mgmt.fim.FIMManagementImpl"
 threadId="SoapConnectorThreadPool : 0"
 componentType=
 "http://www.ibm.com/namespaces/autonomic/Tivoli_componentTypes"/>
 <situation categoryName="ReportSituation">
  <situationType xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                 xsi:type="ReportSituation"
                 reasoningScope="INTERNAL"
                 reportCategory="SECURITY"/>
 </situation>
</CommonBaseEvent>
```

Audit Provisioning (IBM_SECURITY_MGMT_PROVISIONING)

This event type provides information about the provisioning of a cardspace to a user.

The following table lists the elements that can be displayed in the output of an IBM_SECURITY_MGMT_PROVISIONING event. Note that the only instantiation of this event type is for the downloading of information cards; the values used in the sample are within this context.

Table 22. Elements used in IBM_SECURITY_MGMT_PROVISIONING events

Element	Description
action	The action that was performed. The supported actions are: add The XPath is: `CommonBaseEvent/extendedDataElements [@name='action']/values`
mgmtInfo. command	Information about the management operation. The supported management operations are: Download Card The XPath is: `CommonBaseEvent/extendedDataElements [@name='mgmtInfo']/children [@name='command']/values`
registryInfo.name	Name of the registry that the information card belongs to. The value is always set to ″TFIM″. The XPath is: `CommonBaseEvent/extendedDataElements [@name='registryInfo']/children [@name='name']/values`

registryObjectInfo. attributes	The different attributes for this registryObjectInfo object. See Table 2 below for attributes that might be present in the event. Each attribute consists of a source, name, and value. The value of the source attribute is always set to ″user″.
	The XPath for the attribute source is:
	`CommonBaseEvent/extendedDataElements [@name='registryObjectInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='source']/values`
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='registryObjectInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values`
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='registryObjectInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='value']/values`

Table 22. Elements used in IBM_SECURITY_MGMT_PROVISIONING events (continued)

Element	Description

registryObjectInfo.
description
	Description of the registry object. The value is the following string (note the placeholder variables):
	`"Information Card for user: username from federation endpoint: URL_of_getcard.crd_endpoint"`
	The XPath is:
	`CommonBaseEvent/extendedDataElements [@name='registryObjectInfo']/children [@name='description']/values`

registryObjectInfo. nameInApp	Name of the user who is associated with the registry object. The value for this element is the username of the user that is downloading the card. The XPath is: `CommonBaseEvent/extendedDataElements [@name='registryObjectInfo']/children [@name='nameInApp']/values`
registryObjectInfo. type	Type of the registry object. The value is always set to ″InformationCard″. The XPath is: `CommonBaseEvent/extendedDataElements [@name='registryObjectInfo']/children [@name='type']/values`
type	Type of the provisioning operation. The value is always set to ″identity″. The XPath is: `CommonBaseEvent/extendedDataElements [@name='type']/values`
userInfo. appUserName	Username of the user who is downloading a card. The XPath is: `CommonBaseEvent/extendedDataElements [@name='userInfo']/children [@name='appUserName']/values`

Table 23. Attributes for Cardspace Events

Attribute	Definition

CardId	The ID of the information card.
	
	The XPath for the attribute source is:
	`CommonBaseEvent/extendedDataElements`
	`[@name='registryObjectInfo']/children`
	`[@name='attributes']/children`
	`[@name='attribute']/children`
	`[@name='source']/values[contains(.,'user')]`
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements`
	`[@name='registryObjectInfo']/children`
	`[@name='attributes']/children`
	`[@name='attribute']/children`
	`[@name='name']/values[contains(.,'CardId')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements`
	`[@name='registryObjectInfo']/children`
	`[@name='attributes']/children`
	`[@name='attribute']/children`
	`[@name='value']/values(.,'CardId')]`
	`/../../children[@name='value']/values`

CardName	The name of the information card.
	
	The XPath for the attribute source is:
	`CommonBaseEvent/extendedDataElements`
	`[@name='registryObjectInfo']/children`
	`[@name='attributes']/children`
	`[@name='attribute']/children`
	`[@name='source']/values[contains(.,'user')]`
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements`
	`[@name='registryObjectInfo']/children`
	`[@name='attributes']/children`
	`[@name='attribute']/children`
	`[@name='name']/values[contains(.,'CardName')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements`
	`[@name='registryObjectInfo']/children`
	`[@name='attributes']/children`
	`[@name='attribute']/children`
	`[@name='value']/values[contains(.,'CardName')]`
	`/../../children[@name='value']/values`

Table 23. Attributes for Cardspace Events (continued)

Attribute	Definition

IssueTime	The time that the information card was issued.
	
	The XPath for the attribute source is:
	`CommonBaseEvent/extendedDataElements`
	`[@name='registryObjectInfo']/children`
	`[@name='attributes']/children`
	`[@name='attribute']/children`
	`[@name='source']/values[contains(.,'user')]`
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements`
	`[@name='registryObjectInfo']/children`
	`[@name='attributes']/children`
	`[@name='attribute']/children`
	`[@name='name']/values[contains(.,'IssueTime')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements`
	`[@name='registryObjectInfo']/children`
	`[@name='attributes']/children`
	`[@name='attribute']/children`
	`[@name='value']/values[contains(.,'IssueTime')]`
	`/../../children[@name='value']/values`

ExpireTime	The time when the information card expires.
	
	The XPath for the attribute source is:
	`CommonBaseEvent/extendedDataElements`
	`[@name='registryObjectInfo']/children`
	`[@name='attributes']/children`
	`[@name='attribute']/children`
	`[@name='source']/values[contains(.,'user')]`
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements`
	`[@name='registryObjectInfo']/children`
	`[@name='attributes']/children`
	`[@name='attribute']/children`
	`[@name='name']/values[contains(.,'ExpireTime')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements`
	`[@name='registryObjectInfo']/children`
	`[@name='attributes']/children`
	`[@name='attribute']/children`
	`[@name='value']/values[contains(.,'ExpireTime')]`
	`/../../children[@name='value']/values`

Table 23. Attributes for Cardspace Events (continued)

Attribute	Definition

ClaimURI	Supported claims for the information card. A separate ClaimURI attribute is present for each claim that an information card supports.
	
	The XPath for the attribute source is:
	`CommonBaseEvent/extendedDataElements [@name='registryObjectInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='source']/values[contains(.,'user')]`
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='registryObjectInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values[contains(.,'ClaimURI')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='registryObjectInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='value']/values[contains(.,'ClaimURI')] /../../children[@name='value']/values`

Authentication Method	The authentication method for the information card. The value is set to either ″UsernameToken″ or ″SelfSignedSAML″.
	
	The XPath for the attribute source is:
	`CommonBaseEvent/extendedDataElements [@name='registryObjectInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='source']/values[contains(.,'user')]`
	
	The XPath for the attribute name is:
	`CommonBaseEvent/extendedDataElements [@name='registryObjectInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values[contains(., 'AuthenticationMethod')]`
	
	The XPath for the attribute value is:
	`CommonBaseEvent/extendedDataElements [@name='registryObjectInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='value']/values[contains(., 'AuthenticationMethod')] /../../children[@name='value']/values`

Sample of an IBM_SECURITY_MGMT_PROVISIONING event

The following is an example of an IBM_SECURITY_MGMT_PROVISIONING event:

```
<CommonBaseEvent
 creationTime="2008-03-04T07:47:22.677Z"
 extensionName="IBM_SECURITY_MGMT_PROVISIONING"
 globalInstanceId="CE972AC9A4D344DC9AA1DCE9BF397CD560"
 sequenceNumber="6"
```

```
version="1.0.1">
<contextDataElements name="Security Event Factory" type="eventTrailId">
 <contextId>FIM_78c24d9b01181b2694acd929e91f9be5+1361357769</contextId>
</contextDataElements>
<extendedDataElements name="mgmtInfo" type="noValue">
 <children name="command" type="string">
  <values>Download Card</values>
 </children>
</extendedDataElements>
<extendedDataElements name="registryInfo" type="noValue">
 <children name="name" type="string">
  <values>/csiput/cardspace/getcard.crd</values>
 </children>
 <children name="type" type="string">
  <values></values>
 </children>
 <children name="location" type="string">
  <values></values>
 </children>
 <children name="serverPort" type="string">
  <values></values>
 </children>
 <children name="locationType" type="string">
  <values></values>
 </children>
</extendedDataElements>
<extendedDataElements name="registryObjectInfo" type="noValue">
 <children name="description" type="string">
  <values>Information Card for user: shane from federation
          endpoint: /csiput/cardspace/getcard.crd</values>
 </children>
 <children name="type" type="string">
  <values>InformationCard</values>
 </children>
 <children name="attributes" type="noValue">
  <children name="attribute" type="noValue">
   <children name="source" type="string">
    <values>user</values>
   </children>
   <children name="name" type="string">
    <values>ClaimURI</values>
   </children>
   <children name="value" type="string">
    <values>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/
  surname</values>
   </children>
  </children>
  <children name="attribute" type="noValue">
   <children name="source" type="string">
    <values>user</values>
   </children>
   <children name="name" type="string">
    <values>ClaimURI</values>
   </children>
   <children name="value" type="string">
    <values>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/
  emailaddress</values>
   </children>
  </children>
  <children name="attribute" type="noValue">
   <children name="source" type="string">
    <values>user</values>
   </children>
   <children name="name" type="string">
    <values>ClaimURI</values>
   </children>
   <children name="value" type="string">
    <values>http://burtongroup.com/interop/2007/05/identity/
  group</values>
   </children>
  </children>
```

```
<children name="attribute" type="noValue">
 <children name="source" type="string">
  <values>user</values>
 </children>
 <children name="name" type="string">
  <values>CardName</values>
 </children>
 <children name="value" type="string">
  <values>ibm-shane-ut</values>
 </children>
</children>
<children name="attribute" type="noValue">
 <children name="source" type="string">
  <values>user</values>
 </children>
 <children name="name" type="string">
  <values>ClaimURI</values>
 </children>
 <children name="value" type="string">
  <values>http://burtongroup.com/interop/2007/05/identity/
cameratype</values>
 </children>
</children>
<children name="attribute" type="noValue">
 <children name="source" type="string">
  <values>user</values>
 </children>
 <children name="name" type="string">
  <values>ClaimURI</values>
 </children>
 <children name="value" type="string">
  <values>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/
mobilephone</values>
 </children>
</children>
<children name="attribute" type="noValue">
 <children name="source" type="string">
  <values>user</values>
 </children>
 <children name="name" type="string">
  <values>ClaimURI</values>
 </children>
 <children name="value" type="string">
  <values>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/
dateofbirth</values>
 </children>
</children>
<children name="attribute" type="noValue">
 <children name="source" type="string">
  <values>user</values>
 </children>
 <children name="name" type="string">
  <values>AuthenticationMethod</values>
 </children>
 <children name="value" type="string">
  <values>UsernameToken</values>
 </children>
</children>
<children name="attribute" type="noValue">
 <children name="source" type="string">
  <values>user</values>
 </children>
 <children name="name" type="string">
  <values>ClaimURI</values>
 </children>
 <children name="value" type="string">
  <values>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/
stateorprovince</values>
 </children>
</children>
<children name="attribute" type="noValue">
```

```
 <children name="source" type="string">
  <values>user</values>
 </children>
 <children name="name" type="string">
  <values>ClaimURI</values>
 </children>
 <children name="value" type="string">
  <values>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/
gender</values>
 </children>
</children>
<children name="attribute" type="noValue">
 <children name="source" type="string">
  <values>user</values>
 </children>
 <children name="name" type="string">
  <values>ClaimURI</values>
 </children>
 <children name="value" type="string">
  <values>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/
givenname</values>
 </children>
</children>
<children name="attribute" type="noValue">
 <children name="source" type="string">
  <values>user</values>
 </children>
 <children name="name" type="string">
  <values>ClaimURI</values>
 </children>
 <children name="value" type="string">
  <values>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/
website</values>
 </children>
</children>
<children name="attribute" type="noValue">
<children name="source" type="string">
  <values>user</values>
 </children>
 <children name="name" type="string">
  <values>ClaimURI</values>
 </children>
 <children name="value" type="string">
  <values>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/
postalcode</values>
 </children>
</children>
<children name="attribute" type="noValue">
 <children name="source" type="string">
  <values>user</values>
 </children>
 <children name="name" type="string">
  <values>ClaimURI</values>
 </children>
 <children name="value" type="string">
  <values>http://www.ibmidentitydemo.com/claims/assurancelevel</values>
 </children>
</children>
<children name="attribute" type="noValue">
 <children name="source" type="string">
  <values>user</values>
 </children>
 <children name="name" type="string">
  <values>ClaimURI</values>
 </children>
 <children name="value" type="string">
  <values>http://burtongroup.com/interop/2007/05/identity/
groupRole</values>
 </children>
</children>
<children name="attribute" type="noValue">
```

```
 <children name="source" type="string">
  <values>user</values>
 </children>
 <children name="name" type="string">
  <values>ClaimURI</values>
 </children>
 <children name="value" type="string">
  <values>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/
privatepersonalidentifier</values>
 </children>
</children>
<children name="attribute" type="noValue">
 <children name="source" type="string">
  <values>user</values>
 </children>
 <children name="name" type="string">
  <values>ClaimURI</values>
 </children>
 <children name="value" type="string">
  <values>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/
homephone</values
 </children>
</children>
<children name="attribute" type="noValue">
 <children name="source" type="string">
  <values>user</values>
 </children>
 <children name="name" type="string">
  <values>IssueTime</values>
 </children>
 <children name="value" type="string">
  <values>2008-03-04T07:47:22Z</values>
 </children>
</children>
<children name="attribute" type="noValue">
 <children name="source" type="string">
  <values>user</values>
 </children>
 <children name="name" type="string">
  <values>ExpireTime</values>
 </children>
 <children name="value" type="string">
  <values>2009-03-04T07:47:22Z</values>
 </children>
</children>
<children name="attribute" type="noValue">
 <children name="source" type="string">
  <values>user</values>
 </children>
 <children name="name" type="string">
  <values>ClaimURI</values>
 </children>
 <children name="value" type="string">
  <values>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/
locality</values>
 </children>
</children>
<children name="attribute" type="noValue">
 <children name="source" type="string">
  <values>user</values>
 </children>
 <children name="name" type="string">
  <values>ClaimURI</values>
 </children>
 <children name="value" type="string">
  <values>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/
country</values>
 </children>
</children>
<children name="attribute" type="noValue">
 <children name="source" type="string">
```

```
    <values>user</values>
   </children>
   <children name="name" type="string">
    <values>ClaimURI</values>
   </children>
   <children name="value" type="string">
    <values>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/
  streetaddress</values>
   </children>
  </children>
  <children name="attribute" type="noValue">
   <children name="source" type="string">
    <values>user</values>
   </children>
   <children name="name" type="string">
    <values>ClaimURI</values>
   </children>
   <children name="value" type="string">
    <values>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/
  otherphone</values>
   </children>
  </children>
  <children name="attribute" type="noValue">
   <children name="source" type="string">
    <values>user</values>
   </children>
   <children name="name" type="string">
    <values>CardId</values>
   </children>
   <children name="value" type="string">
    <values>https://ibmaus27.lnk.telstra.net/FIM/sps/csiput/cardspace/
  sts/uuid78c24d66-0118-17ae-b153-d929e91f9be5</values>
   </children>
  </children>
 </children>
 <children name="nameInApp" type="string">
  <values>shane</values>
 </children>
</extendedDataElements>
<extendedDataElements name="type" type="string">
 <values>identity</values>
</extendedDataElements>
<extendedDataElements name="userInfo" type="noValue">
 <children name="appUserName" type="string">
  <values>shane</values>
 </children>
 <children name="registryUserName" type="string">
  <values>Not Available</values>
 </children>
</extendedDataElements>
<extendedDataElements name="action" type="string">
  <values>add</values>
</extendedDataElements>
<extendedDataElements name="outcome" type="noValue">
 <children name="result" type="string">
  <values>SUCCESSFUL</values>
 </children>
 <children name="majorStatus" type="int">
  <values>0</values>
 </children>
</extendedDataElements>
<sourceComponentId
 application="ITFIM#6.2"
 component="IBM Tivoli Federated Identity Manager"
 componentIdType="ProductName"
 executionEnvironment="Linux[x86]#2.4.21-27.EL"
 location="localhost.localdomain"
 locationType="FQHostname"
 subComponent="com.tivoli.am.fim.cardspace.delegates.
 CardSpaceGetCardDelegate"
 threadId="WebContainer : 0"
```

```
  componentType=
  "http://www.ibm.com/namespaces/autonomic/Tivoli_componentTypes"/>
  <situation categoryName="ReportSituation">
   <situationType xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                  xsi:type="ReportSituation"
                  reasoningScope="INTERNAL"
                  reportCategory="SECURITY"/>
  </situation>
</CommonBaseEvent>
```

Chapter 3. Overview of the Common Audit Service

The Tivoli Common Audit Service can be used to provide auditing for your environment. **Note:** Common Audit Service *is the same feature as the Common Auditing and Reporting Service. The name has changed to indicate that exploiting products now provide the reporting functionality. Common Audit Service provides only the utilities that enable you to manage the tables used to create reports.*

The Common Audit Service consists of the following parts:

Audit server (Common Audit Service server
The audit server component is used by all exploiting products. It is also referred to as the "audit service."

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

Client The Common Audit Service Version 6.1 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 the Tivoli Access Manager for e-business Version 6.1 C Runtime.

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 Tivoli Access Manager.

Staging utility and XML data store utilities
These utilities enable you 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 76 illustrates how data flows between the components of Common Audit Service and an exploiting product, such as Tivoli Federated Identity Manager..

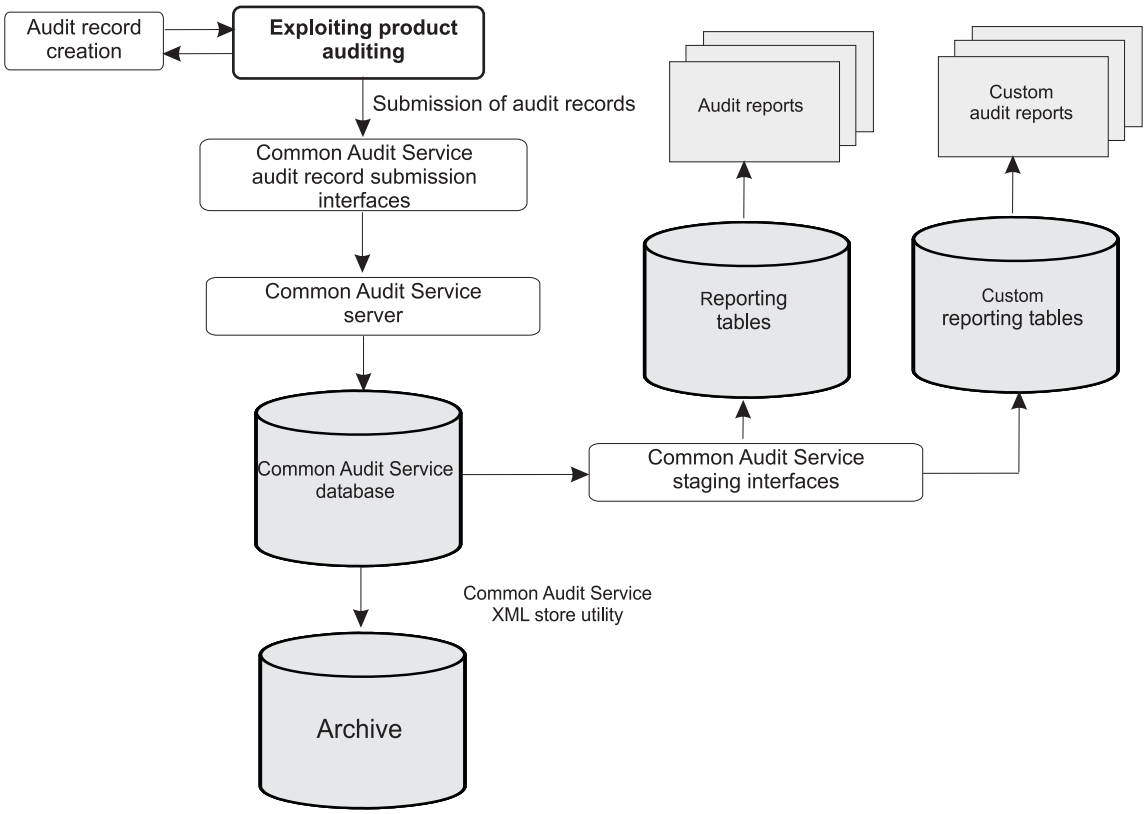


Figure 1. Structure of Common Audit Service

Common Audit Service infrastructure

The Common Audit Service infrastructure provides the mechanisms to submit, centrally collect, and persistently store and report on audit data. Common Audit Service uses the Common Base Event format, which is a standard, XML-based format that defines the structure of a security event. The Common Audit Service Security Event Factory allows for the generation of Common Base Event security events

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

The Common Audit 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.

Scenario for collecting audit data

The following tasks describe a general scenario for collecting audit data to generate a report.

1. Identify the installed IBM security software. For example, you might have Tivoli Federated Identity Manager and Tivoli Access Manager for e-business in your environment.
2. Identify the type of events to audit. For example, to report on trust events for Tivoli Federated Identity Manager, you need to configure Tivoli Federated Identity Manager to send the IBM_SECURITY_TRUST event.

   Refer to Chapter 1, "Tivoli Federated Identity Manager auditing," on page 1 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 will determine the type of reports, frequency of staging, and so on. For example, if you are generating events in millions, you may have to archive and prune archived data more frequently. If you prune frequently, then you will limit your ability to run security events details reports. Also, a large number of security events would increase the time to stage data and force you to schedule report generation much later. In addition, start and end time parameters for the reports must be selected so that the number of security events returned is approximately 100,000. For the purposes of this scenario, we will assume 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 136.
   b. Archive the data from the XML data store. Depending on the volume of security events, you will run the archive process once or twice a week.

      The archival process consists of four phases:
      - Pre-archive
      - Archive 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 137.
   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 136.
5. Generate reports using the report-generation tool of the exploiting product, for example, Tivoli Federated Identity Manager or Tivoli Access 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 4. Installing Common Audit Service

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

The installation of Common Audit 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 Audit 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 Audit Service audit server, see Chapter 6, "Upgrading the Common Audit Service audit server from earlier versions," on page 113.

Installing prerequisite products

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

The Common Audit Service audit server requires the following software. For details about the supported versions of the prerequisite software, see the *Release Notes*.

IBM DB2 Server
The DB2 server is required to configure the audit server; it does not have to be installed in order to install the Common Audit 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 Audit Service or the Common Audit Service Configuration Console.

DB2 client (DB2 8 Administration Client, DB2 9 Client, or DB2 9 Runtime Client)
If the DB2 server is installed on a separate system from the Common Audit 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 9.1, install the DB2 Client (or Runtime Client) Version 9.1. If the DB2 server is DB2 Version 8.2, install DB2

Administration Client Version 8.2. 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 Administration Client on Linux and UNIX systems" on page 81 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.

To install the DB2 client, complete the following steps:

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

   DB2 8 Administration Client
   http://www.ibm.com/software/data/db2/udb/support/downloadv8.html

   DB2 9 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 using either Download Director or FTP.
3. Follow the directions in the installation wizard to install the client.

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 performed on a Deployment Manager node in the WebSphere Application Server Network Deployment edition; otherwise, configuration is performed 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 resides. The host name is the name of the node that is known to the TCP/IP 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 server's database manager configuration file, and 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 Administration Client on Linux and UNIX systems

This section describes how to install the DB2 Administration Client on a Linux or UNIX 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.

To install the DB2 client, complete the following steps:

1. Download the DB2 client for the appropriate Linux or UNIX platform from the following Web site:

   DB2 8 Administration Client
   http://www.ibm.com/software/data/db2/udb/support/downloadv8.html

   DB2 9 client (DB2 Client or DB2 Runtime Client)
   http://www.ibm.com/software/data/db2/udb/support/downloadv9.html
2. Uncompress and untar the file.
3. Run **db2setup** that is located 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 and click **Next**.
8. Select **Typical** installation type and click **Next**.
9. Select **Create a DB2 instance** in the Set up a DB2 instance window.
10. Select the **New User** radio button and specify a user name and password. You can either accept the defaults or change those that are appropriate for your environment. If you have already created a user, you might want to also select the **Existing User** radio button and specify the user name. Click **Next**.
11. Click **Finish**.

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. 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 performed on a Deployment Manager node in the WebSphere Application Server Network Deployment edition; otherwise, configuration is performed on a stand-alone server node. Before cataloguing is performed on a UNIX platform, a DB2 client instance must have been 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:

```
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 resides. The host name is the name of the node that is known to the TCP/IP 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 server's database manager configuration file, and 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 Audit Service on a Windows, Linux, or UNIX 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 perform the following items before you start the installation of Common Audit Service:

- Prepare the values you will use for the installation. See Table 24 on page 86.
- Verify that there is enough space to install the Common Audit Service audit server. Additionally, the system temp directory should 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.
- The required level of WebSphere Application Server must be installed.
- Common Audit Service can be installed 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 Audit 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 Audit 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. The other components of a cluster, such as the managed nodes, HTTP server, and plug-ins, can be set up after the installation of Common Audit Service audit server, but must be set up before the configuration of Common Audit Service is begun.
- Common Audit Service has two separately installable components, which are referred to as features:
  - Audit Service (audit server, configuration utilities, and report setup utilities)
  - Audit Configuration Console (configuration console)
- You can install either or both features, however the following conditions apply:
  - If you are installing Common Audit Service for the first time *in the same profile*, select the default installation settings, which is 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 Audit 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 Audit Service using a root or non-root administrator ID.
  - Common Audit 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 Audit 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 Audit Service interactive installation wizard in either graphical or console (command line) mode.

The Common Audit Service 6.1 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 win32 platform. The corresponding commands for these files are described in this section.

You must use the Java Runtime Environment (JRE) version 1.5 to run Common Audit Service. WebSphere Application Server Version 6.1 includes JRE 1.5. Ensure that you have installed JRE 1.5, 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 1.5).

In the command window, set the environment variable:

```
JAVA_HOME=WAS_HOME/java
```

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

To run the installation wizard in interactive GUI or console mode, navigate 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:

- hp
- linux_i386
- linux_ppc
- linux_s390
- solaris
- usr/sys/inst.images

For Windows platforms:
windows/CARS

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 POWER
install_cars_srv_linuxppc [-console] [-is:javahome *java_home*]

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

For Linux on System z
install_cars_srv_linuxs390 [**-console**] [**-is:javahome** *java_home*]

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

For Windows
install_cars_srv_win32.exe [**-console**] [**-is:javahome** *java_home*]

For running the Java installation executable 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 will start. For a list of the configuration options to enter, see "Audit server installation options" on page 86.

-options-record *response_file*
Generate a response file 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_win32.exe -console
```

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

```
install_cars_srv_solaris
```

Interactive installation using the GUI panels

This section describes the interactive installation for Common Audit Service.

See "Interactive installation" on page 84 for the command you enter to begin the audit server installation. Run the command from the media that contains the audit server installation programs.

Proceed through the installation windows as follows:

1. Select the language that you want to use for the installation. The default is English.
2. Read the license agreement. Press **Next** if you agree with the license agreement, or press **Cancel** to exit the program.
3. The Welcome dialog is displayed, indicating that the installation will install the Common Audit Service component. Click **Next** to continue.
4. Select the path into which you want to install Common Audit Service. A default directory path is provided and is created if necessary. If a Common

Audit Service feature is already installed on this path then only uninstalled features are available for installation. Click **Next** to continue.

5. Select the features that you want to install. By default the Common Audit Service Server and Configuration Console are selected. If a selected feature is already installed in the specified installation path, that feature is not presented. Press **Next** to continue. The installation wizard searches for WebSphere Application Server installations that can be used as target locations to install Common Audit Service.
6. Specify the profile directory of the WebSphere Application Server instance into which you are deploying Common Audit 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. Refer to "Audit server installation options" for information on the default directory path. Press **Next** to continue. The installation wizard checks to determine if the specified installation directory already contains the product files. If product files are detected, you are prompted to specify a different target directory.
7. 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. Press **Next** to continue.
8. In the summary window, ensure that all the information that is shown is correct. If you need to make a change, press **Previous** to return to a previous window; otherwise, press **Next** to continue.
9. 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.

Audit server installation options

This section describes the Common Audit Service installation parameters.

Description

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

Table 24. Interactive installation options and values

Configuration option	Description

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

Table 24. Interactive installation options and values (continued)

Configuration option	Description
Feature selection	Specifies the separately installed features of the Common Audit 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 resides in the selected WebSphere Application Server node. It manages the configuration of Common Audit 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: **CARS6.1.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.
WebSphere Application Server Profile Directory	Specifies the directory path of the WebSphere Application Server profile where you are deploying Common Audit 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, the default path is defined as: *platform_dependent_install_path*/IBM/WebSphere/AppServer/profiles/default where *platform_dependent_install_path* is /opt for UNIX and Linux, and c:\Program Files for Windows platforms. If a WebSphere Application Server Network Deployment installation is detected, then the default profile is Dmgr0; 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.

Table 24. Interactive installation options and values (continued)

Configuration option	Description
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 Audit Service Version 6.1 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 containing all necessary parameters and values, run the interactive installation using the **-options-record** parameter. See "Starting the installation wizard" on page 84 for more information.

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 CD):

For AIX

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

For Linux on Power

install_cars_audit_srv_linuxppc -silent -options *response_file* [**-is:javahome** *java_home*]

For Linux on x86

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_win32.exe -silent -options *response_file* [**-is:javahome** *java_home*]

For running the Java installation executable 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_win32.exe -silent -options serverInstall.rsp
```

Enabling language support

This section describes how to enable language support.

The Common Audit Service Version 6.1 is translated into the following languages:

- Arabic
- Chinese (traditional)
- French
- German
- Japanese
- Czech
- Hebrew
- 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 reside.

To obtain language support for the Common Audit 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 Audit Service Version 6.1 audit server.

The installation media for the Common Audit Service Version 6.1 audit server contains the message catalogs for the various languages into which the audit server is translated.

Install the language packs using the following procedure:

1. Log on as root or as an administrative user.
2. Change to the installation root directory. Refer to the *Release Notes*, included in the product fix pack, for information on how to specify the installation root directory.
3. Change to the nls subdirectory.
4. Run either the interactive installation or console mode installation:
   - For interactive installation, run one of the following commands, depending on your platform:

     For AIX
     install_carslp_aix

     For Linux on POWER
     install_carslp_linuxppc

     For Linux on x86
     install_carslp_linux

     For Linux on System z
     install_carslp_linuxs390

     For Solaris
     install_carslp_solaris

     For Windows
     install_carslp_win.exe
   - For console mode installation, run the following command:

     java -cp carslp.jar run -console
5. Click **Next** to begin the installation.
6. Read the license agreement. If you agree with the terms of the license agreement, select to accept the terms and then click **Next**.
7. Select the language packages you want to install and click **Next**. A dialog showing the location and features of the languages that you selected is displayed. To accept the languages selected, click **Next**.
8. After installation has completed, click **Finish** to exit the wizard.

If it is necessary to uninstall the language support packages, see "Uninstalling language support packages" on page 124 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.

If you intend to customize the schema, it is recommended that you do so *before* you configure the XML data store (database) for Common Audit Service using the configuration console. In order to have a customized XML data store, you must create a new database during configuration. Consequently, the steps described in this topic are not applicable if you are upgrading from an older version of Common Audit Service to version 6.1 and you are using an existing audit database.

Before changing the XML store data definition language script, refer to the performance tuning information in the *IBM Tivoli Access Manager for e-business: Performance Tuning Guide*. Typically, you need to create larger table spaces for use by the XML store tables, and you need to use more than one container for a tablespace to take advantage of parallel I/O operations by DB2. You should also consider customizing the bufferspool 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.

Use the following procedure to customize the script:

1. Identify the changes to the basic data definition language script that you need to make to meet your custom storage needs. This includes identifying the parameters to be modified and the suitable values for those parameters to be set in the script.
2. Save a copy of the of the original data definition language script that was included in theCommon Audit 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 Audit Service.
3. Customize the script by editing the cr_dbobjects.db2 script using 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 will cause a failure in the configuration of the Common Audit Service Server. If a configuration error occurs after customizing the script, refer to the configuration logs to determine if 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 using the GUI configuration console, as described in Chapter 5, “Configuring the audit server,” on page 93.

Chapter 5. Configuring the audit server

This section describes how to configure the Common Audit 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.

Before configuring the audit server, perform the following actions:

1. 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 Audit Service audit server.
2. Ensure that the user credentials (user name and password) are part of the DB2 instance group, usually `db2grp1`.
3. The DB2 TCP/IP port number is required during the audit server configuration. The port number is found in the file /etc/services 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.
4. If the DB2 database is remote, ensure that the DB2 node for the remote database has been cataloged. Use the **db2 catalog** command as shown in "Installing the DB2 Administration Client on Linux and UNIX systems" on page 81.
5. 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 executables of the instance you will be using for Common Audit Service, for example, the default instance. If you are upgrading from a previous version of Common Audit Service to version 6.1, after cataloging the remote DB2 node, ensure that the audit database belonging to the previous version of Common Audit Service that is present on the remote DB2 node has also been cataloged in 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
   ```
6. On a Windows platform, ensure that the db2cmd.exe command is specified in the system PATH variable.
7. On AIX, if you plan to install DB2 9.1 Fix pack 2 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 Audit Service using the Integrated Solutions Console (ISC) module plug-in to the WebSphere Application Server administrative console.

See "Pre-configuration checklist for all platforms" for the items you need to consider before you start to configure the use of Common Audit Service.

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://websphereserver.ibm.com.:9060/ibm/console
2. Log in as a WebSphere Application Server administrator and select **Common Audit Service-> Audit Service Configuration** from the administrative menu. This selection starts the Common Audit Service configuration wizard. The options presented on each window of the wizard are described in "Common Audit Service configuration options" on page 95.
3. The Welcome dialog is displayed, indicating that Common Audit Service must be configured before the application can be used. Click **Next** to continue.
4. 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 Audit Service is installed. Click **Next** to continue.
5. 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. Click **Next** to continue.
6. 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 need to create a cluster on the target WebSphere Deployment Manager and restart the configuration.
7. In the **Audit Database** window, enter the following information to configure the database that is used by Common Audit Service. These options are described in "Common Audit Service configuration options" on page 95. Click **Next** to continue.
   - Database Instance Owner ID
   - Database Instance Owner Password
   - Database Instance Profile Path

     In case of a remote database configuration, on UNIX and Linux 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
8. In the **JDBC Connector** window, enter the following information to configure the JDBC driver that is used to connect to the database. Click **Next** to continue.
   - Database Server Host Name
   - Database Server TCP Service Port
   - JDBC Driver Path
9. Review the list of options you have 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.

10. Review the **Common Audit 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. 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 Audit Service using the configuration console.

Description

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

Table 25. 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.
Deployment target	Specifies the WebSphere Application Server deployment process where you want to deploy Common Audit Service.

Table 25. Interactive configuration values and options (continued)

Configuration option	Description
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 Linux and UNIX systems is `50001`.
JDBC driver path	Specifies the classpath for the JDBC driver JAR files (db2jcc.jar and db2jcc_license_cu.jar) that are used to connect to the Common Audit Service database (XML data store). Usually these JAR files are present in *DB2_INSTALL_ROOT*/java on UNIX and Linux 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 Audit Service event records that are stored in the XML data store.

Configuring JDBC resources in a clustered environment

This section provides post-configuration manual steps for configuring JDBC resources in a clustered environment. These steps should be performed *after* you configure Common Audit Service using the configuration console. In Common Audit Service Version 6.1, 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, perform the manual steps described below using the WebSphere Application Server Administrative Console of the containing Deployment Manager.

Determining the type of cluster

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

Common Audit Service Version 6.1 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, for example, Windows XP.
- 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, perform the manual steps described in "Configuring JDBC resources against a heterogeneous cluster" or "Configuring JDBC resources against a homogeneous cluster" on page 98 using the WebSphere Application Server Administrative Console of the container Deployment Manager.

Configuring JDBC resources against a heterogeneous cluster

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

Follow these instructions:

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-hand 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 running 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 already exist.
8. Initialize the DB2UNIVERSAL_JDBC_DRIVER_PATH variable to a value specifying 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 nodeagents 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 configuring JDBC resources against a homogeneous cluster.

Follow these instructions:

1. Restart all nodeagents 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.

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

1. 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 Audit 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.

Perform the following steps to configure a Web server that is installed on a cluster node:

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->Web servers->New**.
4. Use the wizard to create a new Web server, if one does not already exist.
5. In Step 1, complete the fields as follows, then click **Next**.

   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. In Step 2, select the IHS template radio button, then click **Next**.
7. In Step 3, complete the default fields, then click **Next**.

   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**.
8. In Step 4, confirm your specified settings in **Summary of actions**, then click **Finish**.
9. 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.

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). Refer to "Configuring the Web server plug-in for SSL" on page 131 for information on securing communication with the Web server using SSL.

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 Web server you have installed is stopped.
4. Use the plug-in installer from the WebSphere Application Server product image or disc 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), then click **Next**.
d. If the prerequisite check is passedt, click **Next**; otherwise correct the prerequisites and restart the installation.
e. Select the type of Web server you are configuring and click **Next**.
f. Select **Web server machine (remote)**, then click **Next**.
g. Accept the default location for the plug-ins installation root directory then click **Next**.
h. Browse for the configuration file of the Web server, then click **Next**.
i. Specify a name for the Web server. WebSphere Application Server will use this name to manage the Web server. Click **Next**.
j. Accept the default location for the plugin-cfg.xml file that is created on the Web server host, then click **Next**.
k. 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. Click **Next**.
l. Examine the summary information to ensure the specified settings are correct, then click **Next**.
m. Click **Next** on the pre-installation summary window to start installation.
n. If the post-installation window shows that the installation was successful, click **Next**; otherwise, correct any problems and reinstall the plug-in.
o. Close the installation roadmap and click **Finish** to exit the wizard.

   On UNIX or Linux 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 g, and *webserver_name* is the value specified in step i.
p. 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 (UNIX versus Windows), then you need to convert the plug-in configuration script as follows:
   a. On a UNIX 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 *web_server_machine_encoding* and the *application_server_machine_encoding*, respectively.
   b. **Before** moving from a UNIX 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 Linux or UNIX:
   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_name.sh
      ```

      Note: If the Web server host and WebSphere Application Server host support different operating systems (Unix-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-> Web servers-> *webserver_name*-> Remote Web server management**.
   e. Enter the information in each field and click **OK**.

      Port Specifies the HTTP Server administration server port number (default is 8008).

      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. Enter **Servers-> Web servers**.
   g. Select the check box of the *webserver_name* server. Click **Generate Plug-in** to update the WebSphere Application Server plug-in.
   h. 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:

1. Connect to the WebSphere Application Server Administrative Console on the deployment manager system:
   a. Expand **Servers**.
   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):

1. Connect to the WebSphere Application Server Administrative Console on the deployment manager system:
   a. Expand **Servers**.
   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 Audit Service application to Web server mapping

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

Mapping the Common Audit Service server to the virtual host

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

Follow these steps to map the Common Audit Service server to the virtual host:

1. In the WebSphere Application Server Administrative Console, click **Applications-> 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 Audit Service is operating in a WebSphere Application Server Network Deployment environment, select **Synchronize changes with Nodes** before saving the changes.

Mapping the Common Audit Service server to the target servers

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

Follow these steps to map the Common Audit Service server to the correct target servers:

1. In the WebSphere Application Server Administrative Console, click **Applications-> 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 have been updated against the Web Module.
7. Click **OK**.
8. Save your changes. If Common Audit 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 Audit Service

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

Verifying the configuration settings for the Common Audit Service application

Use the following procedures in the WebSphere Application Server Administrative Console to verify that the Common Audit Service application is configured correctly:

- **Determine that all modules are present:**
  1. Click **Applications-> 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-> Enterprise Applications-> CommonAuditService-> Session Management**.
  2. In the Configuration window, verify that the following general properties have values:

     Enable cookies
     Should be checked.

     Allow overflow
     Should be checked.

     Maximum in-memory session count
     Should be set to 1000 sessions.

     Set timeout
     Should be selected and set for 30 minutes.
  3. Click **Applications-> 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-> Enterprise Applications-> CommonAuditService-> JSP reload options for web modules**.
  6. Verify the following settings in the table:
     - Web Module=Common Audit Service
     - URI=cars-webservice.war, WEB-INF/ibm-web-ext.xmi
     - JSP enabled class reloading=enabled
     - JSP reload interval in seconds=10
  7. Click **Applications-> 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-> Enterprise Applications-> CommonAuditService-> EJB JNDI names**.
  2. Verify the following settings in the table:
     - EJB module=IBMCARSxmlstoreds-ejb
     - EJB=XmlStore
     - URI=IBMCARSxmlstoreds-ejb.jar,META-INF/ejb-jar.xml, Target Resource JNDI
     - Target Resource JNDI Name=ejb/com/ibm/cars/xmlstore/xmlstoreds/XmlStoreLocalHome
  3. The next two steps apply only to cluster configurations: Click **Applications-> 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
   Should be checked.

   Use replication settings from EJB container
   Should be enabled.

- **Ensure that the EJB references are configured:**
  1. Click **Applications-> 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 Audit Service data source

Use the following procedures 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.

- **Verify the JDBC provider:**
  1. Click **Resources-> JDBC-> JDBC Providers**.
  2. Ensure 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 Event Infrastructure
  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-> Data sources**.
  2. Ensure that the scope is set to the cluster or server where Common Audit 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 should be displayed 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 should be displayed 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 Audit Service data store

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

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

    ```
    . ~db2_instance_name/.profile
    db2 connect to audit_db_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 UNIX or Linux systems, from the command line enter:

    ```
    . ~db2_instance_name/.profile
    db2 connect to audit_db_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 Audit Service webservice component

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

- **Determine that the Web application is running:**
  1. Enter **Applications-> EnterpriseApplications**.
  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 webservice port:**

  Enter **Servers-> 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 in order 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 webservice to test the webservice 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 Audit Service server at the cluster scope, or at the node scope, then do the following steps to test the webservice 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 filename 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 belonging 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 than 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 appear in the list of certificates that is displayed.
     h. Click **OK**.
     i. Click **Close** to exit.
  3. Point the browser at the webservice to test the webservice 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.

Customized reports can also access the audit details Java stored procedure. Refer to "Creating custom reports" on page 161 for information on creating custom reports.

Before running a custom audit details report, the Java stored procedure must be deployed on the Common Audit 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 (Linux and UNIX) 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 Audit Service.)

To deploy the Java stored procedure, follow these steps:

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 109.
3. **All platforms:** Run the **ibmcarsddinst** script. See "Running ibmcarsddinst to deploy the Java stored procedure" on page 109.
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 110.

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.

To run the Java stored procedures or user-defined functions, the Linux run-time 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 will only look for the dependent libraries in the /usr/lib directory. Note that you must make the symbolic links on the machine running the DB2 server.

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

```
cd /usr/lib
ln -s JAVA_HOME/AppServer/java/jre/bin/libjava.so .
ln -s JAVA_HOME/AppServer/java/jre/bin/classic/libjvm.so .
ln -s JAVA_HOME/AppServer/java/jre/bin/libhpi.so .
ln -s JAVA_HOME/AppServer/java/jre/bin/libjsig.so .
ln -s JAVA_HOME/AppServer/java/jre/bin/libdbgmalloc.so .
```

```
ln -s JAVA_HOME/AppServer/java/jre/bin/libjitc.so .
ln -s JAVA_HOME/AppServer/java/jre/bin/libzip.so .
ln -s JAVA_HOME/AppServer/java/jre/bin/libxhpi.so .
```

where *JAVA_HOME* is *WAS_HOME*/AppServer/java and *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 1.5 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.

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.

Follow these steps to set the parameter:

1. Verify the existing jdk_path 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 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. Refer to 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 Audit Service configuration console does not include an option to perform this task.

Syntax

For Linux and UNIX

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 containing 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 Linux and UNIX 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.

To ensure that the IBMCARS_DD_REPORT Java stored procedure is deployed correctly, follow these steps:

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 using 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.

To ensure that the IBMCARS_EVENT_DETAIL Java stored procedure is deployed correctly, follow these steps:

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 using the following command:

   ```
   db2 terminate
   ```

Chapter 6. Upgrading the Common Audit Service audit server from earlier versions

This topic describes the considerations and procedures for configuring the Common Audit Service Version 6.1 audit server to use an existing copy of the audit server database.

Refer to Chapter 13, "Problem determination," on page 179 for information on troubleshooting problems with the upgrade procedures.

Considerations for upgrading the Common Audit Service audit server

Upgrade considerations

Consider the following points when preparing to upgrade from Common Audit Service Version 6.0 or version 6.0.1 to version 6.1:

- The XML data store (XMLSTORE database) that is used by earlier versions of the Common Audit Service audit server is the *only* product component that is upgraded to version 6.1. The database is upgraded if you specify to configure Common Audit Service Version 6.1 to use an existing XMLSTORE database that is being used by earlier versions of Common Audit Service.
- Versions of Common Audit Service that are prior to version 6.1 allowed only the root user to install the product on UNIX-based platforms, and allowed only the Administrator user to install the product on Windows platforms. Consequently, the upgrade procedure *must* be executed by a root user on UNIX-based platforms and by an Administrator user on a Windows platforms.
- Although Common Audit Service Version 6.1 supports both DB2 8.2 and DB2 9 Fix pack 2 or later, the DB2 server is *not* upgraded during an upgrade of Common Audit Service. Refer to the DB2 documentation for information on how to upgrade DB2 8.2 to DB2 9 Fix pack 2 or later.

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 6.1.
- 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 goals listed above are achieved by leveraging the following features in Common Audit Service Version 6.1:

- Common Audit Service Version 6.1 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).
- The configuration utility of Common Audit Service Version 6.1 allows you to configure the audit server to use an existing database.
- Common Audit Service Version 6.1 ships duplicate copies of the ConfigurRm.bat script on Windows platforms, and the ConfigureRm.sh script on Linux and UNIX 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 UNIX and Linux platforms, a previous installation of the product can be uninstalled without dropping the associated XMLSTORE database, thereby preventing an accidental removal of the XML data store.

Preparing to upgrade the Common Audit Service audit server

Perform the following tasks *before* you upgrade to version 6.1 of Common Audit Service.

- Back up the existing Common Audit Service XML data store (XMLSTORE database). The upgrade process alters the XMLSTORE database in a transaction mode; consequently, 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 following steps to back up the existing XMLSTORE database:

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

     ```
     db2stop force
      db2start
     ```

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

     For Linux and UNIX systems:

     ```
     mkdir /export/eventxml_bkup
     ```

     For Windows systems:

     ```
     mkdir \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:

     For Linux and UNIX systems:

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

     For Windows systems:

     ```
     chmod  a+w \export\eventxml_bkup
     ```

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

For Linux and UNIX systems:

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

For Windows systems:

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

- 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 Audit Service.
- Ensure that all conditions are met that are described in "Pre-installation checklist for all platforms" on page 82.
- If you are upgrading from Common Audit Service Version 6.0 or version 6.0.1 and it is deployed in a WebSphere Application Server Version 6.0 or version 6.0.1 cluster, set up the IBM HTTP Server Version 6.1 (using the WebSphere Application Server Version 6.1) 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.

Procedures for upgrading the Common Audit Service audit server from earlier versions

This topic describes the procedures for installing and configuring the Common Audit Service Version 6.1 audit server to use an existing earlier version of the audit server database.

Installing Common Audit Service Version 6.1 when upgrading to use an existing database

Follow the procedure described in "Interactive installation" on page 84 to install Common Audit Service interactively, or follow the procedure described in "Silent installation" on page 88 to install Common Audit Service silently in console mode using a response file.

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

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

Configuring Common Audit Service Version 6.1 to use an existing audit database

Follow the procedure that is described below to configure Common Audit Service Version 6.1 to use an existing XML data store (XMLSTORE database) that is being used by an older version of Common Audit Service.

1. Log into the administrative console of the WebSphere Application Server in which you have installed the **Common Audit Service Server** feature of Common Audit Service Version 6.1.
2. Select **Common Audit Service-> Tasks -> Audit Service Configuration** from the left-hand pane of the window to start the configuration wizard for Common Audit Service.

3. Click **Next** in the Welcome window to continue.
4. The Welcome dialog is displayed, indicating that Common Audit Service must be configured before the application can be used. Click **Next** to continue.
5. In the **Audit Service Management Endpoint** 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 Audit Service. Click **Next** to continue.
6. If administrative security is set ON in the target WebSphere Application Server process, enter the WebSphere administrator user name and password in the **Audit Service Management Authentication** window.
7. In the **Configuration target** window, select the configuration target.
8. 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 has cataloged the target remote DB2 server instance on which the target XMLSTORE database is located. On Windows platforms, you might not have to create a DB2 client instance to catalog the remote DB2 server instance. If this is the case, 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 6.1.

   Remote Database Node Name
   Specify a value only if the target XMLSTORE database is located on a remote DB2 server. If the XMLSTORE database is local, leave this field is 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.
9. In the **Create 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 UNIX and Linux platforms, and at *DB2_INSTALL_ROOT*\java on Windows systems.

10. Click **Next** in the **Configuration summary** window to start the configuration of Common Audit Service Version 6.1 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 have successfully configured Common Audit Service Version 6.1 to use the existing lower version of the XMLSTORE database. Additionally, the target lower-versioned XMLSTORE database has been upgraded to version 6.1. 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 Audit Service to start sending events to Common Audit Service Version 6.1, and to ensure that the upgraded existing XMLSTORE database is not accidentally dropped during uninstallation of the older version of Common Audit 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 you have successfully completed the procedures that enable the Common Audit Service Version 6.1 audit server to use the existing, older-version XMLSTORE database, follow the procedures that are described below to prevent an unintended loss of data, to enable your application clients to begin sending events over the new audit server port, and to uninstall the old version of Common Audit Service.

1. Replace the database removal script of the earlier lower version of Common Audit Service with the identically named script that is shipped with Common Audit Service Version 6.1.

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

   For Linux and UNIX systems, replace the old dbConfigureRm.sh script that is in the following directory:

   was60_profile_path/event/dbscripts/eventxml

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

   CARS_HOME/server/etc/upgrade

   For Windows systems, replace the old dbConfigureRm.bat script that is in the following directory:

   was60_profile_path\event\dbscripts\eventxml

   with the new dbConfigureRm.sh 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 Audit Service.

- If the Common Audit Service audit server was upgraded in a stand-alone server environment, perform the following steps:
  a. Log in to the WebSphere Application Server Administrative Console.
  b. Select **Application Servers-> server1-> ports**.
  c. Identify the application port, which is the WC_defaulthost entry in the table.
  d. Stop and restart all versions (old and new) of the Common Audit Service audit server.
  e. Reconfigure one or more Common Audit Service client applications to send audit events to the new application port. The old and new versions of the Common Audit Service audit server will continue writing to the database until all of the clients are configured to use only the new application port and the new server.
- If the Common Audit 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 should not be necessary.
  a. Stop all versions of Common Audit 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 Audit Service audit server. This will automatically stop the audit servers on both of the clusters.
  c. Stop the IBM HTTP Server Version 6.1 that is configured for use with the WebSphere Application Server 6.1 cluster, and stop and IBM HTTP Server Version 6.0 that is configured for use with the WebSphere Application Server 6.0 cluster.
  d. Reconfigure the IBM HTTP Server Version 6.1 that is configured to be used as a load-balancer for the WebSphere Application Server 6.1 cluster to listen on port 80, then restart the same HTTP server. The Common Audit Service Version 6.1 audit server should now be the audit server that stores events in the audit database.

3. Uninstall the *older* version of Common Audit Service using the uninstallation instructions that are provided in the *Auditing Guide* that is supplied with the exploiting product.

After you successfully complete the above procedures, the upgrade to Common Audit Service Version 6.1 is finished.

Chapter 7. Unconfiguring Common Audit Service

This topic describes how to unconfigure the Common Audit Service Version 6.1 audit server, configuration console, and configuration utilities using the Integrated Solutions Console (ISC) module plug-in to the WebSphere Application Server Administrative Console. *NOTE: You must unconfigure Common Audit Service before you uninstall it.*

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://websphereserver.ibm.com:9060/ibm/

3. Log in as a WebSphere Application Server administrator and start the unconfiguration wizard. 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 95.
4. The Welcome dialog is displayed, indicating that Common Audit Service must be unconfigured before the application can be uninstalled. Click **Next** to continue.
5. 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 Audit Service will be unconfigured. Click **Next** to continue.
6. 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. Click **Next** to continue.
7. In the **WebSphere Target Mapping** window, select the path of the WebSphere Application Server deployment target where Common Audit Service is deployed. The list of clusters and independent servers that are available for undeployment are displayed in the drop-down list. You must select an entry from the list. Click **Next** to continue.
8. 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. Click **Next** to continue.

9. Review the list of options you have 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.
10. Review the **Common Audit Service Status** window to determine the outcome of the unconfiguration. If the unconfiguration was unsuccessful, the problems should be corrected and the unconfiguration started again from the Welcome window. Click **OK** to return to the Welcome window.

Chapter 8. Uninstalling Common Audit Service

This topic describes how to uninstall the Common Audit Service Version 6.1 audit server, configuration console, and configuration utilities.

The uninstallation of a Common Audit Service feature involves the following tasks:

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

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

If an uninstallation of the server fails, you must perform the steps that are described in "Failed uninstallation workarounds" on page 188 in order to remove the product from your system.

Uninstallation checklist for all platforms

This topic lists the tasks that you must perform before you attempt to uninstall Common Audit Service.

Before you start the uninstallation wizard to remove either feature (audit server or configuration console) of Common Audit 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 Audit Service from the WebSphere Application Server profile (select to keep the database intact).

If you want to completely remove Common Audit 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 Audit Service is not fully unconfigured before starting uninstallation, a warning message will be displayed in the uninstallion window informing you to completely unconfigure the Common Audit Service components before you uninstall the Common Audit Service. If you continue with the uninstallation, you will have 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 188.

After you have successfully undeployed Common Audit 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 Audit Service interactive uninstallation wizard in either graphical or console (command line) mode.

Before you begin

Ensure that you have undeployed Common Audit Service from the WebSphere Application Server; refer to "Uninstallation checklist for all platforms" on page 121 for more information on undeploying Common Audit 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
   - **Linux or UNIX:** /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 one of the following commands:

For Windows

Use one of the following commands:

uninstall.exe [**-console**] [**-is:javahome** *java_home*]

java -cp uninstall.jar run [**-console**] [**-options-record**]

For Linux or UNIX

Use one of the following commands:

uninstall.bin [**-console**] [**-is:javahome** *java_home*]

java -cp uninstall.jar run [**-console**] [**-options-record**]

Parameters

-console

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

-options-record *response_file*

Generate a response file using 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 using console mode:

```
uninstall.exe -console
```

Interactive uninstallation using the GUI windows

This topic describes the interactive uninstallation of Common Audit Service using the GUI windows.

See "Starting the uninstallation wizard" on page 122 for the command you enter to begin the uninstallation wizard.

Proceed through the windows as follows:

1. Select the language that you want to use for the installation and click **OK**.
2. The Welcome dialog is displayed. Click **Next** to continue.
3. In the **Features** window, select both features to uninstall the audit server, configuration console, and configuration utilitiies. Select **Common Audit Service** to uninstall the audit server and configuration utilities only. Select **Common Audit Service Configuration Console** to uninstall only the configuation console. Click **Next** to continue.
4. 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. Click **Next**to continue.
5. In the **Summary** window, check that the location of the selected features for uninstallation are correct. Click **Back** if you need to change a setting. Click **Next** to begin the uninstallation.
6. The final window shows that the uninstallation was successful or indicates error logs to identify any uninstallation problems.

Silent uninstallation

This topic describes the 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 will be displayed at any time 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
   - **Linux or UNIX:** /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, use one of the following commands:

For Windows

Use one of the following commands:

uninstall.exe -silent -options *response_file*

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

For Linux or UNIX

Use one of the following commands:

uninstall.bin -silent -options *response_file*

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:

```
uninstall.exe -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

This topic describes how to uninstall language support packages for the Common Audit Service.

Uninstall the language packs using the following procedure:

1. Change to one of the following directories:
   - On Linux and UNIX operating systems:
     `opt/CARSLP/lp_uninst`
   - On Windows operating systems:
     `C:\Program Files\CARSLP\lp_uninst`
2. Uninstall the language support packages using the following command:
   java -jar cars_lp_uninstall.jar

Chapter 9. Securing data flow in the operating environment

This topic describes how to secure the exchange of data among components in the Common Audit 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, refer to ″Securing C client events.″

Securing Web service client events

Enabling security for event logging from the Common Audit Service 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 Audit Service Web service roles to users or groups.

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:

1. Use the WebSphere Application Server administrative console to configure the desired user registry. The following user registries are available for configuration:
   - Local operating system. See “Configuring the operating system registry” on page 126 for instructions.
   - LDAP. See “Configuring the LDAP registry” on page 126 for instructions.
   - Custom. See “Configuring a custom registry” on page 127 for instructions.
   - Federated. See “Configuring a federated registry” on page 128 for instructions.
2. Enable the administrative and application security option for the desired user registry that you configured, using the following steps:
   a. Click **Security** → **Secure administration, applications, and infrastructure** → **Enable administrative security**.
   b. Click **Security** → **Secure administration, applications, and infrastructure** → **Enable application security**.
   c. In **Available realm definitions**, select the user registry that you have configured (for example, `Lightweight Directory Access Protocol (LDAP) user registry`).
   d. Click **Security** → **Secure administration, applications, and infrastructure** → **Set as current**. This selection forces validation of any properties that are configured for the selected realm.
   e. 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 saving the changes.

3. Manually add the security policy for the DB2 JDBC driver, see Configuring security policy for the JDBC provider.
4. Enable the Java 2 security option with the following steps:
   a. Click **Security** → **Secure administration, applications, and infrastructure** → **Use Java 2 security to restrict application access to local resources**.
   b. Click **Security** → **Secure administration, applications, and infrastructure** → **Warn if applications are granted custom permissions**.
   c. Click **Security** → **Secure administration, applications, and infrastructure** → **Restrict access to resource authentication data**.
5. Click **Apply** and save the changes.

Configuring the operating system registry:

In a clustered environment it is recommended that you use an LDAP registry (or a federated repository that includes an LDAP registry) in order to maintain consistency of the registry between nodes in the cluster. Note that you should use an LDAP registry if the Network Deployment cell (all of the nodes) is not located on a single machine, or where the WebSphere Application Server is running on UNIX as a non-root user.

From the WebSphere Application Server administrative console, configure the local operating system registry settings:

1. Click **Security** → **Secure administration, applications, and infrastructure** → **Available realm definitions**.
2. Select **Local operating system** from the drop-down list.
3. Click **Security** → **Secure administration, applications, and infrastructure** → **Configure**.
4. Specify the Primary administrative user name property, which is a user with administrative privileges who is defined in the local operating system.
5. Either click the Automatically generated server identity button (recommended for WebSphere Application Server Version 6.1 or higher), or click the Server identity radio button 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`)
6. Click **OK** and then save the changes.

Configuring the LDAP registry:

The Lightweight Directory Access Protocol (LDAP) user registry is used when users and groups are located in an external LDAP directory. In a clustered environment it is recommended that an LDAP registry be used because of the need to maintain consistency of the registry between nodes in the cluster.

From the WebSphere Application Server administrative console, configure the LDAP registry settings:

1. Click **Security** → **Secure administration, applications, and infrastructure** → **Available realm definitions**.
2. Select **Standalone LDAP registry** from the drop-down list.
3. Click **Security** → **Secure administration, applications, and infrastructure** → **Configure**.

4. Specify the Primary administrative user name property, which is a name of a user in your LDAP registry who has administrative privileges.
5. Either click **Automatically generated server identity** (recommended for WebSphere Application Server Version 6.1 or higher), or click **Server identity** and specify the server user ID and password to access the 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`).
6. Specify the following properties:
   - Type of LDAP server (for example, `IBM Tivoli Directory Server`)
   - Host (for example, `server1.tivlab.austin.ibm.com`)
   - Port (for example, `389`)
   - Base distinguished name (for example, `ou=tivoli,o=ibm,c=us`)
   - Bind distinguished name (for example, `cn=root,ou=tivoli,o=ibm,c=us`)
   - Bind password (for example, `abc26xyz`)
   - Search timeout (for example, `120`)
   - Reuse connection (recommended to use the default which is ″enabled″). This property prevents the LDAP connection from reestablishing on each LDAP access.
   - Enable the Ignore case for authorization property if your LDAP server requires it.
7. Select the **SSL enabled** option 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 saving 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.

To enable security in a WebSphere Application Server Network Deployment environment using an LDAP user registry, you need to configure LTPA as the active authentication protocol to authenticate the users. See “Configuring the LTPA authentication mechanism” on page 130 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:

1. Click **Security** → **Secure administration, applications, and infrastructure** → **Available realm definitions**.
2. Select **Standalone custom registry** from the drop-down list.
3. Click **Security** → **Secure administration, applications, and infrastructure** → **Configure**.

4. Specify the Primary administrative user name property, which is a name of a user in your custom registry who has administrative privileges.
5. Either click **Automatically generated server identity** (recommended for WebSphere Application Server Version 6.1 or higher), or click **Server identity** 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`).
   - Custom registry class name (for example, `com.ibm.websphere.security`)
   - Enable the Ignore case for authorization property if your custom class requires it.
6. Click **OK** and save your changes. If you are in a WebSphere Application Server Network Deployment environment, select **Synchronize changes with Nodes** before saving the changes.

Configuring a federated registry:

A federated registry allows consolidation of multiple repositories into a single virtual registry.

From the WebSphere Application Server administrative console, configure the federated registry settings:

1. Click **Security** → **Secure administration, applications, and infrastructure** → **Available realm definitions**.
2. Select **Federated repositories** from the drop-down list.
3. Click **Security** → **Secure administration, applications, and infrastructure** → **Configure**.
4. Specify the Realm name which will apply to this collection of registries.
5. Specify the Primary administrative user name property, which is the name of a user in one of the federated registries who has administrative privileges.
6. Either click **Automatically generated server identity** (recommended for WebSphere Application Server Version 6.1 or higher), or click **Server identity** 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`).
   - Enable the **Ignore case for authorization** property if case sensitivity is not important.
7. Use the Repositories in the realm table to manage the repositories that you want federated. Optionally, you can add the built-in file repository as well as any external LDAP registries.
8. Click **OK** and save your changes. If you are in a WebSphere Application Server Network Deployment environment, select **Synchronize changes with Nodes** before saving 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 using the configuration console. To grant access permissions to the JAR file, add the appropriate grant policy to the app.policy file that is located

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.

Use the following steps to grant permissions to the JAR file:

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 using the following command:

   ```
   wsadmin>set obj [$AdminConfig extract
    cells/cell_name/node/node_name/app.policy
   temp_file_path/app.policy]
   ```

   The *node_name* value should be for a CellManager node if you are using Deployment Manager.

   Ensure that *temp_file_path* exists prior to running the command.

   Use either a single forward slash (/) or a double back slash (\\) while specifying 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 you need to 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 using 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. Follow these steps to update the policy for the JDBC provider.

1. Start the Policy Tool. On UNIX and Linux platforms, use the following command:

   ```
   was_install_root/java/jre/bin/policytool
   ```

   On Windows platforms, use the following command:

   ```
   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 new 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. Click **OK**.
9. In the Policy entry window, the string `permission java.security.AllPermission` is displayed beneath the Permission buttons. Click **Done**.
10. Click **File-> Save** to save the updated file.
11. 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.

1. Click **Security** → **Secure administration, applications, and infrastructure** → **Authentication mechanisms and expiration** → **Key set groups**.
2. 2. Under Key Generation:
   a. Check the **CellLTPAKeySetGroup** 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 saving the changes.
5. Click **Secure administration, applications, and infrastructure**.
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 saving the changes.

Restarting the cluster:
From the WebSphere Application Server administrative console, restart the cluster to enable the global security method using these steps:

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**.

From this point on, you must 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:
From the WebSphere Application Server Administrative Console, use the following steps to configure WebSphere Application Server Secure for Secure Sockets Layer (SSL) authentication:

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 Audit 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 saving the changes.
2. Now you must configure SSL between the WebSphere Application Server and the Web service client. To do this, 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 on the cluster or server name into which Common Audit 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 key store 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 saving 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. 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. Security is implemented using the SSL protocol as follows:

Perform the following steps to set up SSL security for communication between the Web server and the Web server plug-in that is configured in a WebSphere Application Server cluster. Refer to "Configuring a Web server that is installed on a system outside the cluster" on page 99 for information on configuring the Web server plug-in.

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 filename (or accept the default name) in the following field:

   Plug-in key store 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 key store 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 key store.
7. Click **Apply** to save the password changes.
8. Under **Additional Properties**, you can also select the following:

   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 key store.
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 key store 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.

The Common Audit Service Web service client can invoke the Common Audit 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. Refer to the appropriate Web server documentation for details on how to enable SSL.

Follow these steps:

1. Use the IBM HTTP Server IKEYMAN utility to create a CMS key database file and insert the server's personal certificate.

   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 need to 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's certificate in the key database file. It is not needed if the default certificate in the keyfile is used. 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. 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 Audit Service defines three roles called EventSource, eventAdministrator, and eventCreator. The EventSource role defines the source of events to be submitted to Common Audit Service. The eventAdministrator role provides access to the EventAccess and Emitter interfaces. The eventCreator role restricts access to event submission. For more information on configuring the use of security roles, refer to the WebSphere Process Server Information Center.

All roles can be mapped to a user or group using the WebSphere Application Server Administration Console:

1. Click **Applications** → **Enterprise Applications** → **CommonAuditService** → **Security role to users/group mappings**.
2. Select the **EventSource** role.
3. Click on one of the following:
   - **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. If the user registry is Local Operating System, then select `root`, for example.
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 saving the changes.

The All-Authenticated and Everyone meta-groups override any mapped users or groups; therefore, ensure that you clear these meta-groups when mapping a specific user or group.

Securing the XML data store

The XML data store contains audit information and, therefore, must be secured using the security mechanisms provided by DB2 so that only authenticated users with correct privileges access the data.

Chapter 10. Running the server utilities

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

When deploying the Common Audit 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.

On Linux and UNIX systems, set the CLASSPATH variable to include the following file paths:

```
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 Audit Service server. By default, the location is /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.

On Windows systems, set the CLASSPATH variable as:

```
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 Audit 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 142.

- **-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, 2007 at 10:00 PM through October 6, 2007 at 10:00 PM:

```
java com.ibm.cars.staging.Staging -mode historical
  -starttime "Jan 1, 2007 10:00:00 PM GMT"
  -endtime "Oct 6, 2007 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, 2006 at 12:00 AM:

```
java com.ibm.cars.staging.Staging -mode prune
  -prunetime "Oct 6, 2006 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) will very likely fail. If you need more parallelism, increase the number of workers instead of running another instance of the staging utility.

Return codes

In the event of a fatal 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. The recommended procedure is to 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 will 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 142.

- **-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 prior to 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, 2006 5:30:00 AM
- The last date contained in the tables to be archived. For example: Jan 2, 2006 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 performs 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, wherein 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 drill-down reporting. Prior to running the postarchive operation, use the staging utility prune operation to remove the report table data for audits ranging within the begin date and the end date as provided by the prearchive operation. See "Running the staging utility command" on page 136.

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 performs the prearchive operation:

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

The following command performs 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 performs 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 located in *CARS_HOME*\server\etc on Windows and *CARS_HOME*/server/etc on Linux and UNIX systems, where *CARS_HOME* is the installation directory of Common Audit Service.

Sample

Following is a sample ibmcars.properties file:

```
#
# Licensed Materials - Property of IBM
# 5748-XX8
# (c) Copyright International Business Machines Corp. 2004
# 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; this should be fine 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, 2004 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, 2004 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, 2007 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, 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 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, 2007 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
#  results 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 behaviour).
#  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 Audit 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 Audit 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 resides.

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 perform work in parallel. The recommended value for best performance is the number of CPUs of the machine containing 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 perform for the XML data store utilities:

Pre-archive

Use prior to 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, 2006 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, 2005 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 11. Federated Identity Manager reports

This topic describes the software and tasks required to run the out-of-box Federated Identity Manager reports, and describes tasks that are required to create custom reports using Federated Identity Manager audit event data.

Tivoli Federated Identity Manager provides two out-of-box reports, Administrative Events Report and Single Sign-On Summary Report, that demonstrate how audit event data is captured, filtered, formatted, and viewed. These reports are not intended to meet the reporting needs of your organization; you can create custom reports to meet your organization's reporting requirements. For information on the contents of the Administrative Events Report and Single Sign-On Summary Report, refer to "Federated Identity Manager reports" on page 160.

Tivoli Federated Identity Manager requires the installation and configuration of Common Audit Service to produce formatted audit event data that can be included in the out-of-box reports. The Common Audit Service tools enable Federated Identity Manager to:

- Send XML event data to an audit server that securely manages storage of the data in an XML data store.
- Specify which audit event details to stage from the data store into reporting tables.
- Stage audit information into formatted tables that are read by a report viewing tool.

You can generate Federated Identity Manager reports using any of the following utilities:

Federated Identity Manager command line interface (CLI)
: You can generate out-of-box and custom reports using CLI commands. Report output can be formatted in PDF or HTML.

Federated Identity Manager console
: You can generate out-of-box and custom reports using the product console. Report output can be formatted in PDF or HTML.

Tivoli Common Reporting console
: You can generate and run out-of-box and custom reports using Tivoli Common Reporting. Tivoli Common Reporting is a reporting environment that provides a consistent approach across IBM products for viewing and administering reports. Tivoli Common Reporting consists of several components:

  - Data store for storing and organizing report designs, reports, and supporting resources.
  - Web-based user interface for generating and viewing formatted reports. The interface allows users to specify report parameters and other report properties.
  - Command-line interface for working with objects in the data store and for performing additional administrative functions.

  Tivoli Federated Identity Manager Version 6.2 provides a *report package* that enables you to exploit the features of Tivoli Common Reporting. The report

package is a zip file containing the definitions, designs, and the report set hierarchy that are needed for you to run out-of-box operational reports using Tivoli Common Reporting.

You can create custom reports using the Business and Intelligence Reporting Tools (BIRT). BIRT is an open-source set of runtime report utilities that run in an Eclipse environment. These tools enable you to:

- Generate custom reports using the BIRT runtime report engine.
- Create or modify custom report designs, using the open-source BIRT report designer.

All of the utilities described above are provided with Tivoli Federated Identity Manager. The steps required to create reports with these utilities are documented in this auditing guide and in the *Tivoli Common Reporting User's Guide*. Additional instructions on using BIRT to customize reports are described at the following Web site:

http://www.ibm.com/developerworks/tivoli/library/t-tcr/ibm_tiv_tcr_customizing_report_designs.pdf

Before you attempt to use these tools to view the sample reports, or to create and view custom report data, read the concepts and configuration tasks described in the following subtopics to ensure that the audit event data is prepared for report generation.

Required reporting setup tasks

This topic describes the prerequisite steps that you follow to set up your environment to run Federated Identity Manager reports. Finish the tasks in this topic before you attempt to run the reports that are supplied with the product. You must also complete these steps before you create and generate custom reports.

Perform the following setup tasks before you run and view Federated Identity Manager audit report data that is captured and managed using Common Audit Service.

1. Install Federated Identity Manager and create a domain using the Federated Identity Manager console. Refer to the *IBM Tivoli Federated Identity Manager Installation and Configuration Guide* for information on installing Federated Identity Manager and creating a domain.
2. Deploy the Federated Identity Manager runtime into WebSphere Application Server using the Federated Identity Manager console. Refer to the *IBM Tivoli Federated Identity Manager Installation and Configuration Guide* for information on deploying Federated Identity Manager into WebSphere Application Server.
3. Install and configure Common Audit Service on a system that is separate from the system on which you install Federated Identity Manager. The configuration of Common Audit Service sets up the audit server to receive audit events from Federated Identity Manager. Refer to Chapter 4, "Installing Common Audit Service," on page 79 and Chapter 5, "Configuring the audit server," on page 93 for information on configuring Common Audit Service.
4. Begin emitting audit events to the Common Audit Service by performing the following tasks:
   a. Enable auditing in Federated Identity Manager using the Federated Identity Management console. Refer to "Configuring Federated Identity Manager auditing settings to use Common Audit Service" on page 3 for instructions.

b. Select **Audit Events** in the portfolio of the Auditing window to display the types of events for which you want to capture audit data. Refer to "Selecting the events you want to audit" on page 4 for instructions on selecting types of audit events.

   Note:

   - The sample out-of-box reports require the capture of the IBM_SECURITY MGMT_POLICY and IBM_SECURITY_MGMT_AUDIT events in order to provide security management data.
   - Custom reports can use the same events as the out-of-box reports, as well as any of the other audit event types.

   The following guidelines apply to generating the audit events that are associated with a specific group of events that you select in the console:

   - To generate IBM_SECURITY_MGMT_POLICY events, you can create or modify a federation.
   - To generate IBM_SECURITY_MGMT_AUDIT events, modify one or more audit settings in the console. For example, deselect an event such as Trust Service Modules, click **Apply ->Load changes to Tivoli Federated Identity Manager runtime**; then return to the Audit Events window and select **Trust Service Modules**, click **Apply ->Load changes to Tivoli Federated Identity Manager runtime**.
   - The Single Sign-on Summary Events Report uses IBM_SECURITY_AUTHN events.
   - To generate IBM_SECURITY_AUTHN events, you first need to set up a federated single sign-on environment and complete single sign-on transactions.
   - The Administrative Events Report uses IBM_SECURITY_MGMT_POLICY and IBM_SECURITY_MGMT_AUDIT events.

c. Click **Apply ->Load changes to Tivoli Federated Identity Manager runtime**.

5. Stage the audit event types, IBM_SECURITY_MGMT_POLICY, IBM_SECURITY_MGMT_AUDIT, and IBM_SECURITY_AUTHN, which are used to generate the out-of-box Federated Identity Manager reports. (You can also stage these events as well as the other audit event types to create custom reports.)

   Perform the following tasks to stage (set up) the Federated Identity Manager report data:

   a. For the out-of-box reports (Administrative Events Report and the Single Sign-on Summary Events Report), copy the following data definition language (DDL) files (located under *FIM_INSTALL_ROOT*/reports/cars_files) to a corresponding directory on the system where the Common Audit Service server is installed (for example, /opt/IBM/FIM/reports/cars_files).

      - cars_t_mgmt_audit_custom.ddl
      - cars_t_mgmt_policy_custom1.ddl
      - cars_t_mgmt_policy_custom2.ddl
      - cars_t_authn_custom.ddl

      Note: The other Common Audit Service DDL files in this directory are not used by the out-of-box reports; the other DDL files can be used to create custom reports. Refer to "Federated Identity Manager DDL contents" on page 171 for information on each DDL file.

b. To create the out-of-box report tables, run the following commands for the corresponding DDL files:

```
db2 connect to db_name user DB2_username using DB2_password
db2 -tsf cars_t_mgmt_audit_custom.ddl
db2 -tsf cars_t_mgmt_policy_custom1.ddl
db2 -tsf cars_t_mgmt_policy_custom2.ddl
db2 -tsf cars_t_authn_custom.ddl
db2 disconnect db_name
```

Example:

```
db2 connect to eventxml user db2inst1 using mypassword
db2 -tsf cars_t_mgmt_audit_custom.ddl
db2 -tsf cars_t_mgmt_policy_custom1.ddl
db2 -tsf cars_t_mgmt_policy_custom2.ddl
db2 -tsf cars_t_authn_custom.ddl
db2 disconnect eventxml
```

To create custom report tables, run the db2 command for the appropriate DDL file.

c. Navigate to the *FIM_INSTALL_ROOT*/reports/cars-files on the Federated Identity Manager host and make a backup copy of the CARSShredder.conf file that is installed with Federated Identity Manager. Also make a backup copy of the CARSShredder.conf file that is installed on Common Audit Service server host system.

Note: Ensure that you back up the original CARSShredder.conf file on the Federated Identity Manager host. Federated Identity Manager provides this version of the CARSShredder.conf file for the Federated Identity Manager reports; however, you need to back it up for use in creating custom reports for your environment. For more information on using the CARSShredder.conf file to create custom reports, refer to "Working with the CARSShredder.conf configuration file" on page 163.

d. Copy the CARSShredder.conf file that is provided for Federated Identity Manager reports from *FIM_INSTALL_ROOT*/reports/cars-files to the *CARS_HOME*/server/etc directory on the host where the Common Audit Service server is installed.

e. Perform the following tasks to run the staging utility (which uses the CARSShredder.conf file) to stage the audit events into the database tables on the Common Audit Service server:

1) Set the CLASSPATH variable on the Common Audit Service server as follows:

Linux or UNIX systems

Set CLASSPATH to:

CARS_HOME/server/etc: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:

Windows systems

Set CLASSPATH to:

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;

2) Run the staging utility on the Common Audit Service server to stage the event data for reporting. The following example demonstrates running the staging utility command in historical mode:

```
java com.ibm.cars.staging.Staging -mode historical -starttime
"Jan 1, 2008 00:00:00 AM GMT" -endtime "Dec 31, 2020 12:59:59 PM
GMT" -progress 1 -batchsize 1 -dbusername db2inst1 -dbpassword
mydb2password
```

Refer to Chapter 10, "Running the server utilities," on page 135 for more information on running the staging utility.

After you complete the tasks described above, you are ready to run and view the two sample reports supplied with Federated Identity Manager using the following components:

- Federated Identity Manager console (see "Running the out-of-box Federated Identity Manager reports using the console" on page 156)
- Federated Identity Manager command line interface (see "Running Federated Identity Manager reports using the command line interface")

You can also generate and view the Federated Identity Manager reports using Tivoli Common Reporting. Refer to "Importing the Federated Identity Manager out-of-box reports into the Tivoli Common Reporting environment" on page 156 and "Running the out-of-box Federated Identity Manager reports using the Tivoli Common Reporting console" on page 158 for information on setting up and using Tivoli Common Reporting to generate and manage report data.

Running Federated Identity Manager reports using the command line interface

This topic describes how to run out-of-box and custom Federated Identity Manager reports using the command line interface.

Use the following procedure to generate a Federated Identity Manager report containing the audit event data that is captured and stored using Common Audit Service.

1. Perform the setup tasks described in "Required reporting setup tasks" on page 150.
2. If you are running a custom-designed report, put the custom report in the *FIM_INSTALL_ROOT*/report/designs directory (for example, /opt/IBM/FIM/reports/designs). If you are running an out-of-box Federated Identity Manager report, go to the next step.
3. Run the wsadmin command on the system where the Federated Identity Manager runtime is installed to execute the CLI commands for the Federated Identity Manager reports. You must run the wsadmin command for the out-of-box reports, and for custom reports that you create. For example, if your WebSphere Application Server profile is AppSrv01, issue the following command:

   Windows systems: /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/bin/wsadmin.sh

UNIX and Linux systems: \opt\IBM\WebSphere\AppServer\profiles\AppSrv01\bin\wsadmin.bat

4. Create a response file by issuing the following command:

```
$AdminTask runItfimReport {-operation createResponseFile
  -fimDomainName your_domain_name
  -reportDesign name_of_report_design
  -fileId response_file_path}
```

5. Edit the response file that is created and specify the necessary values (see "Example Response File to Run a Report using the CLI" below).
6. Run the report by issuing the following command:

```
$AdminTask runItfimReport {-operation run
  -fimDomainName your_domain_name
  -reportDesign name_of_report_design
  -fileId response_file_path
  -renderType pdf | html}
```

Samples of the runItfimReport command:

Create Report

```
$AdminTask runItfimReport
  {-operation createResponseFile
  -fimDomainName localhost-server1
  -reportDesign administrative_events.rptdesign
  -fileId /dnload/my_response_file.txt}
```

Run report as PDF

```
$AdminTask runItfimReport
  {-operation run
  -fimDomainName localhost-server1
  -reportDesign administrative_events.rptdesign
  -fileId /dnload/my_response_file.txt
  -renderType pdf}
```

Run report as HTML

```
$AdminTask runItfimReport
  {-operation run
  -fimDomainName localhost-server1
  -reportDesign administrative_events.rptdesign
  -fileId /dnload/my_response_file.txt
  -renderType HTML}
```

7. To view the contents of the report, navigate to the *FIM_INSTALL_ROOT*/reports/archives directory (for example, /opt/IBM/FIM/reports/archives) and open the output file using an appropriate viewer. Executed (archived) reports are assigned filetypes of *.pdf or *.html, depending on the value assigned to the -renderType parameter.

Refer to the *Tivoli Federated Identity Manager Administration Guide* for more information on the runItfimReport command.

Listing available reports using the command line interface

This topic describes the commands you can use to view a list of available Federated Identity Manager reports.

You can also run the following commands to determine what reports are available for running and viewing:

List Runnable Reports

List the reports that you can run:

```
$AdminTask listItfimRunnableReports
 {-fimDomainName your_domain}
```

List Running Reports

Lists the reports that are running currently running:

```
$AdminTask listItfimActiveReports
{-fimDomainName your_domain}
```

List Archived Reports

Lists the archived (executed) reports that you can run again:

```
$AdminTask
listItfimArchivedReports {-fimDomainName your_domain}
```

You can run these commands interactively by specifying the -interactive parameter:

```
$AdminTask command_name -interactive
```

Refer to the *Tivoli Federated Identity Manager Administration Guide* for more information on using the report commands.

Sample response files for running a report using the command line interface

This topic provides samples to demonstrate how response files can be used to run Federated Identity Manager reports using the command line interface.

Below are contents from a sample response file that was generated for the Administrative Events Report. The file contains the name-value pairs that represent the parameters for which you need to specify values in order to run a report.

```
#Response File generated for Administrative Events Report
#Wed Nov 28 00:14:52 CST 2007
Administrator=Enter a value here
Date_Range_Start=Enter a value here
Event_Type=Enter a value here
Date_Range_End=Enter a value here
```

Below is an edited response file for the Administrative Events Report. Ensure that you enter the correct values that apply to your Common Audit Service server.

```
#Response File generated for Administrative Events Report
#Wed Nov 28 00:14:52 CST 2007
Administrator=All User IDs
Date_Range_Start=07/01/2007 12:00:00 AM
Event_Type=All Events
Date_Range_End=08/31/2007 12:00:00 AM
```

Below are contents from a sample response file that are generated for the Single Sign-On Summary Report. The file contains the name-value pairs that represent the parameters for which you need to specify values in order to run a report.

```
Single Sign-On Summary Report Response File
#Response File generated for sso_summary.rptdesign
#Wed Nov 28 00:14:52 CST 2007
Date_Range_Start=Enter a value here
Event_Type=Enter a value here
Date_Range_End=Enter a value here
User=All User IDs
```

Below is an edited response file for the Single Sign-On Summary Report. Ensure that you enter the correct values that apply to your Common Audit Service server.

```
Example Single Sign-On Summary Report Response File
#Response File generated for sso_summary.rptdesign
#Wed Nov 28 00:14:52 CST 2007
```

```
Date_Range_Start=01/01/2007 12:00:00 AM
Event_Type=All Events
Date_Range_End=01/31/2007 12:00:00 AM
User=yogi
```

Refer to the *Tivoli Federated Identity Manager Administration Guide* for more information on using response files with the report commands.

Running the out-of-box Federated Identity Manager reports using the console

This topic describes how to use the console to run the reports that are supplied with Federated Identity Manager.

Use the following procedure to generate an out-of-box Federated Identity Manager report (either the Administrative Events Report or the Single Sign-On Summary Report).

1. First perform the setup tasks described in "Required reporting setup tasks" on page 150.
2. In the Integrated Solutions Console, select **Tivoli Federated Identity Manager**-> **Domain Management**-> **Reports**.
3. Click the **Configure Report Settings** tab.
4. Specify values for all of the database fields. All fields are required for Federated Identity Manager to connect to the XML data store used by the Common Audit Service.
5. Click **Apply** ->**Load changes to Tivoli Federated Identity Manager runtime**.
6. Click the **Run Reports** tab. The names of the report design files for the two out-of-box reports and any custom reports that can be run are displayed.
7. Select a report and specify the report parameters. Click the help icon (?) for information on each field.
8. Click **Run** to generate a report.
9. To view the status of reports that are currently running, select the **View Active Reports** tab. A listing of reports that are currently running is displayed.

   To view a list of reports that have completed execution (archived reports), select the **Archived Reports** tab. In this view you can download or delete an archived report.

   To download an archived report, select a report, click **Download**, specify the target location, and click **OK**. After the report is downloaded, you can view the report using a PDF reader or browser, depending on the filetype of the report.

   To delete an archived report, select the report and click **Delete**. The selected report is deleted from the Federated Identity Manager runtime system path, such as, /opt/IBM/FIM/reports/archives.

Importing the Federated Identity Manager out-of-box reports into the Tivoli Common Reporting environment

This topic describes how to run the reports that are supplied with Federated Identity Manager using the Tivoli Common Reporting Console.

In the following steps, the *Tivoli Common Reporting User's Guide* is the tcr_users_guide.pdf file that is located on the reporting installation media that is included with Tivoli Federated Identity Manager.

Use the following procedure to generate an out-of-box Federated Identity Manager report (either the Administrative Events Report or the Single Sign-On Summary Report).

1. First perform the setup tasks described in “Required reporting setup tasks” on page 150.
2. Install Tivoli Common Reporting. You can install Tivoli Common Reporting interactively, using a graphical interface or commands, or noninteractively, in silent mode. Refer to the *Tivoli Common Reporting User's Guide* for complete information on installing Tivoli Common Reporting.
3. Start the Tivoli Common Reporting Server. Refer to ″Starting and stopping the Tivoli Common Reporting server″ in the *Tivoli Common Reporting User's Guide* for information on starting Tivoli Common Reporting.
4. Log into Tivoli Common Reporting using the console. Refer to ″Administering Tivoli Common Reporting″ in the *Tivoli Common Reporting User's Guide* for information on logging into Tivoli Common Reporting.
5. Import the Federated Identity Manager report package into the Tivoli Common Reporting environment. The report package is labeled TFIM6.2_Reports_V*n_yymmdd*.zip, where *n* is the version number and *yymmdd* is the date (for example, TFIM6.2_Reports_V1.0_080527.zip). To import the package, right click on the root of the navigation tree in the Tivoli Common Reporting console and click **Import Report Package**. The Import Report Package dialog is displayed.

   The report package is located on the system where Federated Identity Manager is installed, under *FIM_INSTALL_ROOT*/reports/tcr-pkg.
6. Browse to select the report package zip file in the **Local File** field.
7. Click **Import**. When the file is imported, the navigation tree displays **Tivoli Federated Identity Manager 6.2 Reports** in the navigation tree.
8. Click **Tivoli Federated Identity Manager 6.2 Reports**. The report titles and descriptions are displayed in the right pane.

   You can search reports by name, keywords, and descriptions. For example, if you enter ″administrative″ in the Search box in the top right corner, all reports that contain the word ″administrative″ in the name are displayed. To use advanced search options, click the Search tab. For Federated Identity Manager audit events, enter keywords such as identity, security, and events to filter report data.
9. Configure the Tivoli Common Reporting data source to provide the event data for the reports. The data source must be changed to point to the Common Audit Service XML data store:
   a. Place the db2jcc.jar and db2jcc_license_cu.jar JDBC drivers needed to establish a connection to the database in the following directory. The drivers are located on the Federated Identity Manager runtime system in /opt/IBM/FIM/reports/engine/plugins/org.eclipse.birt.report.data.oda.jdbc_2.2.1.r22x_v20070919drivers.

      tcr_install\lib\birt-runtime-2_2_1\ReportEngine\plugins\org.eclipse.birt.report.data.oda.jdbc_2.2.1.r22x_v20070919\drivers
   b. Right click on a report in the Reports pane and click **Data Sources**. This action displays a list of data source names and types used by the report set. Select **Database Data Source** and click **Edit**. The **Edit Data Source** dialog is displayed.
   c. Enter the JDBC driver, JDBC URL, user ID and password that is used to connect to the XML data store that is managed by the Common Audit Service server. The JDBC driver value is set by default to

com.ibm.db2.jcc.DB2Driver. Alternatively, you can use a JNDI name to establish a connection. You can leave the remaining fields empty.

Note: Changing the data source information applies to all reports, not just the selected report.

Examples of JDBC drivers, JDBC URLs and JAR file names for the different types of databases are as follows:

- Database: DB2
- JDBC Driver: com.ibm.db2.jcc.DB2Driver
- JDBC URL: jdbc://db2://example.com:60000/eventxml
- JAR File: db2jcc.jar, db2jcc_license_cu.jar

d. Click **Save** to save your changes.

After importing the reports and setting up the database connection, you are ready to run a report. Refer to "Running the out-of-box Federated Identity Manager reports using the Tivoli Common Reporting console"

Running the out-of-box Federated Identity Manager reports using the Tivoli Common Reporting console

This topic describes how to run the reports that are supplied with Federated Identity Manager using the Tivoli Common Reporting Console.

In the following steps, the *Tivoli Common Reporting User's Guide* is the tcr_users_guide.pdf file that is located on the reporting installation media that is included with Tivoli Federated Identity Manager.

Use the following procedure to generate an out-of-box Federated Identity Manager report (either the Administrative Events Report or the Single Sign-On Summary Report).

1. First, perform the setup tasks described in "Required reporting setup tasks" on page 150.
2. Next, import the Federated Identity Manager report package into Tivoli Common Reporting using the console (see "Importing the Federated Identity Manager out-of-box reports into the Tivoli Common Reporting environment" on page 156).
3. You can run an on-demand report or create a snapshot that is viewed later at any time. To run an on-demand report:
   a. Click the HTML or PDF button in the **Format** column for that report. A parameter dialog is displayed.
   b. Select the starting and ending dates for the report. If you want to limit the types of events that are included in the report, click the expansion arrow for the **Event_Type** field and specify the event types you want to view. Refer to "Federated Identity Manager reports" on page 160 for information on the report parameters and associated event types.
   c. Click **Run** to start the report.

   To create a report snapshot:
   a. Right click on a report and select **Create Snapshot**. A parameter dialog is displayed.
   b. Specify the parameter values and click **Create**. A dialog is displayed showing the status of the snapshot creation, start and completion timestamps, and user.

c. You can right click on a completed snapshot and select **View as..** to format the snapshot in HTML, PDF, Excel, or Postscript.

Creating reports from existing designs

This topic describes how to create new reports using out-of_box Federated Identity Manager report designs in the Tivoli Common Reporting environment.

By specifying default parameters to a report design you can create new reports from existing designs. For example, you can create a report set called Monthly under which you could store separately the monthly Administrative Events reports.

To create a new report from an existing design of a Federated Identity Manager report, specify the following parameters:

- Date_Range_Start
- Date_Range_End
- Event_Type
- Administrator

You can run this report every month on a specific date without having to enter the parameters again, for example:

```
"Date_Range_Start=08/01/07 12:00AM" "Date_Range_End=08/31/07 12:00AM"
```

1. From the command prompt, navigate to *TCR INSTALL*/bin
2. Find the report design you want to use:

   ```
   trcmd -list -designs -user %username% -password %pw%
   ```

3. Define a new report using the trcmd -defineReport command, for example:

   ```
   trcmd -defineReport -report /Monthly_FIM_Reports/
   Administrative_Events_MONTHLY -designRef /FIM_Reports/
   administrative_events.rptdesign -displayName "Monthly Administrative
   Events Report" -description "Monthly report for audited administrative
   events" -parameters "Date_Range_Start=01/01/08 12:00 AM"
   "Date_Range_End=01/31/08 12:00 AM" "Event_Type=All Events"
   "Administrator=All User IDs" -keywords administrative events security
   monthly -user %username% -password %pw%
   ```

4. Optionally, you can globalize the report name and description. Add the report name and description key value pairs to the FIMREPORTS_global properties file, for example:

   ```
   trcmd -modify -report /Monthly_Reports/Administrative_Events_MONTHLY
   -displayNameKey FIM_title_administrative_events_monthly -descriptionKey
   FIM_description_administrative_events_monthly -defaultName "Monthly
   Administrative Events Report" -globalizedFile FIMReports_global -user
   %username% -password %pw%
   ```

5. You can add the report to an existing report set, for example:

   ```
   trcmd -modify -reportSet /FIM_Reports -reports -include
   /My_FIM_Reports/Administrative_Events_MONTHLY - user %username%
   -password %pw%
   ```

6. You can add the report to the new report set, for example:

   ```
   trcmd -add -reportSet /Monthly_FIM_Reports -reports -include
   /My_FIM_Reports/Administrative_Events_MONTHLY -user %username% -password
   %pw%
   ```

Federated Identity Manager reports

This topic describes the reports that are provided by Federated Identity Manager.

Table 26. Administrative Events report

Administrative Events report	
Description	This report shows various configuration changes that have occurred in the Federated Identity Management environment during a specified time frame. You can view administrative activities such as the creation, modification, and deletion of federations, federation partners, and Web service partners. In addition, auditing changes, such as the enabling or disabling of auditing and where audit logs are sent, are available in this report.
Parameters	**Date_Range_Start** Specify a start date to use when finding events to include in a report. **Date_Range_End** Specify an end date to use when finding events to include in a report. **Event_Type** Specify the type of events to include in a report. The values can be: All Events, Successful Single Sign-on Events, or Failed Single Sign-on Events. The default is All Events. **Administrator** Specify the Federated Identity Manager administrator user IDs to include in a report. You can select All User IDs to include all administrator user IDs, or you can choose a specific user ID. The default is All User IDs.
Audit events used	IBM_SECURITY_MGMT_POLICY, IBM_SECURITY_MGMT_AUDIT

Table 27. Single Sign-On Summary report

Single Sign-On Summary report	
Description	This report shows summary data for single sign-on activities that have occurred in the Federated Identity Management environment. You can view both successful and failed single sign-on events during a given time frame.
Parameters	**Date_Range_Start** Specify a start date to use when finding events to include in a report. **Date_Range_End** Specify an end date to use when finding events to include in a report. **Event_Type** Specify the type of events to include in a report. The values can be: All Events, Successful Single Sign-on Events, or Failed Single Sign-on Events. The default is All Events. **User** Specify the Tivoli Federated Identity Manager user ID to include in a report.
Audit events used	IBM_SECURITY_AUTHN

Creating custom reports

This topic describes how to use the Common Audit Service and the DDL files supplied by Federated Identity Manager to create 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 will provide the desired information. Next, examine the contents of the DDL files and the tables that are generated by the DDL files to determine if the captured event types and attributes are sufficient to meet your reporting needs. You then run the Common Audit 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.

The Common Audit Service staging utility (staging utility) queries a configuration file, CARSShredder.conf, to determine exactly what data to stage. If you need to stage additional data that is not provided in the predefined staging tables, then you have to customize the CARSShredder.conf and create new reporting tables. Custom reports can then be created using the event data that is staged into the reporting tables. Any reporting tool that queries data from a DB2 database can be used to create custom reports; however, it is recommended that you use Common Audit Service and BIRT to design, run, view, and manage custom reports.

To set up the CARSShredder.conf configuration file, back up and then replace the default version of the file with a new, custom version that you build using the CARSShredder.conf.custom.template. You do not modify existing default tables to create custom tables; you create new, additional 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"
- "Steps to support custom reports" on page 162
- "Working with the CARSShredder.conf configuration file" on page 163
- "Sample custom report" on page 168
- "Federated Identity Manager DDL contents" on page 171

Requirements for creating new reporting tables

This topic describes the requirements for creating secondary database tables for custom reports by modifying a copy of the Common Audit Service CARSShredder.conf file and the DDL files. These files are installed when you install the Common Audit Service component. Federated Identity Manager also provides a CARSShredder.conf file and corresponding DDL files that you can back up and use to generate secondary custom report tables specifically for Federated Identity Manager security events. *To create custom tables for Federated Identity Manager security events, it is recommended that you use copies of the files provided with Federated Identity Manager.* Refer to "Required reporting setup tasks" on page 150 for information on the location of the Federated Identity Manager CARSShredder.conf file and corresponding DDL files. Refer to "Federated Identity Manager DDL contents" on page 171 for information on the Federated Identity Manager DDL files.

The audit server installation creates a set of default reporting tables, which an exploiting application, such as Federated Identity 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 will prune 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. This following sections describe these data relationships in detail.

Steps to support custom reports

You must perform the following tasks to support custom reports regardless of the reporting tool you are using to run and view the report data:

1. Create a data definition language (DDL) file that creates custom secondary reporting tables in the XML event store database. Federated Identity Manager provides the DDL files required to set up reporting tables for Federated Identity Manager events. Refer to "Federated Identity Manager DDL contents" on page 171 for information on these DDL files.
2. Run the DDL file to create the reporting tables using the following commands:
   a. `db2 connect to database_name user db2username using db2password`
   b. `db2 -tsf custom.ddl`
   c. `db2 disconnect database_name`
3. Save a copy of the default CARSShredder.conf file that is shipped with Common Audit Service as CARSShredder.conf.default so that you can restore it if needed. You will replace this file with your own version to generate custom reports. The default file is located in the *CARS_HOME*/server/etc directory.
4. Create a copy of the CARSShredder.conf.custom.template file, located in *CARS_HOME*/server/template. 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 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 using SQL language.
c. Generate a custom report to verify that the desired data is included.

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 Audit 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.

CARSShredder.conf 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 Audit 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 that will be staged 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 below in **Event section descriptors**) 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 28 on page 165).

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 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 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/xpaths directory. A default set of keyword-XPath properties files are installed in the *CARS_HOME*/server//template/xpaths directory.

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 28. 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//template/xpaths 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 that will be placed in a column for all events. This could 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, and so on.
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/xpaths/*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/xpaths directory. If the properties file cannot be located or opened, or if the keyword cannot be located in the properties file, the staging utility returns an error and stops processing. In the case of 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 28. 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]#`. Note that in Xpath expressions the first element of an array starts with an index value of 1 instead of 0. If the number of array indices 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 28. 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's globalInstanceId and populate the EVENT_ID column with it.
cars_t_event, cars_seq_number, #RECORD_ID
Maps the event's record_id field to the cars_seq_number column.
cars_t_event, time_stamp, #creationTime#
Maps the event's 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 specifying 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, just 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#. Note that the ending number sign helps differentiates 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 will result 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 Audit 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. It is highly recommended that you use a copy of this file as a starting point, and create additional 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 Audit 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 will attempt to convert from the source format to the target format.

Sample custom report

The purpose of this example custom report is to list resources that were accessed and identify whether access was permitted or denied and, in the case of the denial report, the reason for denial.

Following are the steps required to create this custom report:

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)

     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 CARSShredder.conf.default.
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_plcy, #resourceInfo.nameInPlcy#
   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_attr, usr_attr_name, CommonBaseEvent/
      extendedDataElements[@name='userInfo']/
      children[@name='attributes']/children[@name='name']/
      values[contains(.,'attrname')]
   cars_t_cauthz_attr, usr_attr_value, CommonBaseEvent/
      extendedDataElements [@name='attributes']/
      children[@name='name']/values[contains(.,'attrname')]/
      ../../children[@name='value']/values
   ```

   In the above example, the last two entries show the Xpath expression instead of keywords. 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. Refer to “Running the staging utility command” on page 136.
7. Use the reporting tool in your environment, for example, Microsoft Excel, to view the data from the custom tables.

Customizing reports for Tivoli Federated Identity Manager

Tivoli Federated Identity Manager does not generate its own reports. However, you can create customized reports to view the content of all the security events that are audited by Tivoli Federated Identity Manager.

One of the key functions of Tivoli Federated Identity Manager is identity mapping. Tivoli Federated Identity Manager provides a Trust Server component in which an incoming token can be converted to another outgoing token. For example, an

incoming UserNameToken can be converted to a SAML 1.0 token. The user principal and key attributes associated with a user are changed in these mapping operations. The audit report data displays the before and after identities from these identity mapping operations. Another important function of Tivoli Federated Identity Manager is the authorization of the requested Web service call. The audit report data for authorization displays whether a particular user has access to a Web service operation.

Sample Tivoli Federated Identity Manager custom report

This topic describes in a sample task the steps you must perform to support custom reports regardless of the reporting tool.

You must have Common Audit Service and Tivoli Federated Identity Manager installed.

1. Create a Data Definition Language (DDL) file named custom.ddl that creates custom secondary report tables in the XML event store database. The custom.ddl defines a new table named custom_t_trust. A subset of attributes from the IBM_SECURITY_TRUST events to monitor identity mapping is staged into the columns of the custom_t_trust table. The custom.ddl file contains the following entries:

```
-------- Event-specific tables -----------

create table custom_t_trust
(
   event_id                  VARCHAR(64) not NULL,
   cars_seq_number            BIGINT,
   appliesTo                  VARCHAR(1024),
   issuer                     VARCHAR(1024),
   token                      VARCHAR(1024),
   moduleName                 VARCHAR(1024),
   action                     VARCHAR(1024),
   ruleName                   VARCHAR(1024),
   tokenInfo                  VARCHAR(1024),
   accessDecision             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 report tables using the following command:
   a. db2 connect to *database_name* user *db2username* using *db2password*.
   b. db2 -tsf *custom.ddl*
3. Save the CARSShredder.conf file that is included with Common Audit Service. You might want to save a copy of the default CARSShredder.conf file as CARSShredder.conf.default so that you can restore it, if needed. It is located in *CARS_HOME*/server/etc. You are replacing this file with your own version to generate custom reports.
4. Copy CARSShredder.conf.custom.template located in CARS_HOME/server/template as CARSShredder.conf and place it in *CARS_HOME*/server/etc/ directory.
5. Modify the CARSShredder.conf file to stage any additional event attributes necessary for your custom reports. Add the following entries to the [security_trust] section.

```
custom_t_trust, event_id,         #GLOBAL_ID
custom_t_trust, cars_seq_number,  #RECORD_ID
custom_t_trust, appliesTo,        CommonBaseEvent/extendedDataElements
```

```
                                    [@name='appliesTo']/values
custom_t_trust, issuer,             CommonBaseEvent/extendedDataElements
                                    [@name='issuer']/values
custom_t_trust, token,              CommonBaseEvent/extendedDataElements
                                    [@name='token']/values
custom_t_trust, moduleName,         CommonBaseEvent/extendedDataElements
                                    [@name='moduleName']/values
custom_t_trust, action,             CommonBaseEvent/extendedDataElements
                                    [@name='action']/values
custom_t_trust, ruleName,           CommonBaseEvent/extendedDataElements
                                    [@name='rule']/values
custom_t_trust, tokenInfo,          CommonBaseEvent/extendedDataElements
                                    [@name='tokenInfo']/values
custom_t_trust, accessDecision,     CommonBaseEvent/extendedDataElements
                                    [@name='accessDecision']/values
```

6. Run the staging utility to stage data into the report tables as documented in Chapter 10, "Running the server utilities," on page 135.

   ```
   java com.ibm.cars.staging.Staging -dbpassword db2password  -mode incremental
   ```

   The staging utility uses the modified CARSShredder.conf file.

7. To generate customs reports run the following SQL commands:
   - To view the attributes of an IBM_SECURITY_TRUST event when the trust server does identity mapping:

     ```
     select  t1.cars_seq_number,
            time_stamp,
            appliesTo,
            issuer,
            token,
            moduleName,
            action,
            ruleName,
            tokenInfo
     from cars_t_event t1, custom_t_trust t2
     where t1.cars_seq_number=t2.cars_seq_number
           and t2.action='Map';
     ```

   - To view the attributes of IBM_SECURITY_TRUST event when the trust server does an authorize call:

     ```
     select t1.cars_seq_number,
            time_stamp,
            appliesTo,
            issuer,
            moduleName,
            action,
            accessDecision,
            tokenInfo
     from cars_t_event t1, custom_t_trust t2
     where t1.cars_seq_number=t2.cars_seq_number
           and t2.action='Authorize';
     ```

Federated Identity Manager DDL contents

This topic describes the contents of the data definition language (DDL) files supplied by Federated Identity Manger.

The following table describes the DDL files that are used to create the reporting tables for each group of events that can be generated by Federated Identity Manager.

Note: To create custom report tables, you can use copies of these files. Do not modify the contents of the following original DDL files because they are required to enable the out-of-box reports to run correctly.

- cars_t_mgmt_audit_custom.ddL
- cars_t_mgmt_policy_custom1.ddL
- cars_t_mgmt_policy_custom2.ddl
- cars_t_authn_custom.ddl

Table 29. Federated Identity Manager DDL files

File name	Database table created	Creates tables for
cars_t_mgmt_ audit_custom.ddl	cars_t_mgmt_ auditc	Creates reporting tables for IBM_SECURITY_MGMT_AUDIT events. The out-of-box Administrative Events Report uses this table.
cars_t_mgmt_ policy_custom1.ddl	cars_t_mgmt_ plcyc1	Creates reporting tables for IBM_SECURITY_MGMT_AUDIT events. The out-of-box Administrative Events Report uses this table.
cars_t_mgmt_ policy_custom2.ddl	cars_t_mgmt_ plcyc2	Creates reporting tables for IBM_SECURITY_MGMT_AUDIT events. The out-of-box Administrative Events Report uses this table.
cars_t_authn_ custom.ddl	cars_t_authnc	Creates reporting tables for IBM_SECURITY_AUTHN events. The out-of-box Single Sign-On Summary Report uses this table.
cars_t_authn_ term_custom.ddl	cars_t_authn_ terminatec	Creates reporting tables for IBM_SECURITY_AUTHN_TERMINATE events. Custom reports can be created to use this table.
cars_t_encrypt_ custom.ddl	cars_t_encryptc	Creates reporting tables for IBM_SECURITY_ENCRYPTION events. Custom reports can be created to use this table.
cars_t_federation_ custom.ddl	cars_t_ federationc	Creates reporting tables for IBM_SECURITY_FEDERATION events. Custom reports can be created to use this table.
cars_t_signing_ custom.ddl	cars_t_signingc	Creates reporting tables for IBM_SECURITY_SIGNING events. Custom reports can be created to use this table.
cars_t_trust_ custom.ddl	cars_t_trustc	Creates reporting tables for IBM_SECURITY_TRUST events. Custom reports can be created to use this table.

Creating a custom security event details report

This topic describes how to create a custom security event details report.

The Common Audit Service provides Java stored procedures that enable you to view the 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 returning data for a single specified security event as one XML document.

The Java stored procedure accepts the security event's record ID as a parameter, and then searches the active, inactive, and restore sets of tables. With the reporting tool of your choice, after establishing a connection with the database, use the following SQL commands to access the security event data.

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.

Generating operational reports from archived data

This topic describes the general procedure to create operational reports from archived data.

Run the store utility using 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 have not already been staged will be staged to the reporting tables.

As with other staged data, generate reports either on demand or using scheduling.

- For information on archiving data, refer to "Archiving audit data" on page 177.
- For information on restoring archived data, refer to "Restoring audit data" on page 178.
- For information on running the staging utility, refer to "Running the staging utility command" on page 136.

Examples of XML-formatted security event data

This topic provides examples of XML-formatted data that is produced from the staging tables.

The following examples show two security events generated by a Trust request. The first event is a validation action on the received token and the second event is a mapping action that is generated when the token is mapped to the STSUniversalUser. Note that to fit the information on the page, line continuations on the next line are marked by three periods (...). These are not part of the syntax

```
<CommonBaseEvent creationTime="2007-01-31T20:59:57.625Z"
 extensionName="IBM_SECURITY_TRUST"
 globalInstanceId="CE4454A122E10AB044A1DBB16E020E1D80"
 sequenceNumber="1"
 version="1.0.1">
 <contextDataElements
  name="Security Event Factory"
  type="eventTrailId">
  <contextId>FIM_79f4e4c801101db5aba48cd8e0212be7+656317861</contextId>
 </contextDataElements>
 <extendedDataElements name="token" type="string">
  <values>
   <saml:Assertion
    xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:wss="http://docs.oasis-open.org/wss/2004/01/ ...
       oasis-200401-wss-wssecurity-secext-1.0.xsd"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    AssertionID="Assertion-uuid79b3e60c-0110-1536-9aa6-9a781eece361"
    IssueInstant="2007-01-31T19:48:57Z"
    Issuer="http://issuer/saml11-2"
    MajorVersion="1"
    MinorVersion="1">
   <saml:Conditions
    NotBefore="2007-01-31T19:38:57Z"
    NotOnOrAfter="2007-02-01T12:28:57Z">
    <saml:AudienceRestrictionCondition>
     <saml:Audience>http://appliesto/saml11-2</saml:Audience>
    </saml:AudienceRestrictionCondition>
   </saml:Conditions>
   <saml:AuthenticationStatement
    AuthenticationInstant="2007-01-31T19:48:57Z"
    AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password">
   <saml:Subject>
  </values>
 </extendedDataElements>
 <extendedDataElements name="issuer" type="string">
  <values>http://issuer/saml11</values>
 </extendedDataElements>
 <extendedDataElements name="moduleName" type="string">
  <values>com.tivoli.am.fim.trustserver.sts.modules.SAMLTokenSTSModuleBase</values>
 </extendedDataElements>
 <extendedDataElements name="ruleName" type="string">
  <values>Not Available</values>
 </extendedDataElements>
 <extendedDataElements name="tokenInfo" type="string">
 <values>
  <?xml version="1.0" encoding="UTF-8"?>
  <stsuuser:STSUniversalUser xmlns:stsuuser="urn:ibm:names:ITFIM:1.0:stsuuser">
  <stsuuser:Principal>
   <stsuuser:Attribute name="NameQualifier" type="urn:oasis:names:tc: ...
      SAML:1.0:assertion"/>
    <stsuuser:Attribute name="name" type="urn:oasis:names:tc:SAML:1.1: ...
       nameid-format:emailAddress">
    <stsuuser:Value>BigDummy</stsuuser:Value>
   </stsuuser:Attribute>
  </stsuuser:Principal>
  <stsuuser:AttributeList>
   <stsuuser:Attribute name="ssn" type="http://example.com/federation/v1/namevalue">
    <stsuuser:Value>
     <stsuuser:Value xmlns:stsuuser="urn:ibm:names:ITFIM:1.0:stsuuser"> ...
        555-55-555</stsuuser:Value>
```

```
    </stsuuser:Value>
   </stsuuser:Attribute>
   <stsuuser:Attribute name="MinorVersion" type="urn:oasis:names:tc:SAML:1.0:assertion">
    <stsuuser:Value>1</stsuuser:Value>
   </stsuuser:Attribute>
   <stsuuser:Attribute name="email" type="http://example.com/federation/v1/email">
    <stsuuser:Value>elain@hotmail.com</stsuuser:Value>
   </stsuuser:Attribute>
   <stsuuser:Attribute </values>
 </extendedDataElements>
 <extendedDataElements name="appliesTo" type="string">
  <values>http://appliesto/saml11</values>
 </extendedDataElements>
 <extendedDataElements name="action" type="string">
  <values>Validate</values>
 </extendedDataElements>
 <extendedDataElements="">
  <values>Not Available</values>
 <extendedDataElements>
 </extendedDataElements>
 <extendedDataElements name="outcome" type="noValue">
  <children="">
   <values>SUCCESSFUL</values>
  <children></children>
  <children name="majorStatus" type="int">
   <values>0</values>
  </children>
 </extendedDataElements>
 <sourceComponentId
  application="ITFIM#6.2"
  component="IBM Tivoli Federated Identity Manager"
  componentIdType="ProductName"
  executionEnvironment="Windows XP[x86]#5.1 build 2600 Service Pack 2"
  location="person.location.company.com"
  locationType="FQHostname"
  subComponent="com.tivoli.am.fim.trustserver.sts.modules.SAMLTokenSTSModuleBase"
  threadId="WebContainer : 0"
  componentType="http://www.ibm.com/namespaces/autonomic/Tivoli_componentTypes"/>
 <situation categoryName="ReportSituation">
  <situationType
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:type="ReportSituation"
   reasoningScope="INTERNAL"
   reportCategory="SECURITY"/>
 </situation>
</CommonBaseEvent>

<CommonBaseEvent
 creationTime="2007-01-31T20:59:57.765Z"
 extensionName="IBM_SECURITY_TRUST"
 globalInstanceId="CE4454A122E10AB044A1DBB16E02213050"
 sequenceNumber="2" version="1.0.1">
 <contextDataElements name="Security Event Factory" type="eventTrailId">
  <contextId>FIM_79f4e4c801101db5aba48cd8e0212be7+656317861</contextId>
 </contextDataElements>
 <extendedDataElements
  name="token"
  type="string">
  <values>BigDummy [ Attribute 1 NameQualifier urn:oasis:names:tc:SAML: ...
     1.0:assertion ]
                 [ Attribute 2 name urn:oasis:names:tc:SAML:1.1: ...
     nameid-format:emailAddress
                 [ value 1 BigDummy ] ]
  </values>
 </extendedDataElements>
 <extendedDataElements name="issuer" type="string">
  <values>http://issuer/saml11</values>
 </extendedDataElements>
 <extendedDataElements name="moduleName" type="string">
  <values>com.tivoli.am.fim.trustserver.sts.modules.STSMapDefault</values>
 </extendedDataElements>
 <extendedDataElements name="ruleName" type="string">
```

```
 <values>rule1</values>
</extendedDataElements>
<extendedDataElements name="tokenInfo" type="string">
 <values>me_guest [ Attribute 1 NameQualifier urn:oasis:names:tc:SAML: ...
    1.0:assertion ]
                [ Attribute 2 name urn:ibm:names:ITFIM:5.1:accessmanager
                [ value 1 me_guest ] ]
 </values>
</extendedDataElements>
<extendedDataElements name="appliesTo" type="string">
 <values>http://appliesto/saml11</values>
</extendedDataElements>
<extendedDataElements name="action" type="string">
 <values>Map</values>
</extendedDataElements>
<extendedDataElements name="tokenType" type="string">
 <values>Not Available</values>
</extendedDataElements>
<extendedDataElements name="outcome" type="noValue">
 <children name="result" type="string">
  <values>SUCCESSFUL</values>
 </children>
 <children name="majorStatus" type="int">
  <values>0</values>
 </children>
</extendedDataElements>
<sourceComponentId
 application="ITFIM#6.2"
 component="IBM Tivoli Federated Identity Manager"
 componentIdType="ProductName"
 executionEnvironment="Windows XP[x86]#5.1 build 2600 Service Pack 2"
 location="person.location.company.com"
 locationType="FQHostname"
 subComponent="com.tivoli.am.fim.trustserver.sts.modules.STSMapDefault"
 threadId="WebContainer : 0"
 componentType="http://www.ibm.com/namespaces/autonomic/Tivoli_componentTypes"/>
<situation categoryName="ReportSituation">
 <situationType xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:type="ReportSituation"
  reasoningScope="INTERNAL"
  reportCategory="SECURITY"/>
</situation>
</CommonBaseEvent>
```

Chapter 12. Archiving and restoring audit data

This topic describes how to archive and restore your Common Audit 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 additional 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 will need to be archived to mass storage media prior to being purged from the database. In order to archive security event data from the XML data store, use third party database archive tools in conjunction with the Common Audit Service XML data store utilities.

Use the following XML data store utilities immediately prior to 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 you run the prearchive operation after installing 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/CommonAudit/server/etc] java com.ibm.cars.xmlstoreutils.XmlStoreUtils
  -operation prearchive -dbpassword tivoli
  -configurl FILE:////opt/IBM/Tivoli/CommonAudit/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, 2005 12:04:45 AM GMT

 The last date contained in the table set to be archived is Jul 27, 2005 12:07:05 AM GMT

CBAXU0220I XmlStoreUtility has exited successfully.
```

Figure 2. Sample prearchive output

Detailed information about how to use the XML data store utilities is in "Running the XML data store utilities" on page 137.

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 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);
```

Figure 3. Sample schema of XML event tables and XML event restore tables

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;
```

Figure 4. Sample schema of overflow and overflow restore tables

Detailed information about how to use the XML data store utility is in "Running the XML data store utilities" on page 137.

Chapter 13. Problem determination

This topic provides information for troubleshooting the installation of the Common Audit 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 located in the *Error Message Reference*.

Log files

Log files are produced for the following Common Audit Service components:

- ISMP installer of Common Audit Service audit server
- Common Audit Service configuration utility
- Common Audit Service audit server
- Common Audit Service WebService emitter
- Common Audit 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.

Location

Table 30. 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) **Linux or UNIX**: • *CARS_HOME*/serverInstall.log • *CARS_HOME*/server/logs/sharedLibCreation.log (only for console feature)
Server uninstallation	**Windows**: *CARS_HOME*\serverUninstall.log **Linux or UNIX**: *CARS_HOME*/serverUninstall.log
Note: *CARS_HOME* is the installation directory of Common Audit Service.	

Server utilities log files

The Common Audit Service server utilities produce log files.

Most errors in the server utilities are handled by generating exceptions. When an error occurs, it is logged into 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 139 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 executing, Common Audit Service components create entries in the WebSphere Application Server log files.

The following activities are logged into the WebSphere Application Server logs:

- Installation of Common Audit Service features (Common Audit Service Server and Common Audit Service Configuration Console)
- Configuration of Common Audit Service components using the configuration utility
- Common Audit Service WebService emitter
- Security event factory utilities
- Common Audit Service Web service (Web module)
- Common Audit Service XML data store (EJB module)

By default, the WebSphere Application Server logs for a stand-alone server are located at:

- **Linux or UNIX**: /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 located at:

- **Linux or UNIX**: /opt/IBM/WebSphere/AppServer/profiles/Dmgr01/logs/*dmgr*
- **Windows**: C:\Program Files\IBM\WebSphere\AppServer\profiles\Dmgr01\logs*dmgr*

Configuring log and trace settings

From the WebSphere Application Server Administrative Console, configure the log and trace settings:

1. Click **Troubleshooting** → **Logs and Trace**.
2. Select the server you are configuring.
3. Select the type of settings to update:
   - To change the log settings, click **JVM logs** and make updates to the logging parameters in this window.
   - To change the trace settings, click **Diagnostic trace** and make updates to the configuration of the tracing subsystem on this panel. The **Trace Specification**

field allows you to select which components will have tracing enabled. Refer to the WebSphere Application Server information for the format of trace specifications.

- To change the log details level, click **Change Log Details Levels**, then click the **com.ibm.cars.*** package under General Properties. Select **All messages and Traces** to enable full logging and tracing of all log events, or select a specific log detail level to enable tracing of events that belong only to the selected logging level.

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 troubleshooting the source of errors, consider the use of the traceFilePath and traceLevel values. Setting the trace level to the value 3 (traceLevel=3) causes events resulting from error conditions and from all trace points in the code to be written to the log. Output is written to the file specified by the traceFilePath parameter. The output includes the properties 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 filename, errors will result. To ensure uniqueness, it is recommended that errorFilePath and traceFilePath specify the azn-server-name of the server.

Installation problems

This topic describes some installation problems 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 Audit Service installer does not detect correctly the DB2 software version.

Problem

During installation, the ISMP installer of the Common Audit 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 versions allowed are from Version 8.1.7 or Version
9.1 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 Audit 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 FixPak "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 Audit 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 have run the silent installation, the reason might be that the prerequisite products are not installed, or you have not performed the preinstallation checklist items.

Workaround

Ensure the appropriate prerequisites are on the machine being installed prior to performing a silent installation. Refer to “Pre-configuration checklist for all platforms” on page 93 for additional information on set up 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 Audit Service checks if the target WebSphere Application Server is running. If WebSphere Application Server is stopped, you are notified with an error message specifying that a connection could not be 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 located in the WAS_profile/bin directory:

serverStatus.[bat | sh] *server_name*

If the server is stopped, issue the start server command that is located 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 is not found

This topic describes the problem and workaround you can use if JVM version 1.5 or above is not located during the installation of Common Audit Service.

Problem

During the installation of Common Audit Service , if the installer cannot find JVM version 1.5 or above, you are notified with an error message stating that a suitable JVM could not be found.

Workaround

Rerun the installation program and specify the following option:

-is:javahome *path_to_JVM1.5_home*

Example

The following example runs the Common Audit Service installer and specifies the Java home path:

```
install_cars_audit_srv_win32.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 Audit Service.

Problem

During Installation, if a wrong WebSphere Application Server profile path is passed, or if the perquisite 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.

Debug tracing of installation or uninstallation of Common Audit Service

Run a debug trace, using the -Dis.debug flag, during installation and uninstallation to provide more information if there is a problem.

Purpose

Using the -Dis.debug flag causes InstallShield Multiplatform (ISMP) to display a detailed message about the installation process. This might indicate a problem with the InstallShield Multiplatform product itself or with the Common Audit Service. This is a valuable tool in debugging problems you might encounter during silent installation. Start the installer using 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.

Common Audit Service configuration problems

This topic describes workarounds for some Common Audit Service configuration problems you might encounter.

(AIX) Audit Database configuration fails because operating systems SP2 is not applied

This topic describes a potential problem on AIX if the Audit Database configuration fails because operating system SP2 is not applied.

Problem

In the summary window, an unsuccessful Audit Database configuration is indicated. The following message is displayed in the Common Audit Service Console Status window:

```
Audit Database Configuration: Unsuccessful
CFGMB0029E The database named EVENTXML could not be created. Review the log,
E:/IBM/Tivoli/CommonAuditService/server/logs/dbconfig.log,
to determine the cause of the failure.

Check db2Config.log for following log entries:
SQL1042C An unexpected system error occurred.
```

Workaround

If you are using a DB2 9.1 Fix pack 2 to store audit data, verify:

- Database level by running the db2level command.
- Operating system level (correct service pack is installed) by running the following command: oslevel -s

If the operating system is at the correct service pack level, contact your IBM representative

Example

The following sample output is from the oslevel -s command on an AIX 5.3 system with SP2 applied:

```
5300-05-06
```

Text displays incorrectly in some configuration panels

During configuration, text is displayed in reverse order in some windows, or when using a bidirectional locale, such as Hebrew, the text displays incorrectly.

Problem

During the configuration of Common Audit Service, some windows contain characters that are displayed in reverse order. For example, in the Common Audit 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 WebSphere Administrative Console. After setting the encoding to UTF-8, the English text should display correctly left-to-right, and the Hebrew text should appear correctly.

Use the following steps to set the use of UTF-8 character encoding:

1. In the WebSphere Administrative Console, click **Servers-> Application servers** and select the server 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 installing the audit server:

```
An error occurred while installing the ibmcarsdd.jar JAR file.
```

The error message occurs while deploying 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 108.

C client cannot communicate with the Common Audit Service server

The C client cannot communicate with the Common Audit Service audit server when incorrectly configured.

Problem

The C client cannot communicate with the Common Audit 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 will result 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.

  Note that a valid value for **diskCachePath** is a file path that already exists and includes a valid cache file name.

Common Audit Service upgrade problems

Upgrading Common Audit Service from earlier versions to version 6.1 fails for various reasons

The upgrade of a lower-versioned audit database to the version 6.1 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 6.1.

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 trying to use the installation wizard to uninstall the client or 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, run the following command:

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

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

Restart the WebSphere Application Server, then remove the CarsConfigUtility.jar file manually from the *CARS_HOME*/config/lib folder.

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 the procedure in this section to clean up the system.

The uninstaller of the audit server may 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 standalone single server installation of Common Audit Service:
     a. In the target standalone WebSphere Application Server Administrative Console, select **Applications-> Enterprise Applications**.
     b. Select the **CommonAuditServiceConfiguration** application, and click **Uninstall** to uninstall the Common Audit Service Configuration Utility from the target standalone WebSphere Application Server.

- For a Network Deployment setup of Common Audit Service, uninstall the CommonAuditServiceConfiguration application from the target Deployment Manager by executing 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. Execute following command from the wsadmin command line of the target standalone single server or the Deployment Manager:

   ```
   $AdminApp update isclite modulefile { -operation delete  -contenturi CARS6.1.war}
   ```

3. Remove the Extension MBean Provider for the Configuration Utility if this component is not removed during a failed uninstallation:
   - For a standalone single server installation of Common Audit Service:
     a. In the target standalone 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

The uninstaller of Common Audit Service checks whether components are not unconfigured prior to starting the uninstallation. For this situation, a warning message is displayed and prompts you to fully unconfigure the Common Audit Service components before continuing. If you ignore the warning message and continue with the uninstallation, the configuration components might not uninstall,

and you will forfeit the ability to unconfigure Common Audit Service components using 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:

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 applications from the cluster, select the application 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 have been 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 standalone 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 removing 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 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 using 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.

After completing 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 performed.

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 application's log. The audit data might contain sensitive information.

Workaround

Be aware that the Web service emitter log might contain sensitive data. Access to the application's log file should be protected.

Server utility problems

This topic lists some server utility problems you might encounter.

Exception occurs while running the staging utility

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. Refer to 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 occurring while using the staging utility.

Problem

While running the staging utility, a null pointer exception might be thrown. This failure can be due to an error in the configuration file, an error caused by database failures, or network failures. Typically, database failures are caused by the transaction log filling up, free disk space 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 will run to conclusion. If the error occurs again, rerun the staging utility with a batchsize set to 1. This causes the staging utility to print out 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 quotes. The following example is an incorrect entry in CARSShredder.conf because the order of single and double quotes is inconsistent.

```
cars_t_event, eventType,     ' "AUDIT_AUTHN_CREDS_MODIFY' "
```

If the exception is SQLException, the failure might be due to a database error or a staging error. Refer to the staging utility error log to identify the SQL exception. Errors are frequently caused by a CARShredder.conf file referring to a nonexisting table column, or by including multiple references to an existing target column, as shown in the example below:

```
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 configuring the Common Audit Service server on a WebSphere Application Server Network Deployment cluster configuration, when the DB2 server is remote, the ibmcars.properties file may 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 will not be able to connect to the remote database server, and the following errors will 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.

Problem sending events to Common Audit Service server when security is enabled

This topic describes a problem of a WebSphere Application Server error occurring while attempting to send an event to the Common Audit Service audit server.

Problem

You might receive the ArrayIndexOutOfBoundsExeception in the WebSphere Application Server SystemOut.log while attempting to send an event to the Common Audit Service server when WebSphere Application Server Admin Security and Application Security is enabled. The exception in SystemOut.log follows:

```
[11/14/07 18:00:32:014 EST] 00000027 PrivExAction  W   J2CA0144W:
No mappingConfigAlias found for connection factory or datasource jdbc/eventxml.
[11/14/07 18:00:32:031 EST] 00000027 PrivExAction  W   J2CA0114W: No container-managed
authentication alias found for connection factory or datasource jdbc/eventxml.
[11/14/07 18:00:34:083 EST] 00000026 InternalDB2Un I   DSRA8203I:
Database product name : DB2/LINUXZ64
[11/14/07 18:00:34:113 EST] 00000026 InternalDB2Un I   DSRA8204I:
Database product version : SQL09012
[11/14/07 18:00:34:115 EST] 00000026 InternalDB2Un I   DSRA8205I: JDBC driver name  :
IBM DB2 JDBC Universal Driver Architecture
[11/14/07 18:00:34:118 EST] 00000026 InternalDB2Un I   DSRA8206I:
JDBC driver version  : 3.3.54
[11/14/07 18:00:34:121 EST] 00000026 InternalDB2Un I   DSRA8212I: DataStoreHelper
name is: com.ibm.websphere.rsadapter.DB2UniversalDataStoreHelper@59b759b7.
[11/14/07 18:00:34:127 EST] 00000026 WSRdbDataSour I   DSRA8208I: JDBC driver type  : 4
[11/14/07 18:00:35:409 EST] 00000026 ServiceLogger I com.ibm.ws.ffdc.IncidentStreamImpl
   initialize
FFDC0009I: FFDC opened incident stream file
   /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/
logs/ffdc/server1_6fa06fa_07.11.14_18.00.35_0.txt
[11/14/07 18:00:35:453 EST] 00000026 ServiceLogger I com.ibm.ws.ffdc.IncidentStreamImpl
resetIncidentStream FFDC0010I: FFDC closed incident stream file /opt/IBM
/WebSphere/AppServer/profiles/AppSrv01/logs/ffdc/server1_6fa06fa_07.11.14_18.00.35_0.txt
[11/14/07 18:00:35:482 EST] 0000002b ServiceLogger I com.ibm.ws.ffdc.IncidentStreamImpl
   initialize
FFDC0009I: FFDC opened incident stream file
```

```
   /opt/IBM/WebSphere/AppServer/profiles/AppSrv01/
logs/ffdc/server1_2e092e09_07.11.14_18.00.35_0.txt
[11/14/07 18:00:35:461 EST] 00000026 ExceptionUtil E   CNTR0019E: EJB threw an unexpected
(non-declared) exception during invocation of method "createEvents". Exception data:
com.ibm.ejs.container.CreateFailureException: ; nested exception is:
        java.lang.reflect.InvocationTargetException
        at com.ibm.ejs.container.StatelessBean0.<init>(StatelessBean0.java:172)
        at com.ibm.ejs.container.CMStatelessBean0Factory.create(CMStatelessBean0
Factory.java:40)
        at com.ibm.ejs.container.EJSHome.createBean0(EJSHome.java:926)
        at com.ibm.ejs.container.EJSHome.createBean0(EJSHome.java:1029)
        at com.ibm.ejs.container.activator.UncachedActivationStrategy.atActivate
(UncachedActivationStrategy.java:84)
        at com.ibm.ejs.container.activator.Activator.activateBean(Activator.java
:597)
        at com.ibm.ejs.container.EJSContainer.preInvokeActivate(EJSContainer.jav
a:3469)
        at com.ibm.ejs.container.EJSContainer.preInvoke(EJSContainer.java:2874)
        at com.ibm.cars.xmlstore.xmlstoreds.EJSLocalStatelessXmlStore_cdb69812.c
reateEvents(Unknown Source)
        at com.ibm.cars.webservice.CBEHandlerFactory$XMLDSCBEHandler.send(CBEHan
dlerFactory.java:463)
        at com.ibm.cars.webservice.WSEmitterImpl.sendEvent(WSEmitterImpl.java:33
9)

.....................................

Caused by: java.lang.reflect.InvocationTargetException
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.
java:64)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAcces
sorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:615)
        at com.ibm.ejs.container.StatelessBean0.<init>(StatelessBean0.java:165)
        ... 50 more
Caused by: java.lang.ArrayIndexOutOfBoundsException: Array index out of range: 0
        at com.ibm.ws.management.repository.FileRepository.extract(FileRepositor
y.java:960)
        at com.ibm.ws.management.repository.FileRepository.extract(FileRepositor
y.java:949)
        at com.ibm.cars.xmlstore.xmlstoreds.XmlStoreCustomProperties.setXmlStore
CustomProperties(XmlStoreCustomProperties.java:111)
        at com.ibm.cars.xmlstore.xmlstoreds.XmlStoreCustomProperties.<init>(XmlS
toreCustomProperties.java:80)
        at com.ibm.cars.xmlstore.xmlstoreds.XmlStoreEjb.initializeXmlStore(XmlSt
oreEjb.java:287)
        at com.ibm.cars.xmlstore.xmlstoreds.XmlStoreEjb.ejbCreate(XmlStoreEjb.ja
va:109)
```

The WebSphere Application Server Admin Security and Application Security are enabled, but SSL was not enabled for encrypting WebService SOAP messages. The WebService security roles are mapped to ″everyone.″ The client application configuration did not specify the proper username and password for basic client authentication.

Workaround

Map the WebService security roles to ″All authenticated.″ Edit the application configuration file to include the appropriate WebSphere administrative username and password.

Out of memory error

When sending millions of events to the server, you might encounter a WebSphere out-of-memory error. This error might be due to a stack overflow problem. For

information regarding how to resolve the stack overflow problem, see the IBM Redbook *WebSphere Application Server: Application Server Crash Problem Determination*.

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 grant 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:

IBM World Trade Asia Corporation
Licensing 2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, 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 may 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
USA

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 information 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 measurements 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.

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.

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 softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A complete and current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

Adobe, Acrobat, Portable Document Format (PDF), and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

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.

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

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Index

A

accessibility ix
archiving
  audit data 177
  cleanrestore tables operation 139
  postarchive operation 138
  prearchive operation 138
attributes, staging 163
audience for this book vii
audit configuration event 60
audit data
  archiving 177
  restoring 178
audit infrastructure 76
audit provisioning event 63
audit server
  cleaning up failed uninstallation 189
  configuration checklist 93
  configuring 93
  GUI instructions
    stand-alone server 119
    upgrading the server 115
  installation checklist 82
  installation options 86
  installing 79
  installing in clustered environment 98
  installing prerequisites 79
  installing stand-alone 85
  interactive installation 84
  interactive uninstallation 122
  panel instructions
    stand-alone server 94
  reporting tables 161
  running utilities 135
  silent installation 88
  silent uninstallation 123
  uninstallation checklist 121
  uninstalling 121
  upgrading 113

C

cars_t_authz table 161
cars_t_event table 161
CARSShredder.conf file
  custom template 168
  format 163
  location 163
  purpose 163
  requirements 167
changing XML data store utilities configuration 139
checklist
  pre-installation
    audit server on Linux or UNIX 82
  uninstallation
    audit server 121
checklist, pre-configuration 93
CLASSPATH, setting 135
cleaning up
  failed uninstallation 189
cleanrestore tables set operation 139
CLI, report 153
CLI, response files 155
client events, securing 125
cluster type
  heterogeneous 97
  homogeneous 97
clustered environment
  cluster types 97
  configure Web server 98
  configuring IBM HTTP Server 132
collecting audit data 77
commands
  ibmcarsddinst 109
  installing audit server
    interactive mode 84
    silent mode 88
  running staging utility 136
  setting up symbolic links for Linux 108
  staging utility 136
  uninstalling audit server
    interactive mode 122
    silent mode 124
  XML data store utilities 138
Common Audit Service
  archiving data 177
  configuring 93
  description 75
  installing 79
  reports 149
  restoring data 178
  securing 125
  troubleshooting 179
  unconfiguring 119
  uninstalling 123
  upgrading server 113
  utilities 135
Common Base Event 76
common elements 7
compress property, configuring 98
compressed form
  storing events in 98
configuration console
  installing 79
  uninstalling 123
configuration options, event server 95
configuring
  audit server 93
  checklist 93
  compress property 98
  custom registry 127
  federated registry 128
  ibmcars.properties file 139
  JDBC driver 128
  jdk_path parameter 109
  LDAP registry 126
  log and trace settings 180
configuring *(continued)*
  operating system registry 126
  shredder configuration file 163
  stand-alone server 94
  Web server SSL 132
  WebSphere Application Server SSL 131
  XML store utilities parameters 142
console, report 156
ContextDataElements element 7
creating
  custom reports 161
  custom security event details report 172
  operational reports 173
  reporting tables 161
Custom Audit Details report, setting up to run 108
custom registry, configuring 127
custom reports
  creating 161
  sample 168
  supporting 162

D

data definition language files (DDL) 171
data definition language script, customizing 90
DB2 client
  installing on Linux or UNIX 81
  installing on Windows 80
DB2 server, prerequisite for audit server 79
DB2, securing XML data store 134
debug trace 184
deleting events using staging utility 137
deploying Java stored procedure 108
descriptor, event section 163
Dis.debug flag 184

E

element
  ContextDataElements 7
  format 7
  Outcome 10
  Situation 9
  SourceComponentId 8
elements for
  audit configuration 60
  audit provisioning 63
  encryption 23
  IBM_SECURITY_AUTHN 10
  IBM_SECURITY_AUTHN_TERMINATE 13
  IBM_SECURITY_ENCRYPTION 23
  IBM_SECURITY_FEDERATION 14
  IBM_SECURITY_MGMT_PROVISIONING 63

elements for *(continued)*
IBM_SECURITY_MGMT_AUDIT 60
IBM_SECURITY_MGMT_POLICY 25
IBM_SECURITY_SIGNING 24
IBM_SECURITY_TRUST 18
management 25
name identifier management 14
signing 24
single logout 13
single sign-on 10
trust service 18
emitter log files 180
enabling language support 89
encryption event 23
error file path, setting 181
event
format 7
event section descriptor 163
event server
configuration options 95
event stanza 165
event types 7
events
archiving 177
audit configuration 60
audit provisioning 63
encryption 23
format 76
IBM_SECURITY_AUTHN 10
IBM_SECURITY_AUTHN_ TERMINATE 13
IBM_SECURITY_ENCRYPTION 23
IBM_SECURITY_FEDERATION 14
IBM_SECURITY_MGMT_ PROVISIONING 63
IBM_SECURITY_MGMT_AUDIT 60
IBM_SECURITY_MGMT_POLICY 25
IBM_SECURITY_SIGNING 24
IBM_SECURITY_TRUST 18
infrastructure 76
management 25
restoring 178
signing 24
single logout 13
trust service 18

F

failed uninstallation
cleaning up after 189
federated registry, configuring 128
federation events, name identifier management 14
flag, Dis.debug 184
format
elements 7
events 76

G

global security, setting up 125
GUI instructions
installing stand-alone server 85
stand-alone server 119
upgrading the server 115

H

heterogeneous cluster 97
heterogeneous cluster, determining 97
historical mode, staging data in 137
homogeneous cluster 98
homogeneous cluster, determining 97

I

IBM HTTP Server
configuring for SSL 132
IBM Tivoli Federated Identity Manager library vii
IBM_SECURITY_AUTHN event 10
IBM_SECURITY_AUTHN_ TERMINATE event 13
IBM_SECURITY_ENCRYPTION event 23
IBM_SECURITY_FEDERATION event 14
IBM_SECURITY_MGMT_ PROVISIONING event 63
IBM_SECURITY_MGMT_AUDIT event 60
IBM_SECURITY_MGMT_POLICY event 25
IBM_SECURITY_SIGNING event 24
IBM_SECURITY_TRUST event 18
IBMCARS_DD_REPORT Java stored procedure 172
verifying 110
IBMCARS_EVENT_DETAIL Java stored procedure 172
verifying 110
ibmcars.properties file 139
ibmcarsddinst commands 109
incremental mode, staging data in 137
infrastructure, auditing 76
installing
audit server 79
audit server options 86
checklist
audit server on Linux or UNIX 82
clustered environment 98
configuration console 79
DB2 client on Linux or UNIX 81
DB2 client on Windows 80
interactive mode
audit server 84
language support 90
log files 179
prerequisite products 79
running debug trace 184
silent mode 88
stand-alone server 85
interactive mode
GUI instructions
stand-alone server 119
installation
audit server 84
panel instructions
stand-alone server 94
uninstalling audit server 122
upgrading the server 115

J

Java stored procedure
deploying 108
IBMCARS_DD_REPORT 172
IBMCARS_EVENT_DETAIL 172
setting up to run on Linux 108
verifying 110
JDBC driver
configuring 128
JDBC driver, Policy Tool 129
jdk_path parameter, setting 109

K

keywords, shredder configuration file 165

L

language support 89
LDAP registry, configuring 126
log files
installation 179
server utilities 180
WebSphere Application Server 180

M

management 25
mapping
attributes 163
Web service roles 134
modifying XML data store utilities configuration 139
moving data using shredder configuration file 163

N

name identifier management 14

O

operating system registry
configuring 126
operational reports
creating 173
options
audit server installation 86
options, event server configuration 95
Outcome element 10

P

packages, language 89
panel instructions
stand-alone server 94
parameters
configuring staging utility 142
Policy Tool, editing app.policy 129
postarchive operation 138
pre-configuration checklist, audit server 93

pre-installation checklist
audit server on Linux or UNIX 82
prearchive operation 138
prerequisite products, installing for audit server 79
prerequisite tasks
configuring audit server 93
installing audit server on Linux or UNIX 82
uninstalling audit server 121
properties file, ibmcars.properties 139
prune mode, staging data in 137

R

registry
configuring custom 127
configuring federated 128
operating system
configuring 126
registry, LDAP 126
removing
audit server 121
removing events using staging utility 137
report tables
staging data to 136
reporting
tables 161
reports
creating custom 168
custom 161, 162
Custom Audit Details
setting up to run 108
custom security event details 172
DDL files 171
description 160
designs 159
generating tables for 139
overview 149
response files 155
running using CLI 153
running using console 156
running using Tivoli Common Reporting 156, 158
scenario 77
viewing list using CLI 154
restore tables
cleaning 139
restoring
audit data 178
roles
Web Service
mapping 134
running
audit server utilities 135
debug trace 184
ibmcarsddinst commands 109
staging utility 136
XML data store utilities 137

S

samples
custom report 168
ibmcars.properties file 140
samples *(continued)*
installing audit server
interactive mode 85
silent mode 89
prearchive output 177
running ibmcarsddinst command 110
schema of overflow tables 178
schema of XML event tables 178
shredder configuration file section 167
uninstalling audit server
interactive mode 123
silent mode 124
XML data store utilities 139
scenarios
creating custom report 168
Secure Sockets Layer (SSL)
Web server
configuring 132
WebSphere Application Server
configuring 131
securing
XML data store 134
securing client events 125
security policy for JDBC driver 128, 129
security Web service log files 180
security, setting up 125
server
cleaning up failed uninstallation 189
configuration checklist 93
configuration options 95
deploying Java stored procedure 108
GUI instructions
stand-alone 119
upgrading the server 115
installation checklist 82
installation options 86
installing in clustered environment 98
installing stand-alone 85
interactive installation 84
panel instructions
stand-alone 94
reporting tables 161
uninstallation checklist 121
utilities
log files 180
setting CLASSPATH 135
staging 136
XML data store utilities 137
setting
CLASSPATH 135
jdk_path parameter 109
setting up
global security 125
Java stored procedure
Linux 108
shredder configuration file
template 168
signing event 24
signing/encryption events
encryption 23
signing 24
silent mode
audit server installation 88
response file 88
silent mode *(continued)*
uninstallation
audit server 123
single sign-on event 10, 13
Situation element 9
software prerequisites, installing for audit server 79
SourceComponentId element 8
SSL (Secure Sockets Layer)
Web server
configuring 132
WebSphere Application Server
configuring 131
SSL, setting for Web server 132
staging data
custom reports 161
description 76
staging utility
configuration parameters 142
ibmcars.properties file 139
restoring audit data 178
return codes 137
running 136
setting CLASSPATH for 135
shredder configuration file 163
stanza, event 165
stored procedure, Java 108
storing
events 76

T

tables
creating 161
templates, shredder configuration file 168
third party
archiving 177
Tivoli Common Reporting 156, 158
trace level, setting 181
trace path, setting 181
trace settings
configuring 180
tracing, running debug 184
troubleshooting problems 179
trust service 18

U

uncompressed form
storing events in 98
unconfiguring stand-alone server 119
uninstalling
audit server 121, 122
checklist
audit server 121
cleaning up after 189
language support 124
running debug trace 184
silent mode
audit server 123
window instructions
audit server 123
updating XML data store utilities configuration 139

upgrading
audit server 113
GUI instructions
stand-alone server 115
utilities
running audit server 135
server
setting CLASSPATH 135
staging 136
XML data store 137

V

verifying
Java stored procedure 110
virtual host, mapping Common Audit Service to 102

W

Web server
cluster node 98
configuring for clustered environment 98
installing in clustered environment 98, 99
mapping Common Audit Service to 103
node outside of cluster 99
securing remote communication 132
Web server SSL
configuring 132
Web service
mapping roles 134
Web service client events, securing 125
WebSphere Application Server
cluster types 97
clustered environment
preparing to install 98
configuring
custom registry 127
federated registry 128
operating system registry 126
configuring IBM HTTP Server 132
configuring JDBC driver 128
configuring LDAP registry 126
global security 125
mapping Web service roles 134
Policy Tool 129
prerequisite for audit server 79
propagating plug-in configuration 102
SSL
configuring 131
stand-alone configuration 94
stand-alone installation 85
stand-alone unconfiguration 119
upgrading the server 115
wsadmin command 128

X

XML data source 168
XML data store
archiving 177
XML data store *(continued)*
configuring the compress property for 98
description 76
securing 134
staging data from 136
upgrading 113
XML data store utilities
configuring ibmcars.properties file 139
running 137
XML event store
restoring 178
XML store utilities
configuration parameters 142
setting CLASSPATH for 135
XPath statements, shredder configuration file 165

Printed in USA

GC32-2287-02