

IBM® Tivoli® Federated Identity Manager
Version 6.2.2

Auditing Guide



IBM® Tivoli® Federated Identity Manager
Version 6.2.2

Auditing Guide



Note

Before using this information and the product it supports, read the information in “Notices” on page 213.

Edition notice

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

© Copyright IBM Corporation 2006, 2011.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this publication vii

Intended audience	vii
Publications	vii
IBM Tivoli Federated Identity Manager library	vii
Prerequisite publications.	viii
Related publications	viii
Accessing terminology online	viii
Accessing publications online	ix
Ordering publications	ix
Accessibility	ix
Tivoli technical training	ix
Support information	ix
Conventions used in this book	x
Typeface conventions	x
Operating system-dependent variables and paths	x

Chapter 1. Tivoli Federated Identity Manager auditing. 1

Chapter 2. Managing audit settings 3

Enabling or disabling auditing	3
Activating audit client profile.	3
Creating audit client profile	4
Modifying audit events.	5
Modifying audit client profiles	6
Deleting audit client profile	6

Chapter 3. 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
Federated profiles - OAuth 1.0 (IBM_SECURITY_TRUST)	18
Federated profiles - OAuth 2.0 (IBM_SECURITY_TRUST)	21
Trust service modules - chains (IBM_SECURITY_TRUST)	23
Message security - encryption (IBM_SECURITY_ENCRYPTION)	27
Message security - signing (IBM_SECURITY_SIGNING).	28
Runtime configuration management (IBM_SECURITY_MGMT_POLICY)	29
Audit configuration management (IBM_SECURITY_MGMT_AUDIT).	64
Audit Provisioning (IBM_SECURITY_MGMT_PROVISIONING)	66
User Self Care (IBM_SECURITY_SELFCARE)	77

Chapter 4. Overview of the Common Audit Service 83

Common Audit Service infrastructure	84
Scenario for collecting audit data	84

Chapter 5. Installing Common Audit Service 87

Installing prerequisite products.	87
Installing the DB2 client on Windows systems.	88
Installing the DB2 Administration Client on Linux and UNIX systems.	89
Pre-installation checklist for all platforms	91
Interactive installation	92
Starting the installation wizard	92
Interactive installation using the GUI panels	93
Audit server installation options	94
Silent installation	96
Enabling language support	97
Installing language support packages.	98
Customizing the XML store data definition language script	99

Chapter 6. Configuring the audit server 101

Pre-configuration checklist for all platforms	101
Interactive configuration using the GUI panels	102
Common Audit Service configuration options	103
Configuring JDBC resources in a clustered environment.	105
Determining the type of cluster	105
Configuring JDBC resources against a heterogeneous cluster.	106
Configuring JDBC resources against a homogeneous cluster.	106
Configuring the compress property	106
Configuring a Web server for use in a clustered environment.	107
Configuring a Web server that is installed on a cluster node.	107
Configuring a Web server that is installed on a system outside the cluster	108
Enabling the IBM HTTP Server	110
Propagating the plug-in if the IBM HTTP Server is installed on a WebSphere Application Server node host.	111
Propagating the plug-in if the IBM HTTP Server is installed on a remote host	111
Completing the Common Audit Service application to Web server mapping	112
Verifying your configuration settings for Common Audit Service	113
Verifying the configuration settings for the Common Audit Service application	113

Verifying the configuration settings for the Common Audit Service data source	114
Verifying the configuration settings for the Common Audit Service data store	115
Verifying the configuration settings for the Common Audit Service web service component	116
Deploying the Java stored procedure for an audit details report	117
Setting up to run the Java stored procedures on Linux	118
Setting the jdk_path parameter	118
Running ibmcarsddinst to deploy the Java stored procedure	119
Verifying the deployment of the IBM_CARS_DD_REPORT Java stored procedure	120
Verifying the deployment of the IBM_CARS_EVENT_DETAIL Java stored procedure	120

Chapter 7. Upgrading the Common Audit Service audit server from earlier versions 121

Considerations for upgrading the Common Audit Service audit server	121
Upgrade goals	121
Preparing to upgrade the Common Audit Service audit server	122
Procedures for upgrading the Common Audit Service audit server from earlier versions	123
Installing Common Audit Service Version 6.1 when upgrading to use an existing database	123
Configuring Common Audit Service Version 6.1 to use an existing audit database	123
Post-upgrade steps: remove the old script, configure the clients to use the new port, uninstall the old version of the audit server	125

Chapter 8. Unconfiguring Common Audit Service 129

Chapter 9. Uninstalling Common Audit Service 131

Uninstallation checklist for all platforms	131
Interactive uninstallation	132
Starting the uninstallation wizard	132
Interactive uninstallation using the GUI windows	133
Silent uninstallation	133
Uninstalling language support packages	134

Chapter 10. Securing data flow in the operating environment 137

Securing Web service client events	137
Configuring the server	137
Securing the XML data store	148

Chapter 11. Running the server utilities 149

Preparing to run the server utilities	149
---	-----

Running the staging utility command	150
Running the XML data store utilities	151
The ibmcars.properties file	153
Configuration parameters for the staging utility and XML data store utilities	156

Chapter 12. Tivoli Federated Identity Manager reports 163

Required reporting setup tasks	164
Running Tivoli Identity Manager reports by using the command-line interface	167
Listing available reports by using the command-line interface	168
Sample response files for running a report using the command-line interface	169
Running the out-of-box Tivoli Federated Identity Manager reports by using the console	170
Importing the Tivoli Federated Identity Manager out-of-box reports into the Tivoli Common Reporting environment	171
Running the out-of-box Tivoli Federated Identity Manager reports by using the Tivoli Common Reporting console	172
Creating reports from existing designs	173
Tivoli Federated Identity Manager reports	174
Creating custom reports	175
Requirements for creating new reporting tables	176
Steps to support custom reports	177
Working with the CARSShredder.conf configuration file	178
Creating sample custom reports	184
Customizing reports for Tivoli Federated Identity Manager	185
Sample Tivoli Federated Identity Manager custom report	186
Federated Identity Manager DDL contents	188
Creating a custom security event details report	188
Generating operational reports from archived data	189
Examples of XML-formatted security event data	190

Chapter 13. Archiving and restoring audit data 193

Archiving audit data	193
Restoring audit data	194

Chapter 14. Problem determination 195

Log files	195
Installation log files	195
Server utilities log files	196
WebSphere Application Server log files	196
Configuring log and trace settings	197
Considerations for setting the trace file path, trace level, and error file path during problem determination	197
Installation problems	197
Installer displays an error although the required DB2 software is installed	198
Silent installation does not fail when missing prerequisites	198

Installation does not continue when the target WebSphere Application Server is stopped . . .	199	CarsConfigUtil.jar is not removed during a successful uninstallation of Common Audit Service	204
Installation does not continue when JVM version 1.5 is not found	199	Failed uninstallation workarounds	204
Installation displays an error when WebSphere Application Server software is not found . . .	200	Web service and emitter problems	208
Debug tracing of installation or uninstallation of Common Audit Service	200	Disregard message 0000004a	208
Common Audit Service configuration problems	201	Web service emitter log displays event data . .	208
(AIX) Audit Database configuration fails because operating systems SP2 is not applied .	201	Server utility problems	209
Text displays incorrectly in some configuration panels	201	Exception occurs while running the staging utility	209
SOAP connection fails when a Common Audit Service Configuration Console is deployed in an eWAS environment	202	java.lang.NullPointer exception occurs while running the staging utility	209
Problem deploying the Java stored procedure on a Linux platform	202	Remote database access failure occurs when using staging utility or XML data store utilities .	210
C client cannot communicate with the Common Audit Service server	203	WebSphere Application Server problems	211
Common Audit Service upgrade problems . . .	203	Problem sending events to Common Audit Service server when security is enabled. . . .	211
Common Audit Service uninstallation problems	204	Out of memory error	212
Uninstall.bin not available	204	Notices	213
		Index	217

About this publication

IBM® Tivoli® Federated Identity Manager Version 6.2.2 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 audit IBM Tivoli Federated Identity Manager.

Intended audience

The target audience for this book includes 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.

This book describes an implementation of a Web services solution that supports multiple Web services standards. Readers should have knowledge of specific Web services standards, as obtained from the documentation produced by the standards body for each respective standard.

Readers should be familiar with the development and deployment of applications for use in a Web services environment. This includes experience with deploying applications into an IBM WebSphere® Application Server 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. The section also describes how to access Tivoli publications online and how to order Tivoli publications.

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 Guide*
Provides instructions for installing IBM Tivoli Federated Identity Manager.
- *IBM Tivoli Federated Identity Manager Configuration Guide*
Provides instructions for configuring IBM Tivoli Federated Identity Manager.
- *IBM Tivoli Federated Identity Manager Administration Guide*
Provides instructions for completing administration tasks that are required for all deployments.
- *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 of the IBM Tivoli Federated Identity Manager error messages.
- *IBM Tivoli Federated Identity Manager Troubleshooting 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/topic/com.ibm.tivoli.fim.doc_6.2.2/ic/ic-homepage.html

Prerequisite publications

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

- Tivoli Access Manager Information Center:
<http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?toc=com.ibm.itame.doc/toc.xml>
- IBM WebSphere Application Server Version 8.0 Information Center:
<http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/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 IBM Web sites:

- *Enterprise Security Architecture Using IBM Tivoli Security Solutions*. 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/td/tdprodlist.html>

Accessing terminology online

The IBM Terminology Web site consolidates the terminology from IBM product libraries in one convenient location. You can access the Terminology Web site at <http://www.ibm.com/software/globalization/terminology>

Accessing publications online

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli Information Center Web site at <http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp>.

Note: If you print PDF documents on other than letter-sized paper, set the option in the **File → Print** window that allows Adobe Reader to print letter-sized pages on your local paper.

Ordering publications

You can order hard copies of some publications.

Many countries provide an online ordering service.

Follow these steps to access this service:

1. Go to <http://www-947.ibm.com/support/entry/portal/Documentation>
2. Select **IBM Publications Center** from **Getting Started**.
3. Select your country from **Select a country/region/language to begin** and click the arrow icon.
4. Follow the instructions for how to order hard copy publications on Welcome to the IBM Publications Center.

If your country does not provide an online ordering service, contact your software account representative to order publications.

Follow these steps to find your local contact:

1. Go to <http://www.ibm.com/planetwide/>
2. Click your country name to display a list of contacts.

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/topic/com.ibm.tivoli.fim.doc_6.2.2/ic/ic-homepage.html.

Tivoli technical training

For Tivoli technical training information, refer to the following IBM Tivoli Education Web site at <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> and follow the instructions.

IBM Support Assistant

The IBM Support Assistant (ISA) is a free local software serviceability

workbench 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. To install the ISA software, see the *IBM Tivoli Federated Identity Manager Installation Guide*. Also see: <http://www.ibm.com/software/support/isa>.

Troubleshooting Guide

For more information about resolving problems, see the *IBM Tivoli Federated Identity Manager Troubleshooting 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

This publication uses the following typeface conventions:

Bold

- Lowercase commands and mixed case commands that are otherwise difficult to distinguish from surrounding text
- Interface controls (check boxes, push buttons, radio buttons, spin buttons, fields, folders, icons, list boxes, items inside list boxes, multicolumn lists, containers, menu choices, menu names, tabs, property sheets), labels (such as **Tip:**, and **Operating system considerations:**)
- Keywords and parameters in text

Italic

- Citations (examples: titles of publications, diskettes, and CDs)
- Words defined in text (example: a nonswitched line is called a *point-to-point line*)
- Emphasis of words and letters (words as words example: "Use the word *that* to introduce a restrictive clause."; letters as letters example: "The LUN address must start with the letter *L*.")
- New terms in text (except in a definition list): a *view* is a frame in a workspace that contains data.
- Variables and values you must provide: ... where *myname* represents....

Monospace

- Examples and code examples
- File names, programming keywords, and other elements that are difficult to distinguish from surrounding text
- Message text and prompts addressed to the user
- Text that the user must type
- Values for arguments or command options

Operating system-dependent variables and paths

This publication uses the UNIX convention for specifying environment variables and for directory notation.

When using the Windows command line, replace *\$variable* with *% variable%* for environment variables and replace each forward slash (/) with a backslash (\) in directory paths. The names of environment variables are not always the same in

the Windows and UNIX environments. For example, %TEMP% in Windows environments is equivalent to \$TMPDIR in UNIX environments.

Note: If you are using the bash shell on a Windows system, you can use the UNIX conventions.

Chapter 1. Tivoli Federated Identity Manager auditing

Auditing is a process of maintaining detailed secure logs of critical activities in a business environment. The IBM Tivoli Federated Identity Manager Audit module is designed to generate audit events for major operations that occur in the system. The Tivoli Federated Identity Manager components rely on the Audit module to generate and publish events into the audit event records repository.

Audit event records are generated by various components so that your organization can track, record, analyze, and respond to potential security issues. You can use the Tivoli Federated Identity Manager to audit all the critical activities in single sign-on and security token exchanges.

Critical activities are logged as audit event records, which are also known as events, audit events, and audit records in the following sections. IBM Tivoli Federated Identity Manager audit event records are formatted as Common Base Events (CBE). CBE is an XML-based format for defining the structure and attributes of a security event.

When audit events are created, the different Tivoli Federated Identity Manager components use the audit module to process the events. The Audit module component that processes generated events is called the Audit Event Handler.

The Tivoli Federated Identity Manager, Audit module uses Audit Client Profiles to organize configuration information. Each Audit Client Profile defines an Audit Event Handler. You can define more than one Audit Client Profile in the configuration when you use Tivoli Federated Identity Manager. However, only one Audit Client Profile can be active on the system during runtime operations. During event processing, Tivoli Federated Identity Manager passes the generated event to the active Audit Event Handler for consumption.

The Tivoli Federated Identity Manager Audit module relies on a common event format. This format is used regardless of the Audit Client Profile you enabled during runtime. While different technologies can be used to process the events, the format and creation of the event objects is shared.

Tivoli Federated Identity Manager includes two predefined Audit Client Profiles:

- Common Audit Service (*previously named Common Auditing and Reporting Service*) client profile
- CBE XML File audit client profile

You can use Tivoli Federated Identity Manager to write your own Custom Event Handler. Use the predefined profiles or your Custom Event Handler to generate, format, and send audit events. These events can be stored and can interact with any legacy event system.

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.

Use the Tivoli Federated Identity Manager administration console 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 for a more robust data management and reporting facilities.

When the 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 changed information details.

After an auditing configuration change, an event is logged. The event log contains a list of the current state of all the auditing configuration settings. The changed settings are not provided.

You 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 logoff actions.

Name identifier management

Contains information related to identity mapping actions (creation, deletion, update) that are associated with a user 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. Examples of some trust server action are 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.

User Self Care

Contains information related to User Self Care operations.

Chapter 2. Managing audit settings

You can audit Tivoli Federated Identity Manager events to monitor system activities.

Use the Audit Profile Client table in the console to perform the following auditing management tasks:

- “Enabling or disabling auditing”
- “Activating audit client profile”
- “Creating audit client profile” on page 4
- “Modifying audit events” on page 5
- “Modifying audit client profiles” on page 6
- “Deleting audit client profile” on page 6

Enabling or disabling auditing

Auditing is not enabled by default. If you want to monitor system events, you must enable auditing.

About this task

Enabling and disabling event auditing is separate from specifying auditing values, which is done in the administration console. You can set values without enabling auditing. After enabling auditing, you must specify the values.

Procedure

1. Log on to the console.
2. Click **Tivoli Federated Identity Manager > Domain Management > Auditing**. The Auditing panel opens.
3. Select the **Enable audit** check box to enable auditing, or clear the check box to disable auditing.
4. Click **OK** after completing this task.

What to do next

Be sure to review the audit file settings and modify them as appropriate. See “Modifying audit client profiles” on page 6 for more information.

Activating audit client profile

To enable an audit client profile as active, you must activate it.

About this task

Only one profile can be active at any time on the system.

Procedure

1. Log on to the console.
2. Click **Tivoli Federated Identity Manager > Domain Management > Auditing**. The Auditing panel opens.

3. If auditing is not enabled, select the **Enable audit** to activate auditing.
4. In the Audit Profile table, click the applicable profile in the **Select** column.
5. Click **Make Active**. The selected audit profile becomes activated.
6. Click **Apply** to store the changes for the configuration file. The software displays a confirmation message.
7. Click **Load confirmation changes to Tivoli Federated Identity Manager runtime** to reload your changes. The changes take effect immediately. A check mark in the Current® Profile column of the Audit Clients Profiles table indicates that the selected profile is now active.

Creating audit client profile

Use the wizard to create a customized Audit Client module for Tivoli Federated Identity Manager audit events.

Before you begin

Before you begin, complete the following tasks:

- Publish the custom Audit Event Handler plug-ins to the runtime node.
- Gather the parameter names and values that are to be passed to the plug-in.

Procedure

1. Log on to the console.
2. Click **Tivoli Federated Identity Manager > Domain Management > Auditing**. The Auditing Settings panel opens.
3. If auditing is not enabled, select the **Enable audit** check box to activate auditing.
4. In the Audit Profile table, click **Create**. A wizard is displayed.
5. Enter the name and an optional description for the customized audit profile.
6. Select the appropriate event handler of the profile from the **Audit Event Handler** menu. The following pre-configured event handler options are available:
 - **CBEXMLAuditEventHandler**: Specifies a CBE 1.1 file-based event handler. The Common Base Event (CBE) is a standard format for event records. This implementation, unlike the CARS-based event handler, does not have a 2 K size limitation on audit records. Consequently, Tivoli Federated Identity Manager can include additional information about some audit events. If you choose this event handler, specify these additional settings:
 - a. Enter the directory location of the audit log file in the **Audit log location** field. The audit files are located in this directory.
 - b. Enter an appropriate name in the **Audit log file name** field (Optional).
 - c. In the **Maximum audit file size before rollover (MB)** field, specify the maximum size of the audit file in megabytes.
 - d. In the **Maximum number of audit files before rollover** field, specify the maximum number of audit files before overwriting.
 - **Tivoli Common Auditing and Reporting Service (CARS)**: Provides the text-file-based and WebService-based CARS Audit Event Handler. The CARS profile is the event handler implementation that uses the CARS solution for consuming the generated events. CARS provides a file-based audit records repository or a CARS repository accessible by WebServices request.

With this profile, you can integrate Tivoli Federated Identity Manager events to your CARS-based event handler solution for centralized event handling. This implementation has a 2 K size limitation on audit records.

If you choose this event handler, select one of the following options:

- **Audit file (local file).** If you select this option, specify the additional properties:
 - a. In the **Audit log location** field, enter the directory location of the audit log file. The audit files are located in this directory.
 - b. In the **Maximum audit file size before rollover (MB)** field, specify the maximum size of the audit file in megabytes.
 - c. In the **Maximum number of audit files before rollover** field, specify the maximum number of audit files before overwriting.
- **Tivoli Common Auditing and Reporting Service.** If you select this option, specify the additional properties:
 - a. In the **Web Service URL** field, enter the appropriate URL. You can use either a secure (HTTPS) or non-secure (HTTP) connection.
 - b. In the **Disk cache location** field, specify the directory location for the disk cache files. The disk cache records audit events until they can be sent to CARS. The default directory location of the disk cache (on Linux) is: `http://localhost:9080/CommonAuditService/services/EmitterDisk` cache location.
 - c. Click **Web Service Security Settings** to configure the Secure Socket Layer (SSL) and Authentication settings. Refer to the online help for descriptions of the properties.

Note: You can also deploy a custom-coded event handler. To deploy a custom-coded event handler, implement the `AuditEventHandler` interface and deploy it as an OSGI extension.

7. Click **Finish** to store the settings for the new audit client profile. The software displays a confirmation message.
8. Click **Load confirmation changes to Tivoli Federated Identity Manager runtime** to reload your changes.

Modifying audit events

Several component events are selected to be audited by default. You can modify the setting for each component by clearing or selecting its corresponding check box.

Procedure

1. Log on to the console.
2. Click **Tivoli Federated Identity Manager > Domain Management > Auditing**. The Auditing panel opens.
3. Click **Audit Events**. A list of events with corresponding check boxes shows. A check mark indicates that the event is audited. See the online help for descriptions of the events.
4. Clear or select the check boxes to specify the events you want to audit.
5. Click **OK**.
6. Click **Audit Settings** to ensure that you have enabled auditing as described in “Modifying audit client profiles” on page 6.

Modifying audit client profiles

Review the audit file settings and modify them as required.

About this task

You can view and modify the properties of all audit client profiles in the Audit Client Profile table.

Procedure

1. Log on to the console.
2. Click **Tivoli Federated Identity Manager > Domain Management > Auditing**. The Auditing Settings panel opens.
3. If auditing is not enabled, select the **Enable audit** check box to activate auditing.
4. In the Audit Profile table, click the applicable profile in the **Select** column.
5. Click **Properties** which opens the following panels:
 - Profile Name: Use to modify name and description, and to select an Audit Event Handler.
 - Audit Clients: Use to modify the Audit Event Handler settings.
6. Click **Apply** to store the changes for the configuration file. The software shows a confirmation message.
7. Click **Load confirmation changes to Tivoli Federated Identity Manager runtime** to reload your changes.

Deleting audit client profile

To remove an audit client profile from the Audit Client Profile table, you must delete it.

About this task

The Delete button becomes disabled if you select the current (active) profile.

Procedure

1. Log on to the console.
2. Click **Tivoli Federated Identity Manager > Domain Management > Auditing**. The Auditing Settings panel opens.
3. If auditing is not enabled, select the **Enable audit** check box to activate auditing.
4. In the Audit Profile table, click the applicable profile in the **Select** column.
5. Click **Delete**.
6. Click **OK** to confirm. The software displays a confirmation message.
7. Click **Load confirmation changes to Tivoli Federated Identity Manager runtime** to reload your change, which takes effect immediately.

Chapter 3. IBM Tivoli Federated Identity Manager auditing events

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

The 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
- Audit provisioning
- User Self Care

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 1. 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 1. 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

The following 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 2. 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 2. 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 3. 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 4. 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 use the **ReporterComponentId** field.

Federated profiles - single sign-on (IBM_SECURITY_AUTHN)

This event type is used when a user logs on.

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

Table 5. 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 5. 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 the 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 in a federation. 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"
    reportCategory="SECURITY"/>
</situation>
</CommonBaseEvent>

```

Federated profiles - single logout (IBM_SECURITY_AUTHN_TERMINATE)

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

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

Table 6. Elements for an IBM_SECURITY_AUTHN_TERMINATE event

Element	Description
action	The operation that caused the termination of the authentication to occur - log off 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 off. The XPath is: CommonBaseEvent/extendedDataElements [@name='terminateReason']/values
userInfo.appUserName	Information about the user who is logging off. 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"
    reportCategory="SECURITY"/>
</situation>
</CommonBaseEvent>

```

Federated profiles - name identifier management (IBM_SECURITY_FEDERATION)

This event type is generated when a federation event occurs.

An 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 shown in the output of an IBM_SECURITY_FEDERATION event.

Table 7. Elements for an IBM_SECURITY_FEDERATION event

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

Table 7. 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 that the audit generating component takes. 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 8. 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 shows 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 shows 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 8. 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 shows 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 shows 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>
  </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"
      reportCategory="SECURITY"/>
  </situation>
</CommonBaseEvent>

```

Federated profiles - OAuth 1.0 (IBM_SECURITY_TRUST)

This event type is generated by the trust server and logs information about OAuth 1.0 Audit operations. You can use this event to collect audit logs for an OAuth 1.0 protocol.

The success or failure of the following actions performed by the OAuth Client is logged on to the audit log:

- Requesting a set of temporary credentials
- Requesting a verification code
- Requesting a set of token credentials
- Accessing the protected resource

You can enable the OAuth 1.0 Audit logging function through the Integrated Solutions Console.

1. Go to **Tivoli Federated Identity Manager > Domain Management > Auditing > Audit Events**.

2. On the right panel, select the **OAuth 1.0 (IBM_SECURITY_TRUST)** check box.
3. Click **Apply**.

The following table lists the elements and attributes that can be shown in the output of an OAuth 1.0 Audit operation through the use of the IBM_SECURITY_TRUST event.

Table 9. IBM_SECURITY_TRUST elements applicable to OAuth 1.0 Audit

Element	Description
action	The action being performed. Possible actions are: • request • authorize • access • validate The XPath is: CommonBaseEvent/extendedDataElements [@name='action']/values
appliesTo	The federation ID that the request applies to. The XPath is: CommonBaseEvent/extendedDataElements [@name='appliesTo']/values
eventTrailId	The event trail identifier value, for example, FIM_116320b90110104ab7ce9df3453615a1+729829786 The XPath is: CommonBaseEvent/contextDataElements /[@type='eventTrailId']/contextId
issuer	The OAuth Client responsible for issuing the request. The XPath is: CommonBaseEvent/extendedDataElements [@name='issuer']/values
moduleName	The module in the STS module chain that the action is taken on. The XPath is: CommonBaseEvent/extendedDataElements [@name='moduleName']/values
token	The state_id attribute value that remains the same within an OAuth flow. The XPath is: CommonBaseEvent/extendedDataElements [@name='token']/values
tokenInfo	The token request information <i>after</i> changes are made by the module. Only the first 1024 characters of the token are set. When the action is set to one of the possible actions, this element represents the OAuth token information. The XPath is: CommonBaseEvent/extendedDataElements [@name='tokenInfo']/values

The following is an example of an IBM_SECURITY_TRUST event for OAuth 1.0 access token validation:

```
<CommonBaseEvent
creationTime="2011-08-12T08:33:34.937Z"
extensionName="IBM_SECURITY_TRUST"
version="1.1">
  <contextDataElements name="Security Event Factory"
    type="eventTrailId">
    <contextId>FIM_bd1ffad90131183cb58db45c3985443b+2110779641</contextId>
  </contextDataElements>
  <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>Not Available</values>
  </extendedDataElements>
  <extendedDataElements name="action" type="string">
    <values>validate</values>
  </extendedDataElements>
  <extendedDataElements name="tokenInfo" type="string">
    <values>
      <oauth_audit>
        <attr name="state_id"
          type="state"><v>MiMBrw3Ebe4R86BexAXQ</v></attr>
        <attr name="host"
          type="request"><v>oauth.server.com</v></attr>
        <attr name="authorized"
          type="response:decision"><v>TRUE</v></attr>
        <attr name="scope"
          type="response:attribute"></attr>
        <attr name="username"
          type="response:attribute"><v>wasadmin</v></attr>
        <attr name="expires"
          type="response:decision"><v>2011-08-19T08:33:32Z</v></attr>
        <attr name="path"
          type="request"><v>/fimvt/oauth/sfprotected.jsp</v></attr>
        <attr name="oauth_token"
          type="response:attribute"><v>4tUY0Fo2Cux1wYTciMkn</v></attr>
        <attr name="token_type"
          type="request"><v>validate</v></attr>
      </oauth_audit>
    </values>
  </extendedDataElements>
  <extendedDataElements name="moduleName" type="string">
    <values>com.tivoli.am.fim.trustserver.sts.modules.OAuthTokenModule</values>
  </extendedDataElements>
  <extendedDataElements name="appliesTo" type="string">
    <values>https://oauth.server.com/FIM/MySocialNetwork/oauth10</values>
  </extendedDataElements>
  <extendedDataElements name="token" type="string">
    <values>MiMBrw3Ebe4R86BexAXQ</values>
  </extendedDataElements>
  <extendedDataElements name="issuer" type="string">
    <values>urn:ibm:ITFIM:oauth:consumer:2XGs49n6DoYydYpJjuaE</values>
  </extendedDataElements>
  <sourceComponentId application="ITFIM#6.2.2"
    component="IBM Tivoli Federated Identity Manager"
    componentIdType="ProductName"
    executionEnvironment="Linux[x86]#2.6.38-10-generic-pae"
    location="oauth.server.com" locationType="FQHostname"
    subComponent="validate" threadId="WebContainer : 3"
    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>
```

Federated profiles - OAuth 2.0 (IBM_SECURITY_TRUST)

This event type is generated by the trust server and logs information about OAuth 2.0 Audit operations. You can use this event to collect audit logs for an OAuth 2.0 protocol.

The success or failure of the following actions performed by the OAuth Client is logged in to the audit log:

- Requesting an authorization code
- Requesting an access token
- Accessing the protected resource

You can use the OAuth Audit logging function through the Integrated Solutions Console.

1. Go to **Tivoli Federated Identity Manager > Domain Management > Auditing > Audit Events**.
2. On the right panel, select the **OAuth 2.0 (IBM_SECURITY_TRUST)** check box.
3. Click **Apply**.

The following table lists the elements and attributes that can be shown in the output of an OAuth 2.0 Audit operation through the use of the IBM_SECURITY_TRUST event.

Table 10. IBM_SECURITY_TRUST elements applicable to OAuth 2.0 Audit

Element	Description
action	The action being performed. Possible actions are: • authorization • access_token • resource The XPath is: CommonBaseEvent/extendedDataElements [@name='action']/values
appliesTo	The federation ID that the request applies to. The XPath is: CommonBaseEvent/extendedDataElements [@name='appliesTo']/values
eventTrailId	The event trail identifier value, for example, FIM_116320b90110104ab7ce9df3453615a1+729829786 The XPath is: CommonBaseEvent/contextDataElements /[@type='eventTrailId']/contextId
issuer	The OAuth Client responsible for issuing the request. The XPath is: CommonBaseEvent/extendedDataElements [@name='issuer']/values
moduleName	The module in the STS module chain that the action is taken on. The XPath is: CommonBaseEvent/extendedDataElements [@name='moduleName']/values

Table 10. IBM_SECURITY_TRUST elements applicable to OAuth 2.0 Audit (continued)

Element	Description
token	The state_id attribute value that remains the same within an OAuth flow. The XPath is: CommonBaseEvent/extendedDataElements [@name='token']/values
tokenInfo	The token request information <i>after</i> changes are made by the module. Only the first 1024 characters of the token are set. When the action is set to one of the possible actions, this element represents the OAuth token information. The XPath is: CommonBaseEvent/extendedDataElements [@name='tokenInfo']/values

The following is an example of an IBM_SECURITY_TRUST event for OAuth 2.0 access token validation:

```
<CommonBaseEvent
creationTime="2011-08-15T02:47:41.741Z"
extensionName="IBM_SECURITY_TRUST"
version="1.1">
  <contextDataElements name="Security Event Factory"
    type="eventTrailId">
    <contextId>FIM_cb5663ae01311e5ca446bffd0ba9d0ec+1544491284</contextId>
  </contextDataElements>
  <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>Not Available</values>
  </extendedDataElements>
  <extendedDataElements name="action" type="string">
    <values>resource</values>
  </extendedDataElements>
  <extendedDataElements name="tokenInfo" type="string">
    <values>
      <oauth20_audit>
        <attr name="host"
          type="request"><v>oauth.server.com</v></attr>
        <attr name="authorized"
          type="response:decision"><v>TRUE</v></attr>
        <attr name="oauth_token_client_id"
          type="response:attribute"><v>fEQdScx86FFCPQyxomMK</v></attr>
        <attr name="username"
          type="response:attribute"><v>fEQdScx86FFCPQyxomMK</v></attr>
        <attr name="expires"
          type="response:decision"><v>2011-08-15T02:49:19Z</v></attr>
        <attr name="access_token"
          type="param"><v>EhwwNSaLiUmcBLNcuWud</v></attr>
        <attr name="path"
          type="request"><v>/fimivt/oauth/sfprotected.jsp</v></attr>
        <attr name="access_token"
          type="response:attribute"><v>EhwwNSaLiUmcBLNcuWud</v></attr>
      </oauth20_audit>
    </values>
  </extendedDataElements>
  <extendedDataElements name="moduleName" type="string">
    <values>com.tivoli.am.fim.trustserver.sts.modules.OAuth20TokenModule</values>
  </extendedDataElements>
  <extendedDataElements name="appliesTo" type="string">
    <values>https://oauth.server.com/FIM/MySocialNetwork/oauth20</values>
  </extendedDataElements>
```

```

<extendedDataElements name="token" type="string">
  <values>uudcb565cd7-0131-13bc-8c49-bffd0ba9d0ec</values>
</extendedDataElements>
<extendedDataElements name="issuer" type="string">
  <values>urn:ibm:ITFIM:oauth20:token:bearer</values>
</extendedDataElements>
<sourceComponentId application="ITFIM#6.2.2"
  component="IBM Tivoli Federated Identity Manager"
  componentIdType="ProductName"
  executionEnvironment="Linux[x86]#2.6.38-10-generic-pae"
  location="oauth.server.com" locationType="FQHostname"
  subComponent="resource" threadId="WebContainer : 6"
  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>

```

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 shown in the output of an IBM_SECURITY_TRUST event.

Table 11. 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 out 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 that the request or token applies to. The XPath is: CommonBaseEvent/extendedDataElements [@name='appliesTo']/values
issuer	The party responsible for issuing the token. The XPath is: CommonBaseEvent/extendedDataElements [@name='issuer']/values

Table 11. Elements for an IBM_SECURITY_TRUST event (continued)

Element	Description
moduleName	The module in the STS module chain that the action is taken on. The XPath is: CommonBaseEvent/extendedDataElements [@name='moduleName']/values
ruleName	The rule name used for the mapping module. This element is filled out only when specified action is set to <i>map</i> . The XPath is: CommonBaseEvent/extendedDataElements [@name='ruleName']/values
token	The incoming token that the action is being taken on. Only the first 1024 characters of the token are set. When the action is set to <i>map</i> , this element represents the incoming principal. The XPath is: CommonBaseEvent/extendedDataElements [@name='token']/values
tokenInfo	The internal representation of the user information <i>after</i> changes are made by the module. Only the first 1024 characters of the token are set. When action is set to <i>map</i> , this element represents the outgoing principal. When the action is set to <i>authorize</i> , this element represents the principal for whom the access decision was made. The XPath is: CommonBaseEvent/extendedDataElements [@name='tokenInfo']/values
tokenType	The type of token the module is using. The XPath is: CommonBaseEvent/extendedDataElements [@name='tokenType']/values

Samples of 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. 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"?>
      <stsuser:STUniversalUser xmlns:stsuser="urn:ibm:names:ITFIM:1.0:stsuser">
        <stsuser:Principal>
          <stsuser:Attribute name="NameQualifier"
            type="urn:oasis:names:tc:SAML:1.0:assertion"/>
          <stsuser:Attribute name="name"
            type="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">
            <stsuser:Value>BigDummy</stsuser:Value>
          </stsuser:Attribute>
        </stsuser:Principal>
        <stsuser:AttributeList>
          <stsuser:Attribute name="ssn" type="http://example.com/federation/v1/namevalue">
            <stsuser:Value>
              <stsuser:Value xmlns:stsuser=
                "urn:ibm:names:ITFIM:1.0:stsuser">555-55-555</stsuser:Value>
            </stsuser:Value>
          </stsuser:Attribute>
          <stsuser:Attribute name="MinorVersion"
            type="urn:oasis:names:tc:SAML:1.0:assertion">
            <stsuser:Value>1</stsuser:Value>
          </stsuser:Attribute>
          <stsuser:Attribute name="email" type="http://example.com/federation/v1/email">
            <stsuser:Value>elain@hotmail.com</stsuser:Value>
          </stsuser:Attribute>
        <stsuser: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>
  </extendedDataElements>

```

```

<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.SAMLSecurityModuleBase"
  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 shown in the output of an IBM_SECURITY_ENCRYPTION event.

Table 12. 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"
      reportCategory="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 shown in the output of an IBM_SECURITY_SIGNING event.

Table 13. 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/xmlsig#}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"
      reportCategory="SECURITY"/>
  </situation>
</CommonBaseEvent>
```

Runtime configuration management (IBM_SECURITY_MGMT_POLICY)

This event type generated by IBM Tivoli Federated 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 shown in the output of an IBM_SECURITY_MGMT_POLICY event.

Table 14. 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 31 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

Table 14. Elements for an IBM_SECURITY_MGMT_POLICY event (continued)

Element	Description
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
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 186 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 15. XPaths for shredding and staging attributes

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

Table 15. XPath for shredding and staging attributes (continued)

Field	XPath
policyInfo attributes FederationId value	CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'FederationId')] /../../children [@name='value']/values
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 attributes SAML1.SigningKeyIdentifier	CommonBaseEvent/extendedDataElements [@name='policyInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='name']/values [contains (.,'SAML1.SigningKeyIdentifier')]
policyInfo attributes 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 16. 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 16. 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 16. 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 16. 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 16. 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 attribute 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 16. 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 17. 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 17. 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 17. 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 17. 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 17. 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 17. 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 17. 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 18. 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 18. 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 18. 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 attribute 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 18. 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 attribute 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 18. 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 18. 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 18. 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 must 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 18. 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 18. 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 19. 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 19. 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 19. 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 19. 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 19. 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 attribute 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 20. 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 21. 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 21. 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>
    <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"
  </sourceComponentId>

```

```

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"
    reportCategory="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 22. 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 22. 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=<i>domain_name</i> AuditLogLocation=<i>path</i> CacheLocation=<i>path</i> WebServiceBasicAuthUsername=<i>username</i> WebServiceBasicAuthPassword=<i>password</i> WebServiceKeyIdentifier=<i>keyname</i> WebServiceURL=<i>URL</i> MaxAuditFiles=<i>number</i> AuditFileSize=<i>number</i> UseWebServiceBasicAuth=true false WebServiceKeystore=<i>keystore_name</i> 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 shown in the output of an IBM_SECURITY_MGMT_PROVISIONING event.

Note: 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 23. Elements used in IBM_SECURITY_MGMT_PROVISIONING events

Element	Description
action	The action that was performed. The supported action is add. The XPath is: CommonBaseEvent/extendedDataElements [@name='action']/values
mgmtInfo. command	Information about the management operation. The supported management operation is 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 24 on page 69 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 23. 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: <i>username</i> from federation endpoint: <i>URL_of_getcard.crd_endpoint</i>" 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 user name 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	User name of the user who is downloading a card. The XPath is: CommonBaseEvent/extendedDataElements [@name='userInfo']/children [@name='appUserName']/values

Table 24. 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 24. 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 24. Attributes for Cardspace Events (continued)

Attribute	Definition
ClaimURI	Supported claims for the information card. A separate ClaimURI attribute is present for each claim that the 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>
</extendedDataElements>

```

```

<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/
cameratyp</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>

```

User Self Care (IBM_SECURITY_SELFCARE)

This event type provides information about User Self Care operations.

The following table lists the elements and attribute that can be shown in the output of an IBM_SECURITY_SELFCARE event.

Table 25. IBM_SECURITY_SELFCARE elements

Element	Description
action	Type of operation being performed. The supported operations are: • changePassword • accountCreate • accountDelete • accountRecovery • profileMgmt The XPath is: CommonBaseEvent/extendedDataElements [@name='action']/values
userInfo.appUserName	Name of the user requesting the operation. For some operations, the name can be the user who is handling the request. The XPath is: CommonBaseEvent/extendedDataElements [@name='userInfo']/children [@name='appUserName']/values
registryInfo.name	Name of the registry used the user self care operation. This value is always set to the User Self Care operationID. The XPath is: CommonBaseEvent/extendedDataElements [@name='registryInfo']/children [@name='name']/values
registryInfo.type	Type of the registry used by the User Self Care operation. This value is always set to STS. The XPath is: CommonBaseEvent/extendedDataElements [@name='registryInfo']/children [@name='type']/values

Table 25. IBM_SECURITY_SELFCARE elements (continued)

Element	Description
registryObjectInfo.nameInApp	Name of the user for whom the operation is requested. The XPath is: CommonBaseEvent/extendedDataElements [@name='registryObjectInfo']/children [@name='nameInApp']/values
registryObjectInfo.type	The type of the registry object. Types are User, Group, or Other. The XPath is: CommonBaseEvent/extendedDataElements [@name='registryObjectInfo']/children [@name='type']/values
registryObjectInfo.description	Description of the registry object. The XPath is: CommonBaseEvent/extendedDataElements [@name='registryObjectInfo']/children [@name='description']/values
registryObjectInfo.attributes	Contain different attributes for the registryObjectInfo object. See the table below for attributes that can be included for this event. The presence of these attributes depends on the event action. Not all attributes are present for each event. Each attribute consists of a source, name, and value. The value of the source is either application if the attribute related to application or user if the attribute is related to the user in the operation. 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

The following table shows attributes for the User Selfcare event.

Table 26. Attributes for User Selfcare events

Attribute	Definition
usc.form.email.address	The user email address. 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(., 'usc.form.email.address')] The XPath for the attribute value is: CommonBaseEvent/extendedDataElements [@name='registryObjectInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='value']/values(., 'usc.form.email.address')] /../../children[@name='value']/values
usc.form.userid	The user ID. 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(., 'usc.form.userid')] The XPath for the attribute value is: CommonBaseEvent/extendedDataElements [@name='registryObjectInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='value']/values(., 'usc.form.userid')] /../../children[@name='value']/values

Table 26. Attributes for User Selfcare events (continued)

Attribute	Definition
usc.useragent.hostname	The user agent hostname. 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(.,'usc.useragent.hostname')] The XPath for the attribute value is: CommonBaseEvent/extendedDataElements [@name='registryObjectInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='value']/values(.,'usc.useragent.hostname')] /../../children[@name='value']/values
usc.useragent.ipaddr	The user agent IP address. 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(.,'usc.useragent.ipaddr')] The XPath for the attribute value is: CommonBaseEvent/extendedDataElements [@name='registryObjectInfo']/children [@name='attributes']/children [@name='attribute']/children [@name='value']/values(.,'usc.useragent.ipaddr')] /../../children[@name='value']/values

The following is an example of an IBM_SECURITY_SELFCARE event for the accountCreate operation:

```
<CommonBaseEvent creationTime="2010-02-10T20:44:34.765Z"
  extensionName="IBM_SECURITY_SELFCARE" version="1.0.1">
  <contextDataElements name="Security Event Factory"
    type="eventTrailId">
    <contextId>
      FIM_b9a04859012610189090bdf095cc2ab+164748191
    </contextId>
    </contextDataElements>
  <extendedDataElements name="registryObjectInfo" type="noValue">
    <children name="type" type="string">
```

```

    <values>user</values>
  </children>
  <children name="attributes" type="noValue">
    <children name="attribute" type="noValue">
      <children name="value" type="string">
        <values>9.76.37.100</values>
      </children>
      <children name="name" type="string">
        <values>usc.useragent.ipaddr</values>
      </children>
      <children name="source" type="string">
        <values>user</values>
      </children>
    </children>
    <children name="attribute" type="noValue">
      <children name="value" type="string">
        <values>anho@us.ibm.com</values>
      </children>
      <children name="name" type="string">
        <values>usc.form.email.address</values>
      </children>
      <children name="source" type="string">
        <values>user</values>
      </children>
    </children>
    <children name="attribute" type="noValue">
      <children name="value" type="string">
        <values>guscuser4</values>
      </children>
      <children name="name" type="string">
        <values>usc.form.userid</values>
      </children>
      <children name="source" type="string">
        <values>user</values>
      </children>
    </children>
    <children name="attribute" type="noValue">
      <children name="value" type="string">
        <values>grizzly.austin.ibm.com</values>
      </children>
      <children name="name" type="string">
        <values>usc.useragent.hostname</values>
      </children>
      <children name="source" type="string">
        <values>user</values>
      </children>
    </children>
  </children>
  <children name="description" type="string">
    <values>USC User Account</values>
  </children>
  <children name="nameInApp" type="string">
    <values>guscuser4</values>
  </children>
</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="registryInfo" type="noValue">
  <children name="type" type="string">
    <values>STS</values>
  </children>

```

```

<children name="name" type="string">
  <values>usc/self/account/create/validate/get</values>
</children>
<children name="serverPort" type="string">
  <values></values>
</children>
<children name="locationType" type="string">
  <values></values>
</children>
<children name="location" type="string">
  <values></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>
<extendedDataElements name="action" type="string">
  <values>accountCreate</values>
</extendedDataElements>
<sourceComponentId application="ITFIM#6.2.1"
  component="IBM Tivoli Federated Identity Manager"
  componentIdType="ProductName"
  executionEnvironment="Windows XP[x86]#5.1 build 2600 Service Pack 2"
  location="grizzly.austin.ibm.com" locationType="FQHostname"
  subComponent="com.tivoli.am.fim.trustserver.sts.modules.USCAccountCreateSTSModule"
  threadId="WebContainer : 3"
  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 4. Overview of the Common Audit Service

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

The Common Audit Service consists of the following parts:

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.

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 IBM 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 IBM Tivoli Access Manager for e-business.

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 84 illustrates how data flows between the components of Common Audit Service and an exploiting product, such as Tivoli Federated Identity Manager.

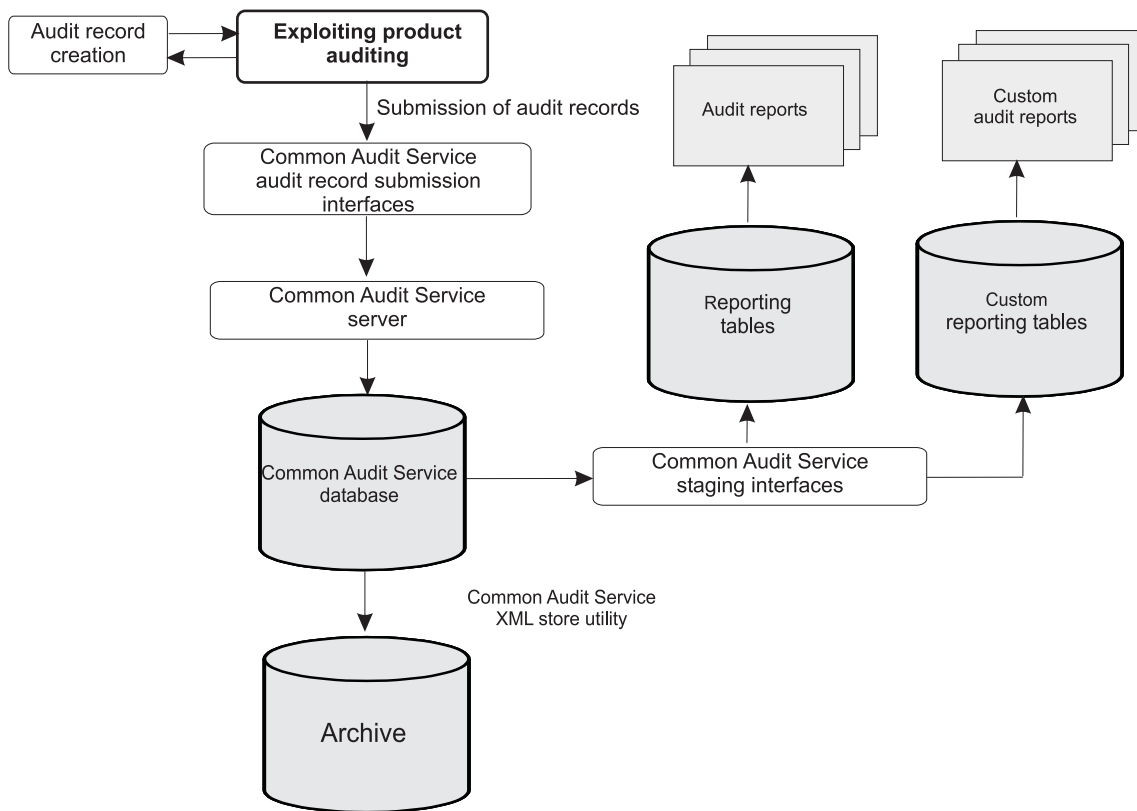


Figure 1. Structure of Common Audit Service

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

This topic describes a scenario for collecting audit data.

Procedure

1. Identify the installed IBM security software.

For example, you might have Tivoli Federated Identity Manager and IBM 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 must configure Tivoli Federated Identity Manager to send the IBM_SECURITY_TRUST event.

See 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 determine the type of reports, frequency of staging, and so on.

For example, if you are generating events in millions, you must archive and prune archived data more frequently. If you prune frequently, then you must limit your ability to run security events details reports.

Security events would also 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. This scenario assumes that the number of security events is 100,000 per day.

4. Perform the data management tasks:

- a. Stage the data for generating reports. You must stage the events from the XML data store tables to report tables. You can run the staging utility in incremental mode every day shortly after midnight, for example, at 12:05 AM. For more information, see “Running the staging utility command” on page 150.

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

The archival process consists of four phases:

- Pre-archive
- Archive by using an archival tool
- Prune the data from the report tables
- Post-archive

Use the XML data store utilities for these processes. For information about these utilities, see “Running the XML data store utilities” on page 151.

- 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. Use the first timestamp from the pre-archive phase as input.

For more information, see “Running the staging utility command” on page 150.

5. Generate reports by using the report-generation tool of the exploiting product, for example, Tivoli Federated Identity Manager or IBM Tivoli Access Manager for e-business.

The following scenario shows a process to generate a report for a security incident investigation:

- A security incident investigation is required 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. You must configure the parameters to review the events in question, and to show the report in a useful format.

- Run the report for the day in which you are interested. The start time is 2:00 AM and end time is 6:00 AM.

Chapter 5. Installing Common Audit Service

The Common Audit Service requires specific features to be installed. This section guides you in the tasks required 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 7, “Upgrading the Common Audit Service audit server from earlier versions,” on page 121.

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 Tivoli Federated Identity Manager 6.2.2 Release Information.

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 89 for more instructions.

Installing the DB2 client on Windows systems

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

Procedure

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

What to do next

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.

Procedure

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 `admc1` directory.
4. Select **Install Products**.
5. Select the radio button for the DB2 client that you are installing.
6. Click **Next** in the Welcome to the DB2 Setup wizard.
7. Click **I accept the terms in the license agreement** if you accept the terms of the license agreement.
8. Click **Next**.
9. Select **Typical** installation type.
10. Click **Next**.

11. Select **Create a DB2 instance** in the Set up a DB2 instance window.
12. Select the **New User** radio button and specify a user name and password. You can either accept the defaults or change 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.
13. Click **Next**.
14. Click **Finish**.

What to do next

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 cataloging 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 to use for the installation. See Table 27 on page 95.
- Verify that there is enough space to install the Common Audit Service audit server. Additionally, the system temp directory must have approximately 20 MB to 50 MB for unpacking the audit server installation JAR file during the installation process. After the installation completes, the temporary directories are removed and the file space is reclaimed.
- 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. Doing so installs 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.
- Use a root or non-root administrator ID to install Common Audit Service.
- 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:

- 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 94.

-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.

Before you begin

See “Interactive installation” on page 92 for the command you enter to begin the audit server installation. Run the command from the media that contains the audit server installation programs.

Procedure

1. Select the language that you want to use for the installation. The default is English.
2. Read the license agreement.
3. Click **Next** if you agree with the license agreement, or click **Cancel** to exit the program. The Welcome dialog is displayed, indicating that the installation will install the Common Audit Service component.
4. Click **Next** to continue.
5. 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.
6. Click **Next** to continue.
7. 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.
8. Click **Next** to continue. The installation wizard searches for WebSphere Application Server installations that can be used as target locations to install Common Audit Service.
9. 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. See “Audit server installation options” for information on the default directory path.
10. Click **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.
11. If WebSphere Application Server global security is set, you are prompted in the next window to enter the WebSphere Application Server administrator ID and password.
12. Click **Next** to continue.
13. In the summary window, ensure that all the information that is shown is correct.
14. If you need to make a change, click **Previous** to return to a previous window; otherwise, click **Next** to continue.
15. 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 27. 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
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 <i>WAS_HOME</i> /AppServer/systemApps/isclite.ear directory. CarsConfigUtil.jar Comprises the utility class for the configuration MBean. This file is extracted into the <i>CARS_HOME</i> /server/cons_lib directory.

Table 27. Interactive installation options and values (continued)

Configuration option	Description
WebSphere Application Server Profile Directory	Specifies the directory path of the WebSphere Application Server profile where you are deploying Common Audit Service. If the installation wizard detects a WebSphere Application Server profile, the <i>WAS_HOME</i> value from this installation is used to create the default profile directory path as: <i>WAS_HOME/AppServer/profiles/AppSrv01</i> If a WebSphere Application Server installation is not detected, the default path is defined as: <i>platform_dependent_install_path/IBM/WebSphere/AppServer/profiles/default</i> where <i>platform_dependent_install_path</i> 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.
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 92 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:java_home
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 (supported by management console only)
- Chinese (traditional)
- French
- German
- Japanese
- Czech
- Hungarian
- Italian
- Korean
- Polish
- Portuguese (Brazil)
- Russian

- Simplified Chinese
- Spanish

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

Before you begin

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.

Procedure

1. Log on as root or as an administrative user.
2. Change to the installation root directory. See the Readme 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.
7. Click **Next**.
8. Select the language packages you want to install.
9. Click **Next**. A dialog showing the location and features of the languages that you selected opens.
10. To accept the languages selected, click **Next**.
11. After installation has completed, click **Finish** to exit the wizard.

What to do next

If it is necessary to uninstall the language support packages, see “Uninstalling language support packages” on page 134 for instructions.

Customizing the XML store data definition language script

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

Before you begin

If you must customize the schema, customize it *before* you configure the XML data store (database) for Common Audit Service. Use the configuration console to customize the schema. You must create a database during configuration to have a customized XML data store. Consequently, the steps described in this topic are not applicable under the following circumstances:

- If you are upgrading from an older version of Common Audit Service to version 6.1, and
- If you are using an existing audit database

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

You must also consider customizing the buffer pool size to minimize disk I/O, and customizing the database configuration parameters. Creating custom report tables for the purposes of generating custom reports can be done post configuration.

Procedure

1. Identify the changes to the basic data definition language script that you must make to meet your custom storage needs. The changes include:
 - Identifying the parameters to be modified
 - Identifying the suitable values for those parameters to set in the script
2. Save a copy of the original data definition language script that was included in the Common 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. Use a text editor to customize the script by editing the `cr_dbobjects.db2` script. 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 cause failure in the configuration of the Common Audit Service Server. If a configuration error occurs after customizing the script, see the configuration logs. The configuration logs indicates 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. Use the GUI configuration console to configure the audit server to customize the script. See Chapter 6, “Configuring the audit server,” on page 101 for details.

Chapter 6. Configuring the audit server

Use the configuration console to configure the Common Auditing and Reporting Service.

This topic describes how to configure the Common Audit Service Version 6.1 audit server using the configuration console. This topic also describes the required tasks to use the Common Audit Service audit server in a WebSphere Application Server clustered environment.

Pre-configuration checklist for all platforms

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

Procedure

1. 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 (svcname) 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 89.
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, ensure that the following conditions are met:

- The remote DB2 node has been cataloged.
- The audit database of the previous version of Common Audit Service that is 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.

Before you begin

See “Pre-configuration checklist for all platforms” on page 101 for the items you need to consider before you start to configure the use of Common Audit Service.

Procedure

1. Open a Web browser and set the value of the URL to the administrative console port of the WebSphere Application Server deployment manager or stand-alone server.

The value is the same as the one specified as the target profile during installation of the Audit Configuration Console.

The default port value is 9060 or 9043 for a secure console.

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

2. Log on as a WebSphere Application Server administrator.
3. 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 103.

The Welcome dialog opens, indicating that Common Audit Service must be configured before the application can be used.

4. Click **Next** to continue.
5. In the **Common Audit Service Host** window, enter the host name and SOAP port number of the target WebSphere Application Server process (deployment manager or stand-alone single server) where Common Audit Service is installed.
6. Click **Next** to continue.
7. In the **WebSphere Security** window, if global security is enabled on the target WebSphere Application Server process, select the **Global Security** check box, enter the WebSphere Application Server administrator name, and enter the password.
8. Click **Next** to continue.
9. In the **WebSphere Target Mapping** window, select the configuration target. The list of clusters and independent servers that are available for deployment are shown 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 Application Server and restart the configuration.
10. 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 103.
 - Database Instance Owner ID
 - Database Instance Owner Password
 - Database Instance Profile Path

For 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

11. Click **Next** to continue.
12. In the **JDBC Connector** window, enter the following information to configure the JDBC driver that is used to connect to the database.
 - Database Server Host Name
 - Database Server TCP Service Port
 - JDBC Driver Path
13. Click **Next** to continue.
14. Review the list of options you have selected in the configuration Summary window. If the options are correct, select **Finish** to begin the configuration.
15. 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.
16. 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.
17. 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 configuration console.

Table 28. 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.

Table 28. Interactive configuration values and options (continued)

Configuration option	Description
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.
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.

Table 28. Interactive configuration values and options (continued)

Configuration option	Description
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 <i>DB2_INSTALL_ROOT</i> /java on UNIX and Linux platforms, and in <i>DB2_INSTALL_ROOT</i> \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 manual 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” on page 106 or “Configuring JDBC resources against a homogeneous cluster” on page 106 using the WebSphere Application Server Administrative Console of the container Deployment Manager.

Configuring JDBC resources against a heterogeneous cluster

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

Procedure

1. Log on 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 the changes to take effect.

Configuring JDBC resources against a homogeneous cluster

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

Procedure

1. Restart all nodeagents from the administrative console of the Deployment Manager.
2. Restart the Deployment Manager itself.
3. Restart the target cluster from the administrative console of the Deployment Manager.

Configuring the compress property

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

About this task

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

Procedure

1. Edit the `WAS_HOME/profile/profilename/config/ibmcars/ibmcarsserver.properties` file and set the value of the `xmlstore.compress` property to false (`xmlstore.compress=false`).
2. Restart the 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. The data storage process must 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.

Procedure

1. Ensure that you have a functioning WebSphere Application Server cluster. For instructions on setting up a clustered environment, see the WebSphere Application Server Information Center: <http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp>.
2. Ensure that the 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.
4. If no web servers exist, select **Servers > Web servers > New** to create a new Web server from the wizard.
5. In Step 1 of the wizard, complete the fields as follows.

Select Node

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

Server name

Enter the Web server name.

Type Leave the default as **IBM HTTP Server**.

6. Click **Next**.
7. In Step 2, select the IHS template radio button.
8. Click **Next**.
9. In Step 3, complete the default fields.

Port

Typically you can leave the default value as 80.

Web server installation location

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

Plug-in installation location

Leave the default value as **all**.

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

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

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

About this task

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

See the WebSphere Application Server Information Center: <http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp> for detailed instructions on configuring a Web server and an application on separate systems (remote).

See “Configuring the Web server plug-in for SSL” on page 145 for information about securing communication with the Web server using SSL.

Procedure

1. Ensure that you have 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.
 - c. Click **Next**.
 - d. Read and accept the license agreement (if you agree with its terms).
 - e. Click **Next**.
 - f. If the prerequisite check is passed, click **Next**; otherwise correct the prerequisites and restart the installation.
 - g. Select the type of Web server you are configuring.
 - h. Click **Next**.
 - i. Select **Web server machine (remote)**.

- j. Click **Next**.
- k. Accept the default location for the plug-ins installation root directory.
- l. Click **Next**.
- m. Browse for the configuration file of the Web server.
- n. Click **Next**.
- o. Specify a name for the Web server. WebSphere Application Server uses this name to manage the Web server.
- p. Click **Next**.
- q. Accept the default location for the plugin-cfg.xml file that is created on the Web server host.
- r. Click **Next**.
- s. 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.
- t. Click **Next**.
- u. Review the summary information to ensure the specified settings are correct.
- v. Click **Next**.
- w. Click **Next** on the pre-installation summary window to start the installation.
- x. If the post-installation window shows that the installation was successful, click **Next**; otherwise, correct any problems and reinstall the plug-in.
- y. Close the installation roadmap.
- z. Click **Finish** to exit the wizard.

On UNIX or Linux systems, the plug-in script is the file *plug-in_installation_root/bin/configurewebserver_name.sh*

On Windows systems, the plug-in script is the file *plug-in_installation_root\bin\configurewebserver_name.bat*

where *plug-in_installation_root* is the value specified in step g, and *webserver_name* is the value specified in step i.

aa. 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/
configurewebserver_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.

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

Use the WebSphere Application Server Administrative Console to propagate the plug-in if the IBM HTTP Server is installed on a WebSphere Application Server node host.

Procedure

1. Connect to the WebSphere Application Server Administrative Console on the deployment manager system:
 - a. Expand **Servers**.
 - 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.
3. Click **Web servers**.
4. Select the check box for the Web server you are using.
5. Select **Propagate Plug-in**.

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

Use the WebSphere Application Server Administrative Console to propagate the plug-in if the IBM HTTP Server is installed outside of the cluster.

Procedure

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 the IBM HTTP Server and the HTTP administrative server.
4. Restart the IBM HTTP Server and the HTTP administrative server.
5. Stop the cluster.
6. 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. This topic describes steps for mapping the Common Audit Service to either the virtual host or the target servers.

Mapping the Common Audit Service server to the virtual host

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

Procedure

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.

About this task

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

Procedure

1. In the WebSphere Application Server Administrative Console, click **Applications > Enterprise Applications > CommonAuditService > Manage Modules**.
2. Verify that the IBMCARXmlstores-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. The procedures use the WebSphere Application Server Administrative Console to inspect the deployment parameters of various application components of Common Audit Service. The verification procedures also use the IBM DB2 command-line interface to review the database instance that must be present after a correct deployment.

Verifying the configuration settings for the Common Audit Service application

Verify that the Common Audit Service application is configured correctly.

Complete the following procedures in the WebSphere Application Server Administrative Console.

- **Determine that all modules are present:**
 1. Click **Applications > Enterprise Applications > CommonAuditService > Manage Modules**.
 2. Ensure that the **IBMCARSxmlstores-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**
Must be checked.
 - Allow overflow**
Must be checked.
 - Maximum in-memory session count**
Must be set to 1000 sessions.
 - Set timeout**
Must 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.xml
 - 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
Must be checked.

Use replication settings from EJB container
Must 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

Verify that the data source used by the EJB component is configured correctly and can connect to the audit database.

Complete the following procedures in the WebSphere Application Server Administrative Console to verify the configuration settings for the Common Audit Service data source.

- **Verify the JDBC provider:**
 1. Click **Resources > JDBC > JDBC Providers**.
 2. Ensure that the scope setting is set to **All scopes**.
 3. Verify the following settings in the table:

- Name=Event_DB2Xml_JDBC_Provider
 - Scope=*expected_cluster_or_server_scope*
 - Description=DB2 Universal JDBC Driver Provider (XA) for the Common 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 must be shown 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 must be shown 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

Verify that the data store is set up correctly for Common Audit Service.

Complete the following command-line procedures to verify the configuration settings for the Common Audit Service data store.

- **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 must show:

```

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 must show:

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

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

Verify that the Common Audit Service web service component is running correctly.

Complete the following procedures in the WebSphere Application Server Administrative Console.

- **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. The green right arrow indicates that the Application Status is running.

- **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 shows:

```
{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 show the general properties of the certificate.

- f. Enter a directory path and file name for the certificate file name.
- g. Click **OK** to extract the certificate from the keystore.

Note: 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 from the current host, copy the certificate file obtained in step 1 to the browser host.
 - b. Select **Edit > Preferences > Advanced > Security**.
 - c. Click **View Certificates**.
 - d. Select the **Web Sites** tab to view the list of site certificates.
 - e. Click **Import**.
 - f. Go to the certificate file obtained in Step a, or from Step 1 under **Obtain a security certificate**.
 - g. Click **Open**. The certificate must show in the list of certificates that shows.
 - 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 shows:

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

Hi there, this is a Web service!

Deploying the Java stored procedure for an audit details report

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

Before you begin

Customized reports can also access the audit details of the Java stored procedure. See “Creating custom reports” on page 175 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.)

Procedure

1. **Linux only:** See “Setting up to run the Java stored procedures on Linux.”
2. **All platforms:** See “Setting the `jdk_path` parameter.”
3. **All platforms:** Run the `ibmcarsddinst` script. See “Running `ibmcarsddinst` to deploy the Java stored procedure” on page 119.
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 120.

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 runtime linker must be able to access certain Java shared libraries. DB2 must also 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.

Before you begin

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

Procedure

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

```
db2 get dbm cfg
```

Look for the `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. See the *IBM DB2 Command Reference* for more information.

Running `ibmcarsddinst` to deploy the Java stored procedure

Use the following information to run the operating system-specific `ibmcarsddinst` script to deploy the Java stored procedures on the server machine. The Common 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

The 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 IBM_CARS_DD_REPORT Java stored procedure

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

Procedure

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

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

Where the *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 "IBM_CARS_DD_REPORT" of type
"PROCEDURE" having compatible arguments was found.
```

2. Use the following command to end the DB2 session:

```
db2 terminate
```

Verifying the deployment of the IBM_CARS_EVENT_DETAIL Java stored procedure

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

Procedure

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

```
db2 connect to eventxml user user using password
db2 "call IBM_CARS_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 show the security event details as name-value pairs.
- Specify "xml" to set off special formatting of the data. This step is the equivalent of calling the IBM_CARS_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 "IBM_CARS_EVENT_DETAIL" of type
"PROCEDURE" having compatible arguments was found.
```

2. Use the following command to end the DB2 session:

```
db2 terminate
```

Chapter 7. 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.

See Chapter 14, “Problem determination,” on page 195 for information about troubleshooting problems with the upgrade procedures.

Considerations for upgrading the Common Audit Service audit server

This topic describes considerations for upgrading the Common Audit Service audit server from versions 6.0 or 6.0.1 to version 6.1.

Consider the following points when preparing to upgrade the Common Audit Service audit server:

- 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. See the DB2 documentation for information on how to upgrade DB2 8.2 to DB2 9 Fix pack 2 or later.

Upgrade goals

Learn the Common Audit Service Version 6.1 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

This topic describes the tasks needed *before* upgrading to version 6.1 of Common Audit Service.

Procedure

1. Back up the existing 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.

In addition, 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.

- a. 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
```

- b. 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.

- c. 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
```

- d. 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
```

2. Ensure that all prerequisite software is installed on the system before installing the product. The required software is listed in the Tivoli Federated Identity Manager 6.2.2 Release Information for the product that is using Common Audit Service.
3. Ensure that all conditions are met that are described in “Pre-installation checklist for all platforms” on page 91.
4. 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

You must install Common Audit Service at a location that is different from the location where the existing earlier version of Common Audit Service is installed.

Complete the procedure described in “Interactive installation” on page 92 to install Common Audit Service interactively. You can also complete the procedure described in “Silent installation” on page 96 to use a response file to install Common Audit Service silently in console mode.

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

You can configure Common Audit Service Version 6.1 to use an existing audit database.

About this task

Follow these steps to configure Common Audit Service Version 6.1 to use an existing XML data store (XMLSTORE database). An earlier version of Common

Audit Service uses the existing XML data store (XMLSTORE database):

Procedure

1. Log on to the administrative console of the WebSphere Application Server. Use the WebSphere Application Server where 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** to start the configuration wizard for Common Audit Service.
3. Click **Next** in the Welcome window to continue. The Welcome dialog shows, indicating that Common Audit Service must be configured before the application can be used.
4. 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.
6. Click **Next** to continue.
7. If administrative security is set ON in the target WebSphere Application Server process, enter the WebSphere administrator user name and password in the **Audit Service Management Authentication** window.
8. In the **Configuration target** window, select the configuration target.
9. In the **Audit Database configuration** window, enter the following information:

Database Instance Owner ID

Specify the name of the user who owns 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 owns 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 so, specify the installation root location of the underlying DB2 client in this field.

Audit Database Name

Specify the name of the existing lower-versioned target XMLSTORE database to be upgraded to version 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 blank. This field specifies the cataloged node name of the remote

DB2 server instance that hosts the target XMLSTORE database, as shown in the local DB2 Administration client.

10. 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.

11. 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 shown in the status window.

Results

A status of **SUCCESS** for all server components indicates that you have successfully configured Common Audit Service Version 6.1 to use the existing earlier version of the XMLSTORE database. Additionally, the target earlier-version of XMLSTORE database has been upgraded to version 6.1.

After completing the procedure, complete 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.” Completing the post-upgrade steps causes clients that use the older version of Common Audit Service to start sending events to Common Audit Service Version 6.1. The process also ensures that the upgraded existing XMLSTORE database is not accidentally dropped during uninstallation of the earlier 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

You must complete a series of tasks to complete the Common Audit Service upgrade.

About this task

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 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.

Procedure

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 on 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 continues 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 automatically stops 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.

Results

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

Chapter 8. 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.

Before you begin

You must unconfigure Common Audit Service before you uninstall it.

Procedure

1. Disconnect all applications from the DB2 database used as the XML data store. The following commands show an example of how to restart DB2 and ensure that no applications are connected:
`db2stop force`
`db2start db2`
`db2 list applications`
2. Open a Web browser and set the value of the URL to the administrative console port of the WebSphere Application Server Deployment Manager or stand-alone server that was specified as the target profile during installation (default port value is 9060 or 9043 for a secure console).
Example: `http://websphereserver.ibm.com:9060/ibm/`
3. Log on 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 103. The Welcome dialog opens, indicating that Common Audit Service must be unconfigured before the application can be uninstalled.
4. 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.
6. Click **Next** to continue.
7. In the **WebSphere Security** window, if global security is enabled on the target WebSphere Application Server process, select the **Global Security** check box.
8. Enter the WebSphere Application Server administrator name and password.
9. Click **Next** to continue.
10. 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 shown in the drop-down list. You must select an entry from the list.
11. Click **Next** to continue.
12. In the **Audit Database** window, the configured values for the database instance owner ID, XML datastore name, and TCP/IP service port are shown. 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.

13. Click **Next** to continue.
14. Review the list of options you have selected in the Summary window.
15. Select **Finish** if the options are correct.
If one or more options are incorrect, use **Back** to return to a window and make the appropriate changes.
16. Review the **Common Audit Service Status** window to determine the outcome of the unconfiguration. If the unconfiguration was unsuccessful, the problems should be corrected and the unconfiguration started again from the Welcome window.
17. Click **OK** to return to the Welcome window.

Chapter 9. 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. 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 204 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 undeploy either the audit server or configuration console feature of Common Audit Service, determine if you want to keep, or remove the database that you use as an XML data store.

To undeploy the audit server but maintain the database, run the unconfiguration wizard. Undeploy only Common Audit Service from the WebSphere Application Server profile. Select to keep the database intact.

To completely undeploy Common Audit Service from a system, run the unconfiguration wizard. Then, undeploy the audit server from the WebSphere Application Server profile. Select to remove the database and staging tables.

Note: If the Common Audit Service is not fully unconfigured before starting uninstallation, a warning message shows in the uninstallation window. The warning message prompts you to completely unconfigure the Common Audit Service components before you uninstall the Common Audit Service. If you continue with the uninstallation, you must 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 204.

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. See “Uninstallation checklist for all platforms” on page 131 for more information about 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 starts.

-options-record *response_file*

Generate a response file by using the options you choose on each panel and write it to the specified file. After you run this interactive uninstallation, you can then use this response file to run a silent uninstallation. It contains all of the appropriate parameters and values.

-is:javahome *java_home*

Specify the home directory of the Java Virtual Machine that the uninstallation launcher uses.

Sample

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

```
uninstall.exe -console
```

Interactive uninstallation using the GUI windows

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

Before you begin

See “Starting the uninstallation wizard” on page 132 for the command you enter to begin the uninstallation wizard.

Procedure

1. Select the language that you want to use for the installation.
2. Click **OK**. The Welcome dialog opens.
3. Click **Next** to continue.
4. In the **Features** window, select both features to uninstall the audit server, configuration console, and configuration utilities.
 - a. Select **Common Audit Service** to uninstall the audit server and configuration utilities only.
 - b. Select **Common Audit Service Configuration Console** to uninstall only the configuration console.
5. Click **Next** to continue.
6. If WebSphere Application Server global security is set, you are prompted to enter the WebSphere Application Server administrator ID and password in the WebSphere Application Server Security Details window.
7. Click **Next** to continue.
8. In the **Summary** window, check that the location of the selected features for uninstallation are correct.
9. (Optional) Click **Back** if you need to change a setting.
10. Click **Next** to begin the uninstallation.
11. 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

You must 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.

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 10. 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, see the topic on *Securing C client events* in the WebSphere Application Server Information Center.

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

This topic describes the server configuration tasks to use the client.

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

This topic describes how to configure the WebSphere Application Server security.

Procedure

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 138 for instructions.
 - LDAP. See “Configuring the LDAP registry” on page 139 for instructions.
 - Custom. See “Configuring a custom registry” on page 140 for instructions.
 - Federated. See “Configuring a federated registry” on page 141 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**.

Note: If you are in a WebSphere Application Server Network Deployment environment, select **Synchronize changes with Nodes** before saving the changes.

- f. Save 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**.
6. Save the changes.

Configuring the operating system registry:

Maintain the consistency of the registry between nodes in cluster by either using an LDAP registry, or a federated repository that includes an LDAP registry.

Before you begin

Note: You must use an LDAP registry under the following circumstances:

- If the Network Deployment cell is not located on a single machine, or
- If WebSphere Application Server is runs on UNIX as a non-root user in your computing environment

The Network Deployment cell consists of all of the nodes.

About this task

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

Procedure

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 **Automatically generated server identity** (if you use WebSphere Application Server Version 6.1 or later), or click **Server identity** 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**.
 7. Save the changes.

Configuring the LDAP registry:

Use the Lightweight Directory Access Protocol (LDAP) user registry when users and groups are located in an external LDAP directory.

About this task

In a clustered environment, use an LDAP registry to maintain consistency of the registry between nodes in the cluster.

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

Procedure

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** (if you use WebSphere Application Server Version 6.1 or later), or click **Server identity** and specify the server user ID and password, to access 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).
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 (use this property to make the default, which is **enabled**, active). 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. (Optional) Select the **SSL enabled** option to enable SSL communication between the LDAP server and WebSphere Application Server.
8. 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.

9. Click **OK**.

Note: If you use a WebSphere Application Server Network Deployment environment, select **Synchronize changes with Nodes** before saving the changes.

10. Save the changes.
11. Select **Test connection** to check the validity of the specified information. Otherwise, the validity of the information is not confirmed until this registry is selected as the current repository.

What to do next

You must configure LTPA as the active authentication protocol to authenticate the users. Then, use an LDAP user registry to enable security in a WebSphere Application Server Network Deployment environment. See “Configuring the LTPA authentication mechanism” on page 143 for instructions.

Configuring a custom registry:

A custom registry is any registry that implements the `com.ibm.websphere.security.UserRegistry` interface.

About this task

From the WebSphere Application Server administrative console, configure the custom registry settings.

Procedure

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** (if you use WebSphere Application Server Server Version 6.1 or later), 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**.

Note: If you are in a WebSphere Application Server Network Deployment environment, select **Synchronize changes with Nodes** before saving the changes.

7. Save your changes.

Configuring a federated registry:

You can use a federated registry to consolidate multiple repositories into a single virtual registry.

About this task

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

Procedure

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 applies 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** (if you use WebSphere Application Server Version 6.1 or later), 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 and any external LDAP registries.
8. Click **OK**.

Note: If you are in a WebSphere Application Server Network Deployment environment, select **Synchronize changes with Nodes** before saving the changes.

9. Save your changes.

Configuring the security policy for the JDBC provider:

Use the wsadmin tool to grant permissions to the JAR file.

About this task

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.

Procedure

1. Start the wsadmin tool of the target profile for which you want to configure a Common Audit Service server instance.
2. Extract the policy file to a temporary location 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:

This topic describes how to edit the app.policy file using the Policy Tool.

About this task

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.

Procedure

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 shown.
2. Click **OK** to dismiss the error.
3. Click **File > Open**.
4. Navigate the directory tree in the Open window to the temporary file *temp_file/app.policy*.
5. Select the file, and click **Open**. The existing code base entries are listed in the window.
6. Click **Add Policy Entry** to create a new code base entry.

7. 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*.
8. Click **Add Permission** to add the permission for the JDBC driver.
9. In the permissions window, select the **AllPermission** entry in the drop-down list.
10. Click **OK**.
11. In the Policy entry window, the string permission `java.security.AllPermission` is shown beneath the **Permission** buttons.
12. Click **Done**.
13. Click **File > Save** to save the updated file.
14. Click **File > Exit** to exit the tool.

Configuring the LTPA authentication mechanism:

From the WebSphere Application Server administrative console, configure Lightweight Third-Party Authentication (LTPA) token authentication.

Procedure

1. Click **Security > Secure administration, applications, and infrastructure > Authentication mechanisms and expiration > Key set groups**.
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**.

Note: If you are in a WebSphere Application Server Network Deployment environment, select **Synchronize changes with Nodes** before saving the changes.

5. Save the changes.
6. Click **Secure administration, applications, and infrastructure**.
7. Click **OK**.

Note: If you are in a WebSphere Application Server Network Deployment environment, select **Synchronize changes with Nodes** before saving the changes.

8. Save the changes.

Restarting the cluster:

This topic describes how to restart the cluster.

About this task

Use the WebSphere Application Server administrative console to restart the cluster to enable the global security method.

Procedure

1. From the WebSphere Application Server administrative console, expand **Servers**.
2. Click **Clusters**, and select the target cluster.
3. Click **Stop**.
4. Stop and restart the deployment manager system with the WebSphere Application Server security enabled method.
5. Stop and restart the agent on the managed nodes with the WebSphere Application Server security enabled method.
6. Expand **Servers**.
7. Click **Clusters**, and select the target cluster.
8. Click **Start**.

What to do next

From this point on, 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.

There 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:

This topic describes how to configure WebSphere Application Server for SSL.

About this task

Use the WebSphere Application Server Administrative Console to configure WebSphere Application Server Secure for Secure Sockets Layer (SSL) authentication.

Procedure

1. Create an SSL configuration entry:
 - a. Click **Security > SSL certificate and key management**.
 - b. Click **SSL Configuration** from the Related Items list.
 - c. Click **New** to create an SSL configuration specifically for Common 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**.

Note: If you are in a WebSphere Application Server Network Deployment environment, select **Synchronize changes with Nodes** before saving the changes.

- k. Save the changes.
- 2. 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 **CARSSLConfiguration** 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**.
 - h. Save the changes.
 - i. Click **Security > SSL certificate and key management**.
 - j. Select to dynamically update the run time when SSL configuration changes occur.
 - k. Click **Apply**.

Note: If you are in a WebSphere Application Server Network Deployment environment, select **Synchronize changes with Nodes** before saving the changes.

- l. Save the changes.

Configuring the Web server plug-in for SSL:

This topic describes how to set SSL security for communication between the Web server and a WebSphere Application Server Web server plug-in. 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.

About this task

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 108 for information on configuring the Web server plug-in.

Procedure

1. From the WebSphere Application Server Administrative Console, click **Servers > Web servers**.
2. Select the Web server name.

3. Click **Plug-in properties**.
4. Under **Repository copy of Web server plug-in files**, specify the keystore 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**.
10. Confirm that at least one personal certificate is in the keystore.
11. 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.

Before you begin

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. See the appropriate Web server documentation for details on how to enable SSL.

Procedure

1. Use the IBM HTTP Server **IKEYMAN** utility to create a CMS key database file and insert the 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**.
- e. Click **Save**. When you click **Save**, the information is written to the security.xml file and the Web server plug-in. For example, /opt/IBM/WebSphere/Plugins/config/webserver1_hostname/plugin-cfg.xml is automatically updated.

5. Enable security on your installed Web server.

- a. Click **Servers > Web servers > your_web_server > Global directives**.
 - b. Under **Global Directives** specify the following information:
 - Select **Security enabled**.
 - Enter CARSServerKey in **Key store certificate alias**.
 - Enter *:443 in **Listen ports**.
 - c. Click **Apply**.
 - d. Click **Save** to enable port 443 for listening on the Web server.
6. Stop the IBM HTTP Server and IBM HTTP Administrative Server.
7. Restart the IBM HTTP Server and IBM HTTP Administrative Server.

8. Stop Application Server. In a clustered environment, stop the cluster.
9. Restart WebSphere Application Server. In a clustered environment, 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.

About this task

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

Procedure

1. Click **Applications > Enterprise Applications > CommonAuditService > Security role to users/group mappings**.
2. Select the **EventSource** role.
3. Click 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**.

Note: If you are in a WebSphere Application Server Network Deployment environment, select **Synchronize changes with Nodes** before saving the changes.

9. Save the changes

What to do next

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 11. Running the server utilities

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

This topic covers:

- 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.

About this task

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/sql1lib/java/db2java.zip:  
DB2INSTANCE_OWNER/sql1lib/java/db2jcc.jar:  
DB2INSTANCE_OWNER/sql1lib/function:  
DB2INSTANCE_OWNER/sql1lib/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 data staging 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 preceding syntax, 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 156.

- **-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 is 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
  -pruntime "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) might fail. If you need more parallelism, increase the number of workers instead of running another instance of the staging utility.

Return codes

If a fatal error occurs 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. Fix the situation that caused the error, and rerun the staging utility. The return code of the staging utility is 0 on success, and 1 on error. A return code of 0 indicates that the staging utility has completed its work, or the **-help** parameter was specified. A return code of 1 indicates that 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 156.

- **-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 150.

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

The following information 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, duplication 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, and the accepted values. You can specify the parameters either in the command line, in the configuration file, or both.

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 use 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. See the DB2 documentation for 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 uses to connect to the database. This user must be the owner of the database instance where the XML data store is located.

Command (staging and XML data store)

`-dbusername value`

Configuration

`util.db.user=value`

Value Valid user name.

Database host name

Denotes the database server host name where DB2 is running.

Command (staging and XML data store)

`-dbhostname value`

Configuration

`util.db.hostname=value`

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

Database instance owner password

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

Command (staging and XML data store)

-dbpassword *value*

Configuration

util.db.passwd=*value*

Value Correct password for the specified user.

Database name

Denotes the name of the audit database.

Command (staging and XML data store)

-dbname *value*

Configuration

util.db.name=*value*

Value Valid database name. The default is eventxml.

Database port number

Specifies the port number on which the DB2 instance is listening. The Database port number must 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. The XML data store utility uses the driver class name in forming the URL string 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 runs in historical mode. 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 must 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 uses to perform work in parallel. Use the number of processors of the machine containing the database, plus one as the value for best performance.

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 before archiving data from the XML data store tables. The prearchive operation prints out the data needed for archiving, such as the names and the dates of the tables to archive.

Post-archive

Use to remove the data from the inactive XML data store tables.

Clean restore table set

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

Command (XML data store)

-operation *value*

Configuration

Not used.

Value Possible values are:

- prearchive
- postarchive
- cleanrestore

Progress report

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

Note: Progress reports might be less frequent than every *N* security events.

For example, if the event batch size parameter is larger than *N*, progress is reported roughly after every batch. The default behavior is that if the value of the progress parameter is 0, progress is not reported.

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 are 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 is 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 runs 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 12. Tivoli Federated Identity Manager reports

This topic describes the software and tasks required to run the out-of-box Tivoli Federated Identity Manager reports. It also describes tasks that are required to create custom reports by using Tivoli 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. The out-of-box reports 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 the reporting requirements of your organization. For information about the contents of the Administrative Events Report and Single Sign-On Summary Report, see “Tivoli Federated Identity Manager reports” on page 174.

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

- 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 Tivoli Federated Identity Manager reports by using any of the following utilities:

Tivoli Federated Identity Manager command-line interface (CLI)

You can generate out-of-box and custom reports by using CLI commands. Report output can be formatted in PDF or HTML.

Tivoli Federated Identity Manager console

You can generate out-of-box and custom reports by 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 by 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. Users can specify report parameters and other report properties by using the interface.
- Command-line interface for working with objects in the data store and for performing additional administrative functions.

Tivoli Federated Identity Manager Version 6.2.2 provides a *report package* that enables you to use 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 by using Tivoli Common Reporting.

You can create custom reports by using the Business and Intelligence Reporting Tools (BIRT). BIRT is an open source set of runtime report utilities that run in an Eclipse environment. Use the tools to do the following tasks:

- Generate custom reports by using the BIRT runtime report engine.
- Create or modify custom report designs by 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*. See the additional instructions on using BIRT to customize reports in 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 Tivoli 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 the Tivoli Federated Identity Manager audit report data that is captured and managed by using Common Audit Service.

1. Use the Tivoli Federated Identity Manager console to install Tivoli Federated Identity Manager and create a domain.
See the *IBM Tivoli Federated Identity Manager Installation Guide* and the *IBM Tivoli Federated Identity Manager Configuration Guide* for information about installing Tivoli Federated Identity Manager and creating a domain.
2. Install and configure Common Audit Service on a system that is separate from the system on which you install Tivoli Federated Identity Manager. The configuration of Common Audit Service sets up the audit server to receive audit events from Tivoli Federated Identity Manager.
See Chapter 5, "Installing Common Audit Service," on page 87 and Chapter 6, "Configuring the audit server," on page 101 for information about configuring Common Audit Service.
3. Begin emitting audit events to the Common Audit Service by performing the following tasks:
 - a. Enable auditing in Tivoli Federated Identity Manager using the Tivoli Federated Identity Manager console. See "Enabling or disabling auditing" on page 3 for instructions.
 - b. Select **Audit Events** in the portfolio of the Auditing window to show the types of events for which you want to capture audit data. See "Modifying audit events" on page 5 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, and 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, clear 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, first you must 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**.

4. 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 Tivoli Federated Identity Manager reports. You can also stage these events and the other audit event types to create custom reports.

Stage (set up) the Tivoli Federated Identity Manager report data by following these steps:

- 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. See “Federated Identity Manager DDL contents” on page 188 for information about 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. Go to the *FIM_INSTALL_ROOT*/reports/cars-files on the Tivoli Federated Identity Manager host and make a backup copy of the CARSShredder.conf file that is installed with Tivoli Federated Identity Manager. 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 Tivoli Federated Identity Manager host. Tivoli Federated Identity Manager provides this version of the CARSShredder.conf file for the Tivoli Federated Identity Manager reports. However, you must back it up for use in creating custom reports for your environment. For more information about using the CARSShredder.conf file to create custom reports, see “Working with the CARSShredder.conf configuration file” on page 178.

- d. Copy the CARSShredder.conf file that is provided for Tivoli 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
```

See Chapter 11, “Running the server utilities,” on page 149 for more information about 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:

- Tivoli Federated Identity Manager console (see “Running the out-of-box Tivoli Federated Identity Manager reports by using the console” on page 170)
- Tivoli Federated Identity Manager command-line interface (see “Running Tivoli Identity Manager reports by using the command-line interface”)

You can also generate and view the Tivoli Federated Identity Manager reports using Tivoli Common Reporting. See “Importing the Tivoli Federated Identity Manager out-of-box reports into the Tivoli Common Reporting environment” on page 171, and “Running the out-of-box Tivoli Federated Identity Manager reports by using the Tivoli Common Reporting console” on page 172 for information about setting up and using Tivoli Common Reporting to generate and manage report data.

Running Tivoli Identity Manager reports by using the command-line interface

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

About this task

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

Procedure

1. Perform the setup tasks described in “Required reporting setup tasks” on page 164.
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 Tivoli Federated Identity Manager report, go to the next step.
3. Run the `wsadmin` command on the system where the Tivoli Federated Identity Manager runtime is installed. Do so to run the CLI commands for the Tivoli 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).

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}
```

See the *IBM Tivoli Federated Identity Manager Administration Guide* for more information about the runItfimReport command.

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

Listing available reports by using the command-line interface

This topic describes the commands you can use to view a list of available Tivoli 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
```

See the *IBM Tivoli Federated Identity Manager Administration Guide* for more information about 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 Tivoli Federated Identity Manager reports by using the command-line interface.

The following contents are 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 must 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
```

The following information is an edited response file for the Administrative Events Report. 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
```

The following contents are 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 must 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
```

The following information is an edited response file for the Single Sign-On Summary Report. 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
```

See the *IBM Tivoli Federated Identity Manager Administration Guide* for more information about using response files with the report commands.

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

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

About this task

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

Procedure

1. Perform the setup tasks described in “Required reporting setup tasks” on page 164.
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 Tivoli 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 shown.
7. Select a report and specify the report parameters. Click the help icon (?) for information about 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 list of reports that are currently running is shown.

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 by using a PDF reader or browser, depending on the file type of the report.

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

Importing the Tivoli 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 Tivoli Federated Identity Manager by using the Tivoli Common Reporting Console.

Before you begin

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

About this task

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

Procedure

1. Perform the setup tasks described in “Required reporting setup tasks” on page 164.

2. Install Tivoli Common Reporting.

You can install Tivoli Common Reporting interactively, by using a graphical interface or commands, or non-interactively, in silent mode. See the *Tivoli Common Reporting User's Guide* for complete information about installing Tivoli Common Reporting.

3. Start the Tivoli Common Reporting Server.

See the **Starting and stopping the Tivoli Common Reporting server** section in the *Tivoli Common Reporting User's Guide* for information about starting Tivoli Common Reporting.

4. Log on to Tivoli Common Reporting by using the console.

See the **Administering Tivoli Common Reporting** section in the *Tivoli Common Reporting User's Guide* for information about logging on to Tivoli Common Reporting.

5. Import the Tivoli Federated Identity Manager report package into the Tivoli Common Reporting environment.

The report package is labeled `TFIM6.2_Reports_Vn_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 the root of the navigation tree in the Tivoli Common Reporting console and click **Import Report Package**. The Import Report Package dialog opens.

The report package is located on the system where Tivoli 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 shows **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 shown 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 shown.

To use advanced search options, click the **Search** tab. For Tivoli 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 must establish a connection to the database in the following directory.

The drivers are located on the Tivoli Federated Identity Manager runtime system in /opt/IBM/FIM/reports/engine/plugins/
org.eclipse.birt.report.data.oda.jdbc_2.2.1.r22x_v20070919\drivers.
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 shows a list of data source names and types used by the report set.
 - c. Select **Database Data Source** and click **Edit**. The **Edit Data Source** dialog opens.
 - d. 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
- e. Click **Save** to save your changes.

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

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

Before you begin

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.

About this task

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

Procedure

1. Perform the setup tasks described in “Required reporting setup tasks” on page 164.
2. Import the Tivoli Federated Identity Manager report package into Tivoli Common Reporting by using the console.
See “Importing the Tivoli Federated Identity Manager out-of-box reports into the Tivoli Common Reporting environment” on page 171) for details.
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 opens.
- 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. Then, specify the event types you want to view.

See the “Tivoli Federated Identity Manager reports” on page 174 for information about the report parameters and associated event types.

- c. Click **Run** to start the report.

To create a report snapshot:

- a. Right click a report and select **Create Snapshot**. A parameter dialog opens.
- b. Specify the parameter values.
- c. Click **Create**. A dialog opens showing the status of the snapshot creation, start and completion timestamps, and user.
- d. You can right click 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 by using Tivoli Federated Identity Manager report designs in the Tivoli Common Reporting environment.

About this task

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 can store separately the monthly Administrative Events reports.

To create a report from an existing design of a Tivoli 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 entering the parameters again, for example:

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

Procedure

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

```
trcmd -list -designs -user %username% -password %pw%
```
3. Use the `trcmd -defineReport` command to define a new report, 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. 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%
```

Tivoli Federated Identity Manager reports

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

Table 29. 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.

Table 29. Administrative Events report (continued)

Administrative Events 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 are: All Events, Successful Single Sign-on Events, or Failed Single Sign-on Events. The default is All Events. Administrator Specify the Tivoli 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 30. 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 Tivoli Federated Identity Manager to create custom reports.

About this task

You can generate custom reports from predefined reporting tables, and you can also create new custom tables.

Procedure

1. Identify the event types and specific elements of each event type that provides the required information.
2. Examine the contents of the DDL files and the tables that are generated by the DDL files. This task determines if the captured event types and attributes are sufficient to meet your reporting needs.
3. Run the Common Audit Service staging utility against the CARSShredder.conf configuration file. This step stages the event data into the reporting tables from the live database tables that contain the captured event data.

The Common Audit Service staging utility queries a configuration file, CARSShredder.conf, to determine exactly what data to stage.

If you must stage additional data that is not provided in the predefined staging tables, customize the CARSShredder.conf, and create new reporting tables. Use the event data that is staged into the reporting tables to create Custom reports.

You can use any reporting tool that queries data from a DB2 database to create custom reports. However, you can use Common Audit Service and BIRT to design, run, view, and manage custom reports.

To set up the CARSShredder.conf configuration file, follow these steps:

- a. Back up the file.
- b. Use the
`CARSShredder.conf.custom.template`

to build a custom version to replace the default version of the file.

Do not modify the existing default tables to create custom tables. Create new, additional reporting tables to hold data for custom reports. The staging utility stages custom data into these newly defined tables.

What to do next

Stage data for custom reports. 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 177
- “Working with the CARSShredder.conf configuration file” on page 178
- “Creating sample custom reports” on page 184
- “Federated Identity Manager DDL contents” on page 188

Requirements for creating new reporting tables

To create a secondary database table for custom reports, you must modify a copy of the Common Audit Service CARSShredder.conf file and the DDL files.

The CARSShredder.conf file and the DDL files are installed when you install the Common Audit Service component. Tivoli Federated Identity Manager also provides a CARSShredder.conf file and corresponding DDL files that you can back up.

You can also use the files to generate secondary custom report tables specifically for Tivoli Federated Identity Manager security events. *To create custom tables for Tivoli Federated Identity Manager security events, use copies of the files provided with Tivoli Federated Identity Manager.*

See “Required reporting setup tasks” on page 164 for information about the location of the Tivoli Federated Identity Manager CARSShredder.conf file and corresponding DDL files.

See “Federated Identity Manager DDL contents” on page 188 for information about the Tivoli Federated Identity Manager DDL files.

The audit server installation creates a set of default reporting tables so that applications like Tivoli Federated Identity Manager can query it 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 must store a custom subset of the audit event data. To store a custom subset of data, create new secondary tables to hold this data. Columns in the custom tables correspond to attributes (elements) of the audit events. *Do not alter the primary table, cars_t_event, because staging utility functionality is closely tied to this table.*

The only two requirements for creating custom secondary tables are:

- Each secondary table must have a column named cars_seq_number which is defined as a foreign key referring to cars_seq_number in cars_t_event.
- The cars_seq_number column must include the rule 'ON DELETE CASCADE'.

The second requirement is necessary if you plan to use the staging utility to prune the reporting tables. The staging utility prunes events from the cars_t_event table and use DB2 to delete the corresponding events from the secondary tables.

Before you begin the process of generating custom reports, examine the mappings between the attributes (elements) in an event and the target reporting table columns. The following sections describe these data relationships in detail.

Steps to support custom reports

This topic describes the steps to support custom reports.

About this task

You must follow these procedures to support custom reports regardless of the reporting tool you are using to run and view the report data.

Procedure

1. Create a data definition language (DDL) file that creates custom secondary reporting tables in the XML event store database.

Tivoli Federated Identity Manager provides the DDL files required to set up reporting tables for Tivoli Federated Identity Manager events. See “Federated Identity Manager DDL contents” on page 188 for information about these DDL files.

2. Run the DDL file to create the reporting tables by 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. 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 by using the modified CARSShredder.conf file.
7. To test and debug the output:
 - a. Run the staging utility in historical mode to test a small amount of data.
 - b. Verify that event attributes are correctly staged by using SQL language.
 - c. Generate a custom report to verify that the required data is included.

Working with the CARSShredder.conf configuration file

The Common Audit Service staging utility references the CARSShredder.conf file to determine what data to move into the reporting tables.

Purpose of the CARSShredder.conf file

The CARSShredder.conf configuration file specifies the mapping between the attributes of an event in the XML data store, and the reporting table columns. The mapping between the attributes of an event in the XML data store is in Common Base Event XML format.

In the CARSShredder.conf file, specify a list of event types that are recognized by the staging utility. For each event type, 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

1. Specify the version in the CARSShredder.conf file for the staging utility to support keyword mapping.

2. Use the *key_xpath_map_file* mapping file when you specify version 2, `CONFIGURATION_VERSION=2.0`
3. Specify the list of event types that are 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 in the following **Event section descriptors** section, or
 - List each event type right before you specify the mapping of the attributes.
4. Specify the mapping of the attributes for each event type. Doing so determines which attributes are staged into the reporting tables. See Table 31 on page 181 for details.

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 is 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, [runtime], ibm_cba_audit_runtime
IBM_CBA_AUDIT_RUNTIME_KEY, 1.0.1, [runtime_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_runtime], ibm_security_runtime
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_SELF CARE, 1.0.1, [security_selfcare], ibm_security_selfcare
IBM_SECURITY_ATTACK, 1.0.1, [security_attack], ibm_security_attack
```

Table 31. Event stanza format of the XML shredder configuration file

Event stanzas	
<i>table, column, [XPath constant keyword #keyword# #keyword:[arrayindex][arrayindex]#]</i>	
<i>table</i>	Specifies the database table name.
<i>column</i>	Specifies the column in the database table.
<i>XPath locator string</i>	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 <i>CARS_HOME</i>/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 of the following parameters: • Corresponding keyword • XPath locator string
<i>constant</i>	Specifies a constant value that is placed in a column for all events. This variable can be: • A string, such as 'AUDIT_AUTHZ' or an integer or other constant. • Any valid clause that DB2 permits in an SQL INSERT STATEMENT, such as CURRENT TIMESTAMP, CURRENT DATE, and so on.
<i>keyword</i>	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.

Table 31. Event stanza format of the XML shredder configuration file (continued)

Event stanzas	
#keyword#	The staging utility searches for the keyword value in CARS_HOME/server/etc/paths/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/paths directory. The staging utility returns an error and stops processing under either of the following circumstances: • If the properties file cannot be located or opened • If the keyword cannot be located in the properties file If 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: <pre>CommonBaseEvent/extendedDataElements [@name='userInfoList']/children [@name='userInfo'][1]/ children[@name='attributes'] /children[@name='attribute'] [1]/children[@name='name']/values</pre>
#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: <pre>CommonBaseEvent/extendedDataElements[@name='userInfoList'] /children[@name='userInfo'][3]/ children[@name='attributes'] /children[@name='attribute'][2]/children[@name='name'] /values</pre> To stage the value of the name element from the second attribute of the third userInfoList, specify #userInfo.attribute.name:[3][2]#. Note: In XPath, expressions the first element of an array starts with an index value of 1 instead of 0. If the amount of array index specified in the keyword does not match the amount of arrays in the XPath locator string in the mapping file, the staging utility: • Returns an error • Stops processing

Table 31. 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. <pre>[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</pre> 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 globalInstanceId of the event, and populate the EVENT_ID column with it.
cars_t_event, cars_seq_number, #RECORD_ID	Maps the record_id field of the event to the cars_seq_number column.
cars_t_event, time_stamp, #creationTime#	Maps the creationTime of the event 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, 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: The ending number sign helps differentiate the keyword from the reserved keywords, such as `#GLOBAL_ID`.

- You cannot specify a column name of a target table multiple times. The following shredder file is incorrect and results in runtime error:

```
cars_t_event, src_comp,          #sourceComponentId.component#
cars_t_event, src_comp,          #sourceComponentId.subComponent#
```

Using the `CARSShredder.conf.custom.template`

When you install the Common 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 must be staged for each of the event types.

This process is done so that the staging utility can function correctly in incremental, historical, and pruning modes. Use a copy of this file as a starting point, and create 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. The data type of the target column is also not identified. The staging utility, during the initialization phase, determines the column type by inspecting the target tables. Ensure that the attribute selected from the event or the source table is appropriately matched to the targeted table column. If a type mismatch occurs, the staging utility attempts to convert from the source format to the target format.

Creating sample custom reports

A sample custom report lists resources that were accessed and identifies whether access was permitted or denied. If the access was denied, the reason for denial is listed in the report.

Procedure

1. Create a custom.ddl file that is used to create a custom secondary table. The custom.ddl file contains the following entries:

```
create table cars_t_cauthz
(
    cars_seq_number      BIGINT,
    res_name_in_app      VARCHAR(1024),
    res_name_in_plcy     VARCHAR(1024),
    res_type             VARCHAR(1024),
    access_dcn           VARCHAR(1024),
    access_dcn_rsn       VARCHAR(1024),
    action               VARCHAR(1024)

    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. See "Running the staging utility command" on page 150 for details.
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 shows 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.

Before you begin

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

Procedure

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` file 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 by using the following command:
 - a. `db2 connect to database_name user db2username by 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 11, “Running the server utilities,” on page 149.

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

The staging utility uses the modified CARSShredder.conf file.

7. To generate custom 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 Tivoli Federated Identity Manager.

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 Tivoli 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 32. 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 so you can 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. Then, it returns data for a single specified security event as one XML document.

The Java stored procedure accepts the security event 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

Show 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 by 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 are staged to the reporting tables.

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

- For information about archiving data, see “Archiving audit data” on page 193.
- For information about restoring archived data, see “Restoring audit data” on page 194.
- For information about running the staging utility, see “Running the staging utility command” on page 150.

Examples of XML-formatted security event data

This topic provides examples of XML-formatted data that are 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. The second event is a mapping action that is generated when the token is mapped to the STSUniversalUser.

Note: To fit the information about 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"?>
<stsuser:STSUniversalUser xmlns:stsuser="urn:ibm:names:ITFIM:1.0:stsuser">
<stsuser:Principal>
<stsuser:Attribute name="NameQualifier" type="urn:oasis:names:tc: ...
SAML:1.0:assertion"/>
<stsuser:Attribute name="name" type="urn:oasis:names:tc:SAML:1.1: ...
nameid-format:emailAddress">
<stsuser:Value>BigDummy</stsuser:Value>
</stsuser:Attribute>
</stsuser:Principal>
<stsuser:AttributeList>
<stsuser:Attribute name="ssn" type="http://example.com/federation/v1/namevalue">
<stsuser:Value>
<stsuser:Value xmlns:stsuser="urn:ibm:names:ITFIM:1.0:stsuser"> ...
555-55-555</stsuser:Value>
</stsuser:Value>
</stsuser:Attribute>
<stsuser:Attribute name="MinorVersion" type="urn:oasis:names:tc:SAML:1.0:assertion">
<stsuser:Value>1</stsuser:Value>
</stsuser:Attribute>
<stsuser:Attribute name="email" type="http://example.com/federation/v1/email">
<stsuser:Value>elain@hotmail.com</stsuser:Value>
</stsuser:Attribute>
<stsuser:Attribute </values>
</extendedDataElements>
<extendedDataElements name="appliesTo" type="string">
<values>http://appliesto/saml1</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://appliedto/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 13. Archiving and restoring audit data

This topic describes how to archive and restore your Common Audit Service audit data.

A security events continue to be stored into the XML data store, available disk space is eventually 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 depends on a combination of available disk space, event submission rates, and also requirements for how long the data must be readily available for interactive reporting.

Archiving audit data

Security event data must be archived to mass storage media before being purged from the database. Use third party database archive tools with the Common Audit Service XML data store utilities to archive security event data from the XML data store.

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.

The following figure shows 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 151.

Restoring audit data

Archived audit event data can be restored into the XML event restore and overflow restore tables in the XML data store for investigation or reporting purposes.

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.

The following figure shows 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

The following figure shows 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 151.

Chapter 14. Problem determination

This topic provides information for troubleshooting the installation of the Common Audit Service components.

Error messages and descriptions are located in the *IBM Tivoli Federated Identity Manager Error Message Reference*.

The topic provides the following information:

- “Log files”
- “Installation problems” on page 197
- Configuration problems
- Upgrade problems
- Uninstallation problems
- Staging utility problems
- Trace level for XML data store
- Debug trace log

Log files

The Common Audit Service components generate 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 C client
- 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 33. Installation log files

Type	Default message log location
Server installation	Windows: • <code>CARS_HOME\serverInstall.log</code> • <code>CARS_HOME\server\logs\sharedLibCreation.log</code> (only for console feature) Linux or UNIX: • <code>CARS_HOME/serverInstall.log</code> • <code>CARS_HOME/server/logs/sharedLibCreation.log</code> (only for console feature)

Table 33. Installation log files (continued)

Type	Default message log location
Server uninstallation	Windows: <i>CARS_HOME</i> \serverUninstall.log
	Linux or UNIX: <i>CARS_HOME</i> /serverUninstall.log
Note: <i>CARS_HOME</i> 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. The error message is also 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 153 for more information.

When run as a postarchive operation, the *XmlStoreUtility* captures the output from DB2 commands in the *XmlStoreUtilsScript_<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

You can configure the log and trace settings to provide you with the information that you need when determining an error.

Procedure

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 of 3 (`traceLevel=3`) causes the following events to be written in the log:

- Events resulting from error conditions
- Events from all trace points in the code

The 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 file name, errors occur. To ensure uniqueness, the `errorFilePath` and `traceFilePath` must 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 the workaround to 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

The following information shows a 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. See "Pre-configuration checklist for all platforms" on page 101 for additional information about setting up before installation.

Installation does not continue when the target WebSphere Application Server is stopped

This topic describes the problem and the workaround to use when an installation does not continue because the WebSphere Application Server is stopped.

Problem

During installation, Common Audit Service checks if the target WebSphere Application Server is running. If the 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 the workaround to 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

Run the installation program again 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 the workaround to 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 prerequisite WebSphere Application Server software is not found, the installer returns an error:

Please enter a valid WebSphere profile path

Workaround

Specify a valid WebSphere Application Server profile path.

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
```

Note

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:tsohlaol 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**.
4. 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

This topic describes an error that might occur when you deploy 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 118.

C client cannot communicate with the Common Audit Service server

This topic describes the problem and the workaround to use if the C client is not correctly configured to communicate with the Common Audit Service audit server.

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 following points might be the possible reasons for the failure:

- The target DB2 server instance that is hosting the existing lower-versioned audit database was not started.

- The wrong credentials were specified for the DB2 instance owner in the Audit database window of the Common Audit Service Configuration Console during the upgrade.
- The target lower-versioned database became corrupted and is not a valid XML data store database.
- The 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.
- The 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 these reasons for failure as a checklist to 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 the Common Audit Service uninstallation problems that you might encounter, and provides workarounds to help resolve the problems.

Uninstall.bin not available

Run the `java -cp uninstall.jar run` command if the `uninstall.bin` does not exist.

When you try 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.

Run the following command, instead of using the installation wizard:

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

CarsConfigUtil.jar is not removed during a successful uninstallation of Common Audit Service

This topic refers to a successful uninstallation of Common Audit Service but without removing the `CarsConfigUtil.jar` file.

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.

About this task

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

Procedure

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.
- c. 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**.
- c. 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**.
 - c. 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**.
 - d. 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**.
 - d. 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.

About this task

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.

Procedure

1. Remove the deployed applications from the 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.

To uninstall applications from the cluster:

- 1) Select the application you want to uninstall.
 - 2) 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 automatically removes 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**. 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.
7. Stop the Deployment Manager.
8. Restart the cluster to start all managed node server instances. If any of the managed node server instances do not start, in the WebSphere Application Server Administrative Console of the network Deployment Manager:
 - a. Click **System administration** > **Nodes**.
 - b. Select the node on which the server instance did not start.
 - c. Click **Full Resynchronize** on the top menu.

Results

After 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

This topic describes message 0000004a.

The following message that is generated by the Web service and emitter can be ignored:

```
0000004a WSDDJAXRPCChan W
com.ibm.ws.webservices.engine.deployment.wsdd.WSDDJAXRPCHandlerInfoChain
getHandlerChain WWS3389E:
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

This topic describes an exception that 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.NullPointerException 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, run the staging utility again 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. See the staging utility error log to identify the SQL exception. Errors are frequently caused by a `CARSShredder.conf` file referring to a nonexistent 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 `ArrayIndexOutOfBoundsException` 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 DSR8203I:
Database product name : DB2/LINUXZ64
[11/14/07 18:00:34:113 EST] 00000026 InternalDB2Un I DSR8204I:
Database product version : SQL09012
[11/14/07 18:00:34:115 EST] 00000026 InternalDB2Un I DSR8205I: JDBC driver name :
IBM DB2 JDBC Universal Driver Architecture
[11/14/07 18:00:34:118 EST] 00000026 InternalDB2Un I DSR8206I:
JDBC driver version : 3.3.54
[11/14/07 18:00:34:121 EST] 00000026 InternalDB2Un I DSR8212I: DataStoreHelper
name is: com.ibm.websphere.rsadapter.DB2UniversalDataStoreHelper@59b759b7.
[11/14/07 18:00:34:127 EST] 00000026 WSRdbDataSour I DSR8208I: 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.xmlstores.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.xmlstores.XmlStoreCustomProperties.setXmlStore
CustomProperties(XmlStoreCustomProperties.java:111)
        at com.ibm.cars.xmlstore.xmlstores.XmlStoreCustomProperties.<init>(XmlS
toreCustomProperties.java:80)
        at com.ibm.cars.xmlstore.xmlstores.XmlStoreEjb.initializeXmlStore(XmlSt
oreEjb.java:287)
        at com.ibm.cars.xmlstore.xmlstores.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 give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law :

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to

IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

If you are viewing this information in softcopy form, the photographs and color illustrations might not be displayed.

Trademarks

IBM, the IBM logo, and ibm.com[®] are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and trademark information; at www.ibm.com/legal/copytrade.shtml.

Adobe, Acrobat, PostScript and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.



Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Index

Numerics

0000004a error message 208

A

- accessibility ix
- archive
 - audit data 193
- archived data
 - creating reports from 189
- archiving samples
 - prearchive output 193
- attributes
 - staging 178
- audit
 - disabling 3
 - enabling 3
 - overview 1
- audit configuration event 64
- audit configuration management
 - elements for 64
- audit data
 - archiving 193
 - collecting 85
 - restoring 194
- audit details report
 - deploying Java stored procedure 117
- audit events
 - modifying 5
- audit infrastructure
 - introduction 84
- audit profile client table
 - auditing task 3
- audit provisioning event 67
- audit server
 - checklist
 - configuration 101
 - installation 91
 - configuring 101
 - fixing failed uninstallation 206
 - GUI instructions
 - stand-alone server 129
 - upgrading the server 123
 - installation options 95
 - installing
 - prerequisites 87
 - stand-alone 94
 - panel instructions
 - stand-alone server 102
 - removing 131
 - removing components 206
 - reporting tables 176
 - running utilities 149
 - silent installation 96
 - silent uninstallation 133
 - uninstallation checklist 131
 - uninstalling 131
 - upgrade preparation 122
 - upgrading
 - overview 121

- audit server (*continued*)
 - upgrading (*continued*)
 - procedure 123
 - wizard
 - interactive installation 92
 - interactive uninstallation 132
- audit server utilities
 - using 149
- audit settings
 - description 3

C

- C client
 - communication
 - problem and workaround 203
- cars_t_authz table 176
- cars_t_event table 176
- CARSShredder.conf file
 - purpose 178
- checklist
 - audit server
 - installing 91
 - uninstalling 131
 - pre-configuration
 - all platforms 101
 - pre-installation
 - audit server on Linux or UNIX 91
 - uninstallation
 - audit server 131
- CLASSPATH
 - setting 149
- cleanrestore tables operation
 - archiving 152
- cleanrestore tables set operation 152
- CLI
 - report 167
 - response files 169
- client events
 - securing 137
- client profile
 - activating 3
 - creating 4
 - deleting 6
 - modifying 6
- cluster
 - configuring JDBC resources 105
 - restarting 143
- cluster type
 - heterogeneous 105
 - homogeneous 105
- clustered environment
 - cluster types 105
 - configuring IBM HTTP Server 146
- command line interface (CLI)
 - response files 169
- commands
 - ibmcarsddinst 119
 - installing audit server
 - interactive mode 92

- commands (*continued*)
 - installing audit server (*continued*)
 - silent mode 96
 - running staging utility 150
 - setting up symbolic links for Linux 118
 - staging utility 150
 - uninstalling audit server
 - interactive mode 132
 - silent mode 134
 - XML data store utilities
 - post-upgrade 152
- Common Audit Service
 - archiving data 193
 - configuring 101
 - debug tracing 200
 - description 83
 - installing 87
 - mapping virtual host from 112
 - reports 163
 - restoring data 194
 - securing 137
 - troubleshooting 195
 - unconfiguring 129
 - uninstalling 133
 - upgrading server 121
 - using existing database 123
 - utilities 149
 - verifying settings 113
 - Web server mapping from 112
- Common Audit Service console
 - embedded WebSphere Application Service deployment
 - problems and workarounds 202
- Common Audit Service settings
 - overview 113
- Common Base Event 84
- common elements
 - security events 7
- compress property
 - configuring 106
- compressed form
 - storing events in 106
- configuration console
 - installing 87
 - uninstalling 133
- configuration issues
 - overview 201
 - problems and workarounds
 - C client and server
 - communication 203
 - configuration panels 201
 - embedded WebSphere Application Service 202
 - Java stored procedure on Linux 202
- configuration management
 - elements for 29
- configuration options
 - event server 103

- console
 - Common Audit Service
 - problems and workarounds 202
 - installing 87
 - reports 170
- ContextDataElements element 7
- conventions
 - typeface x
- Custom Audit Details report
 - setting up to run 117
- custom registry
 - configuring 140
- custom reports
 - creating 175
 - sample 185
 - supporting 177
- custom security event details report
 - creating 189
- custom template
 - CARSShredder.conf file 178

D

- data definition language files (DDL) 188
- data definition language script
 - customizing 99
- data source
 - verifying configuration 114
- data store
 - verifying settings 115
- database tools
 - archiving 193
- DB2
 - securing XML data store 149
- DB2 client
 - installing on Linux or UNIX 89
 - installing on Windows 88
- DB2 server
 - prerequisite for audit server 87
- debug trace
 - using 200
- descriptor, event section 178
- directory names, notation x
- Dis.debug flag
 - tracing installation 200
 - tracing uninstallation 200

E

- education
 - See Tivoli technical training
- EJB module
 - configuration 114
- element
 - ContextDataElements 7
 - format 7
 - Outcome 7
 - Situation 7
 - SourceComponentId 7
- emitter log files 196
- enabling language support 97
- encryption
 - elements 27
- encryption event 27
- environment variables, notation x

- error file path
 - setting 197
- error messages
 - 0000004a 208
- errors
 - audit database configuration 201
- event
 - format 7
- event section descriptor 178
- event server
 - configuration options 103
 - installing 87
- event stanza 178
- event types 7
- events
 - archiving 193
 - audit configuration 64
 - audit provisioning 67
 - deleting using staging utility 150
 - encryption 27
 - IBM_SECURITY_AUTHN 10
 - IBM_SECURITY_AUTHN_
 - TERMINATE 13
 - IBM_SECURITY_ENCRYPTION 27
 - IBM_SECURITY_FEDERATION 14
 - IBM_SECURITY_MGMT_
 - PROVISIONING 67
 - IBM_SECURITY_MGMT_AUDIT 64
 - IBM_SECURITY_MGMT_POLICY 29
 - IBM_SECURITY_SELFCARE 77
 - IBM_SECURITY_SIGNING 28
 - IBM_SECURITY_TRUST 18, 21, 23
 - infrastructure 84
 - management 29
 - restoring 194
 - signing 28
 - single logout 13
 - storing uncompressed form 106
 - trust service 23
 - User Self Care 77

F

- failed uninstallation
 - removing audit server
 - components 206
- federated profiles
 - OAuth 1.0 18
 - OAuth 2.0 21
- federated registry
 - configuring 141
- federation events
 - name identifier management 14
- flag
 - Dis.debug 200
- format
 - elements 7
 - events 84

G

- global security
 - configuring 137
- GUI instructions
 - installing stand-alone server 94
 - stand-alone server 129

- GUI instructions (*continued*)
 - upgrading the server 123

H

- heterogeneous cluster
 - configuring 106
 - determining 105
- historical mode
 - staging data in 150
- homogeneous cluster
 - configuring 106
 - determining 105

I

- IBM HTTP Server
 - configuring for SSL 146
 - enabling 111
 - propagating plug-in
 - local 111
 - remote 111
- IBM_SECURITY_AUTHN event
 - description 10
- IBM_SECURITY_AUTHN_ TERMINATE
 - event
 - description 13
- IBM_SECURITY_ENCRYPTION event
 - description 27
- IBM_SECURITY_FEDERATION event
 - description 14
- IBM_SECURITY_MGMT_
 - PROVISIONING event
 - description 67
- IBM_SECURITY_MGMT_AUDIT event
 - description 64
- IBM_SECURITY_MGMT_POLICY event
 - description 29
- IBM_SECURITY_SELFCARE event
 - description 77
- IBM_SECURITY_SIGNING event
 - description 28
- IBM_SECURITY_TRUST event
 - description
 - chains 23
 - OAuth 1.0 18
 - OAuth 2.0 21
- IBMCARS_DD_REPORT Java stored
 - procedure
 - creating report 189
 - verifying 120
- IBMCARS_EVENT_DETAIL Java stored
 - procedure
 - creating report 189
 - verifying 120
- ibmcars.properties file
 - configuring 153
- ibmcarsddinst commands
 - using 119
- incremental mode
 - staging data in 150
- infrastructure
 - audit 84
- installation
 - audit server 87
 - audit server options 95

- installation (*continued*)
 - checklist
 - audit server on Linux or UNIX 91
 - configuration console 87
 - DB2 client on Linux or UNIX 89
 - DB2 client on Windows 88
 - interactive mode
 - overview 92
 - wizard 92
 - language support 98
 - log files 195
 - prerequisite products 87
 - running debug trace 200
 - silent mode 96
 - stand-alone server 94
- installation issues
 - overview 198
 - problems and workarounds
 - DB2 software 198
 - JVM 199
 - missing WebSphere Application Server 200
 - stopping WebSphere Application Server 199
- interactive mode
 - audit server
 - installing 92
 - uninstalling 132
 - stand-alone server
 - GUI instructions 129
 - panel instructions 102
 - upgrading the server 123

J

- Java stored procedure
 - deploying 117
 - IBMCARS_DD_REPORT 189
 - IBMCARS_EVENT_DETAIL 189
 - Linux
 - problems and workarounds 202
 - Linux platform
 - problems and workarounds 202
 - setting up on Linux 118
 - verifying 120
 - IBMCARDS_DD_REPORT 120
 - IBMCARS_EVENT_DETAIL 120
- JDBC driver
 - configuring 141
 - Policy Tool 142
- JDBC provider
 - security policy for
 - configuring 141
 - Policy tool 142
- JDBC resources
 - configuring 105
- jdk_path parameter
 - configuring 118

K

- keyword mapping
 - shredder configuration file 178

L

- language support 97
- LDAP registry
 - configuring 139
- log and trace settings
 - configuring 197
- log files
 - installation 195
 - overview 195
 - server utilities 196
 - WebSphere Application Server 196
- LTPA authentication mechanism
 - configuring 143

M

- mapping
 - attributes 178
 - Web service roles 148

N

- name identifier management
 - elements 14
- notation
 - environment variables x
 - path names x
 - typeface x

O

- operating system registry
 - configuring 138
- operational reports
 - creating 189
 - operational reports 189
- options
 - audit server installation 95
- ordering publications ix
- outcome element 7

P

- packages
 - language 97
- panel instructions
 - stand-alone server 102
- parameters
 - configuring staging utility 157
- path names, notation x
- Policy Tool
 - editing app.policy 142
 - using 142
- post-archive operation
 - archiving 152
- post-upgrade steps 125
- pre-archive operation
 - archiving 152
- pre-configuration
 - audit server 101
- pre-installation checklist
 - audit server on Linux or UNIX 91
- prerequisite products
 - installing for audit server 87

- prerequisite tasks
 - configuring audit server 101
 - installing audit server on Linux or UNIX 91
 - uninstalling audit server 131
- properties file
 - ibmcars.properties 153
- prune mode
 - staging data in 150
- publications
 - ordering ix
 - related viii

R

- registry
 - configuring
 - custom 140
 - federated 141
 - operating system 138
 - LDAP 139
- remote host
 - IBM HTTP Server on 111
- report tables
 - staging data to 150
- reporting setup tasks
 - overview 164
- reporting tables
 - creating 176
 - requirements 176
- reports
 - console 171
 - custom 175, 177
 - custom audit details
 - setting up to run 117
 - custom security event details 189
 - customizing 186
 - DDL files 188
 - description 174
 - designs 173
 - generating tables for 152
 - importing 171
 - overview 163
 - response files 169
- running
 - overview 172
 - using CLI 167
 - using console 170
- sample 185
- scenario 85
- viewing list using CLI 168

- roles
 - Web service
 - mapping 148

S

- samples
 - custom report 185
 - ibmcars.properties file 153
 - installing audit server
 - interactive mode 92
 - silent mode 96
 - running ibmcarsddinst command 119
 - schema of overflow tables 194
 - schema of XML event tables 194

- samples *(continued)*
 - shredder configuration file
 - section 178
 - uninstalling audit server
 - interactive mode 133
 - silent mode 134
 - XML data store utilities 152
- scenarios
 - creating custom report 185
- Secure Sockets Layer (SSL)
 - configuring 144
 - setting for Web server 145
- Web server
 - configuring 146
- WebSphere Application Server
 - configuring 144
- security Web service log files 196
- security, setting up 137
- server
 - checklist
 - configuration 101
 - installation 91
 - uninstallation 131
 - configuration options 103
 - configuring stand-alone 102
 - deploying Java stored procedure 117
 - fixing failed uninstallation 206
 - GUI instructions
 - stand-alone 129
 - upgrading the server 123
 - installation options 95
 - installing stand-alone 94
 - interactive installation 92
 - panel instructions
 - stand-alone 102
 - reporting tables 176
 - unconfiguring stand-alone 129
 - utilities
 - log files 196
 - setting CLASSPATH 149
 - staging 150
 - XML data store utilities 152
- server utility issues
 - java.lang.NullPointerException
 - running staging utility 209
 - overview 209
 - staging utility
 - exception 209
 - java.lang.NullPointerException
 - exception 209
 - using staging utility
 - remote database access
 - failure 210
 - XML data store utilities
 - remote database access
 - failure 210
- shredder configuration file
 - configuring 178
 - template 178
- signing events
 - elements 28
- silent installation
 - missing prerequisites
 - problems and workarounds 198
- silent mode
 - audit server installation 96
 - response file 96

- silent mode *(continued)*
 - uninstallation
 - audit server 133
- single logout
 - element description 13
- single sign-on
 - elements for 10
- single sign-on event
 - IBM_SECURITY_AUTHN 10
- situation element 7
- SOAP connection
 - Common Audit Service console
 - embedded WebSphere Application Service 202
- software prerequisites
 - installing for audit server 87
- SourceComponentId element 7
- SSL (Secure Sockets Layer)
 - configuring 144
 - setting for Web server 145
- Web server
 - configuring 146
- WebSphere Application Server
 - configuring 144
- staging data
 - custom reports 175
 - description 84
- staging utility
 - configuration parameters 157
 - error 209
 - ibmcars.properties file 153
 - removing events 150
 - restoring audit data 194
 - running 150
 - setting CLASSPATH for 149
 - shredder configuration file 178
 - using 150
- stanza
 - event 178
- stored procedure
 - Java 117
- storing
 - events 84

T

- tables
 - audit profile client 3
 - creating 176
 - restore 152
- templates
 - shredder configuration file 178
- Tivoli Common Reporting
 - importing reports 171
 - running 172
- Tivoli technical training ix
- trace
 - running debug 200
- trace level
 - setting 197
- trace path
 - setting 197
- trace settings
 - configuring 197
- training, Tivoli technical ix
- troubleshooting problems 195
- trust service 23

- trust service *(continued)*
 - elements for 23
- typeface conventions x

U

- uninstallation
 - audit server 131
 - checklist
 - audit server 131
 - interactive 132
 - language support 134
 - removing audit server
 - components 206
 - running debug trace 200
 - silent mode
 - audit server 133
 - starting wizard 132
 - window instructions
 - audit server 133
- uninstallation issues
 - overview 204
 - problems and workarounds
 - CarsConfigUtil.jar 204
 - failed workarounds 205
 - removing audit server
 - components 205
 - uninstall.bin 204
- upgrade
 - audit server 121
 - GUI instructions
 - stand-alone server 123
- upgrade goals 121
- upgrade issues
 - overview 203
- User Self Care
 - elements for 77
- User Self Care event 77
- utilities
 - running audit server 149
 - server
 - setting CLASSPATH 149
 - staging 150
 - XML data store 152

V

- variables, notation for x
- virtual host
 - mapping Common Audit Service to 112

W

- Web server
 - cluster node 107
 - configuring
 - clustered environment 107
 - installing
 - node inside a cluster 108
 - node outside of cluster 107
 - mapping Common Audit Service to 112
 - securing remote communication 145
- Web server mapping
 - Common Audit Service to 112

- Web server SSL
 - configuring 146
- Web service
 - mapping roles 148
- Web service and emitter issues
 - overview 208
 - problems and workarounds
 - 0000004a error message 208
 - web service emitter log 209
- Web service client events
 - configuring server 137
 - securing 137
- Web service component
 - verifying settings 116
- Web service emitter log
 - problems and workarounds 209
- WebSphere Application Server
 - cluster types 105
 - configuring
 - custom registry 140
 - federated registry 141
 - operating system registry 138
 - configuring IBM HTTP Server 146
 - configuring JDBC driver 141
 - configuring LDAP registry 139
 - global security 137
 - mapping Web service roles 148
 - Policy Tool 142
 - prerequisite for audit server 87
 - SSL
 - configuring 144
 - stand-alone configuration 102
 - stand-alone unconfiguration 129
 - upgrading the server 123
- WebSphere Application Server issues
 - out of memory error 212
 - overview 211
 - sending events to server
 - problems and workarounds 211
- WebSphere Application Server node host
 - IBM HTTP Server in 111
- WebSphere Application Server SSL
 - configuring 144
- wsadmin command 141
- XML store utilities (*continued*)
 - setting CLASSPATH for 149
- XML store utilities parameters
 - configuring 157
- XML-formatted security event data
 - examples 190
- XPath statements
 - shredder configuration file 178

X

- XML data source 178
- XML data store
 - archiving 193
 - configuring the compress property
 - for 106
 - description 84
 - securing 149
 - staging data from 150
 - upgrading 121
- XML data store utilities
 - configuring ibmcars.properties
 - file 153
 - running 152
 - using 152
- XML data store utilities configuration
 - updating 153
- XML event store
 - restoring 194
- XML store utilities
 - configuration parameters 157



Printed in USA

GC32-2287-04

