Sun Java™ System

Identity Manager 6.0 Audit Logging

2005Q4M3

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819-4483-10

Copyright © 2006 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

Use is subject to license terms.

This distribution may include materials developed by third parties.

Sun, Sun Microsystems, the Sun logo, Java, SunTone, The Network is the Computer, We're the dot in .com and iForce are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

This product is covered and controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

Waveset, Waveset Lighthouse, and the Waveset logo are trademarks of Waveset Technologies, a Wholly-Owned Subsidiary of Sun Microsystems, Inc..

Copyright © 2000 The Apache Software Foundation. All rights reserved.

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

Copyright © 2003 AppGate Network Security AB. All rights reserved.

Copyright © 1995-2001 The Cryptix Foundation Limited. All rights reserved.

Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
THIS SOFTWARE IS PROVIDED BY THE CRYPTIX FOUNDATION LIMITED AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE CRYPTIX FOUNDATION LIMITED OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Third party trademarks, trade names, product names, and logos contained in this document may be the trademarks or registered trademarks of their respective owners.

Contents

Preface

This guide provides detailed information about the Sun Java™ System Identity Manager auditing and centralized logging systems.

How to Find Information in this Guide

The guide is organized into these sections:

- *Identity Manager Auditing* — Describes how the Identity Manager auditing system records events.
- *Audit Log Database Schema* — Provides tables that list the audit data schema values for supported database types.

Related Documentation and Help

Sun Microsystems provides additional printed and online documentation and information to help you install, use, and configure Identity Manager:

- *Identity Manager Installation*

  Step-by-step instructions and reference information to help you install and configure Identity Manager and associated software.
- *Identity Manager Upgrade*

  Step-by-step instructions and reference information to help you upgrade and configure Identity Manager and associated software.
- *Identity Manager Administration*

  Procedures, tutorials, and examples that describe how to use Identity Manager to provide secure user access to your enterprise information systems.
- *Identity Manager Technical Deployment Overview*

  Conceptual overview of the Identity Manager product (including object architectures) with an introduction to basic product components.
- *Identity Manager Workflows, Forms, and Views*

  Reference and procedural information that describes how to use the Identity Manager workflows, forms, and views — including information about the tools you need to customize these objects.
- *Identity Manager Resources Reference*

  Reference and procedural information that describes how to load and synchronize account information from a resource into Sun Java™ System Identity Manager.

- *Identity Manager Deployment Tools*

  Reference and procedural information that describes how to use different Identity Manager deployment tools; including rules and rules libraries, common tasks and processes, dictionary support, and the SOAP-based Web service interface provided by the Identity Manager server.

- *Identity Manager Tuning, Troubleshooting, and Error Messages*

  Reference and procedural information that provides guidance for tuning Sun Java™ System Identity Manager, provide instructions for tracing and troubleshooting problems, and describe the error messages and exceptions you might encounter as you work with the product.

- *Identity Manager Help*

  Online guidance and information that offer complete procedural, reference, and terminology information about Identity Manager. You can access Help by clicking the Help link from the Identity Manager menu bar. Guidance (field-specific information) is available on key fields.

Product Support

If you have problems with Identity Manager, contact customer support using one of the following mechanisms:

- The online support web site at http://www.sun.com/service/online/us
- The telephone dispatch number associated with your maintenance contract

We'd Like to Hear from You!

We would like to know what you think of this guide and other documentation provided with Identity Manager. If you have feedback — positive or negative — about your experiences using this product and documentation, please send us a note:

Sun Microsystems, Inc.
5300 Riata Park Court
Austin, TX 78727
Attn: Identity Manager Information Development

Email: idm-idd@sun.com

Draft

Draft

1 Identity Manager Auditing

This chapter describes how the Sun Java™ System Identity Manager auditing system records events. The information is organized as follows:

- *Overview*
- *What Does Identity Manager Audit?*
- *Creating Events*
- *Audit Configuration*
- *Database Schema*
- *Log Database Keys*
- *Preventing Audit Log Tampering*

Overview

The purpose of Identity Manager auditing is to record who did what, when, and to which Identity Manager objects. The interface has three main components:

- **Logger –** Events move from the logger to the handler.
- **Handler –** Calls the filter to prevent unwanted events from being written to the audit log.
- **Filter –** If the event passes the filter class, the handler then logs the record to the database using the repository interface.

Note For detailed information about using the auditing system user interface, see *Sun Java™ System Identity Manager Administration*.

What Does Identity Manager Audit?

Most default auditing is carried out by internal Identity Manager components. However, there are interfaces that allow events to be generated from workflow or from Java code.

The default Identity Manager audit instrumentation focuses on four main areas:

- **Provisioner** – An internal component known as the provisioner may generate audit events.
- **View Handlers** – In the view architecture, the view handler needs to generate audit records. A view handler should always audit when objects are created or modified.

- **Session** – The session methods (such as `checkinObject`, `createObject`, `runTask`, `login`, and `logout`) create an audit record after completing an auditable operation. Most of the instrumentation is pushed into the view handlers.
- **Workflow** – By default, only the approval workflows are instrumented to generate audit records. These generate an audit event when requests are approved or rejected. The workflow feature's interface to the audit logger is through the com.waveset.session.WorkflowServices application.

Creating Events

Although Identity Manager handles internal auditing, in some cases you may want to log audit events from custom workflows.

Auditing from Workflow

Use the com.waveset.session.WorkflowServices application to generate audit events from any workflow process. The following table describes the arguments available for this application.

Argument	Type	Description
`op`	String	Operation for WorkflowServices. Must be set to audit.
`type`	String	Name of the object type that is being audited.
`action`	String	Name of the action that was performed.
`status`	String	Name of the status for the specified action.
`name`	String	Name of the object being affected by the specified action.
`resource`	String	(Optional) Name of the resource where the object being changed resides.
`accountId`	String	(Optional) Account ID that is being modified. This should be a native resource account name.
`error`	String	(Optional) Localized error string to accompany any failures.
`reason`	String	(Optional) Name of the ReasonDenied object, which maps to an internationalized message describing the causes of common failures.

Argument	Type	Description
attributes	Map	(Optional) Map of attribute names and values that were added or modified.
parameters	Map	(Optional) Maps up to five additional names or values that are relevant to an event.
organizations	List	List of organization names or IDs where this event will be placed. This is used for organizational scoping of the audit log. If not present, the handler will attempt to resolve the organization based on the type and name. If the organization cannot be resolved, the event is placed in Top (the highest level of the organizations hierarchy).
originalAttributes	Map	(Optional) Map of old attribute values. The names should match the ones listed in the attributes argument. The values will be any previous value you wish to save in your audit log.

Note Refer to the Object Types, Actions, and Results table on page 1-15 for default object, action, and status names.

Examples

The following example illustrates a simple workflow activity. It shows the generation of an event that will log a resource deletion activity named ADSIResource1, performed by ResourceAdministrator:

```
<Activity name='createEvent'>
   <Action class='com.waveset.session.WorkflowServices'>
   <Argument name='op' value='audit'/>
   <Argument name='type' value='Resource'/>
   <Argument name='action' value='Delete'/>
   <Argument name='status' value='Success'/>
   <Argument name='subject' value='ResourceAdministrator'/>
   <Argument name='name' value='ADSIResource1'/>
   </Action>
   <Transition to='end'/>
</Activity>
```

The next example shows how you can add specific attributes to a workflow that tracks the changes applied by each user in an approval process to a granular level. This addition typically will follow a ManualAction that solicits input from a user.

ACTUAL_APPROVER is set in the form and in the workflow (if approving from the approvals table) based on the person who actually performed the approval. APPROVER identifies the person to whom it was assigned.

```
<Action name='Audit the Approval'
   application='com.waveset.session.WorkflowServices'>
     <Argument name='op' value='audit'/>
     <Argument name='type' value='User'/>
     <Argument name='name' value='$(CUSTOM_DESCRIPTION)'/>
     <Argument name='action' value='approve'/>
     <Argument name='accountId' value='$(accountId)'/>
     <Argument name='status' value='success'/>
     <Argument name='resource' value='$(RESOURCE_IF_APPLICABLE)'/>
     <Argument name='loginApplication' value='$(loginApplication)'/>
     <Argument name='attributes'>
       <map>
          <s>fullname</s><ref>user.accounts[Lighthouse].fullname</ref>
          <s>jobTitle</s><ref>user.accounts[Lighthouse].jobTitle</ref>
          <s>location</s><ref>user.accounts[Lighthouse].location</ref>
          <s>team</s><ref>user.waveset.organization</ref>
          <s>agency</s><ref>user.accounts[Lighthouse].agency</ref>
       </map>
    </Argument>
    <Argument name='originalAttributes'>
      <map>
<s>fullname</s>
        <s>User's previous fullname</s>
        <s>jobTitle</s>
        <s>User's previous job title</s>
        <s>location</s>
        <s>User's previous location</s>
        <s>team</s>
        <s>User's previous team</s>
        <s>agency</s>
        <s>User's previous agency</s>      </map>
    </Argument>
    <Argument name='attributes'>
      <map>
         <s>firstname</s>
         <s>Joe</s>
         <s>lastname</s>
         <s>New</s>
      </map>
    </Argument>
    <Argument name='subject'>
       <or>
          <ref>ACTUAL_APPROVER</ref>
          <ref>APPROVER</ref>
      </or>
    </Argument>
    <Argument name='approver' value='$(APPROVER)'/>
</Action>
```

Audit Configuration

The Audit Configuration (`#ID#Configuration:AuditConfiguration`) object is defined in the `sample/auditconfig.xml` file. This configuration object has an extension that is a generic object. At the top level, it has four attributes that are also generic objects:

- `filterConfiguration`
- `extendedTypes`
- `extendedActions`
- `extendedResults`

filterConfiguration

The `filterConfiguration` attribute lists event groups, which are used to enable one or more events to pass through the event filter. Each group listed in the `filterConfiguration` attribute has the attributes listed in the following table:

Attribute	Type	Description
`groupName`	String	Name of the group.
`displayName`	String	Message catalog key that represents the name of a group.
`enabled`	String	Boolean flag indicating if the entire group is enabled or disabled. This is an optimization for the filtering object.
`enabledEvents`	List	List of generic objects that describe the events a group enables. An event must be listed to enable its logging. Each object listed must have these attributes: • `objectType` (String) – Name the objectType. • `actions` (List) – List of one or more actions. • `results` (List) – List of one or more results.

The following example shows the default resource management group:

```
<Object name='Resource Management'>
  <Attribute name='enabled' value='true'/>
  <Attribute name='displayName'
             value='UI_RESOURCE_MGMT_GROUP_DISPLAYNAME'/>
  <Attribute name='enabledEvents'>
    <List>
      <Object>
        <Attribute name='objectType' value='Resource'/>
        <Attribute name='actions' value='ALL'/>
        <Attribute name='results' value='ALL'/>
      </Object>
      <Object>
        <Attribute name='objectType' value='ResourceObject'/>
        <Attribute name='actions' value='ALL'/>
        <Attribute name='results' value='ALL'/>
      </Object>
    </List>
  </Attribute>
</Object>
```

There are eight default event groups:

- Account Management
- Configuration Management
- Identity Manager Logins/Logoffs
- Password Management
- Resource Management
- Role Management
- Security Management
- Task Management
- Changes Outside Identity Manager

You can configure each group from the Audit Events page of the Identity Manager administrative interface (`configure/auditeventconfig.jsp`). The page allows you to configure successful or failed events for each group. The interface does not support adding or modifying enabledEvents for groups, but you can do this by using the Identity Manager debug pages.

The following sections list the default event groups and the events they enable.

Account Management

This group is enabled by default.

Type	Actions
Resource Account	Create, Update, Delete, Enable, Disable, Reject, Approve, Rename
Identity Manager Account	Create, Update, Delete, Enable, Disable, Rename

Configuration Management

This group is enabled by default.

Type	Actions
Configuration	All Actions
UserForm	All Actions
Rule	All Actions
EmailTemplate	All Actions
LoginConfig	All Actions
Policy	All Actions
XMLData	Import
Log	All Actions

Identity Manager Login/Logoff

This group is enabled by default.

Type	Actions
User	Login, Logoff, Credentials Expired
Administrator	Login, Logoff, Credentials Expired

Password Management

This group is enabled by default.

Type	Actions
Resource Account	Change/Reset Password

Resource Management

This group is enabled by default.

Type	Actions
Resource	All Actions
Resource Object	All Actions
ResourceForm	All Actions
ResourceAction	All Actions
AttrParse	All Actions

Role Management

This group is disabled by default.

Type	Actions
Role	All Actions

Security Management

This group is enabled by default.

Type	Actions
ObjectGroup	All Actions
AdminGroup	All Actions
Administrator	All Actions
EncryptionKey	All Actions

Task Management

This group is disabled by default.

Type	Actions
TaskInstance	All Actions
TaskDefinition	All Actions
TaskSchedule	All Actions
TaskResult	All Actions
ProvisioningTask	All Actions

Changes Outside Identity Manager

This group is disabled by default.

Type	Actions
ResourceAccount	NativeChange

extendedTypes

Each new Type that you add to the `com.waveset.object.Type` class can be audited. A new Type must be assigned a unique two-character database key, which is stored in the database. All new Types are added to the various audit reporting interfaces. Each new Type to be logged to the database without being filtered must be added to an audit event groups `enabledEvents` attribute (as described with the `enabledEvents` attribute).

There may be situations in which you want to audit something that does not have an associated `com.waveset.objectType`, or where you want to represent an existing type more granularly.

For example, the `WSUser` object stores all of the user's account information in the repository. Instead of marking each event as a `USER` type, the auditing process splits the `WSUser` object into two different audit types (Resource Account and Identity Manager Account). Splitting the object in this way makes it easier to find specific account information in the audit log.

Add extended audit types by adding to the `extendedObjects` attribute. Each extended object must have the following attributes:

Argument	Type	Description
`name`	String	The name of the type, which is used when constructing AuditEvents and during event filtering.
`displayName`	String	A message catalog key that represents the name of the type.
`logDbKey`	String	Two-character database key to use when storing this object in the Log table. See *Log Database Keys* for reserved values.
`supportedActions`	List	Actions supported by the object type. This attribute will be used when creating audit queries from the user interface. If this value is null, all actions will be displayed as possible values to be queried for this object type.
`mapsToType`	String	(Optional) The name of the `com.waveset.object.Type` that maps to this type, if applicable. This attribute is used when attempting to resolve an object organizational membership if not already specified on the event.
`organizationalMembership`	List	(Optional) A default list of organization IDs where events of this type should be placed, if they do not already have assigned organizational membership.

Note All customer-specific keys should start with the # symbol to prevent duplicate keys when new internal keys are added.

The following example shows the extended type Identity Manager Account.

```
<Object name='LighthouseAccount'>
   <Attribute name='displayName' value='LG_LIGHTHOUSE_ACCOUNT'/>
   <Attribute name='logDbKey' value='LA'/>
   <Attribute name='mapsToType' value='User'/>
   <Attribute name='supportedActions'>
      <List>
         <String>Disable</String>
         <String>Enable</String>
         <String>Create</String>
         <String>Modify</String>
         <String>Delete</String>
         <String>Rename</String>
      </List>
   </Attribute>
</Object>
```

extendedActions

Audit actions typically map to `com.waveset.security.Right` objects. When adding new Right objects, you must specify a unique two-character `logDbKey`, which will be stored in the database. You may encounter situations where there is no right to correspond to a particular action that must be audited. You can extend actions by adding them to the list of objects in the `extendedActions` attribute.

Each object must include the following attributes:

Attribute	Type	Description
`name`	String	The name of the action, which is used when constructing AuditEvents and during event filtering.
`displayName`	String	A message catalog key that represents the name of the action.
`logDbKey`	String	Two-character database key to use when storing this action in the Log table. See *Log Database Keys* for reserved values.

Note All customer-specific keys should start with the # symbol to prevent duplicate keys when new internal keys are added.

This example shows adding an action for Logout.

```
<Object name='Logout'>
  <Attribute name='displayName' value='LG_LOGOUT'/>
  <Attribute name='logDbKey' value='LO'/>
</Object>
```

extendedResults

In addition to extending audit types and actions, you can add results. By default, there are two results: *Success* and *Failure*. You can extend results by adding them to the list of objects in the `extendedResults` attribute.

Each object must include these attributes:

Attribute	Type	Description
`name`	String	The name of the result, which is used when setting the status on AuditEvents and during event filtering.
`displayName`	String	A message catalog key that represents the name of a result.
`logDbKey`	String	One-character database key to use when storing this result in the Log table. See the section titled Database Keys for reserved values.

Note All customer-specific keys should use the range 0–9 to prevent duplicate keys when new internal keys are added.

Database Schema

There are two tables in the Identity Manager database that are used to store audit data:

- **waveset.log** – Stores most of the event details.
- **waveset.logattr** – Stores the IDs of the organizations to which each event belongs.

waveset.log

This section lists the various column names and data types found in the `waveset.log` table. The data types are taken from the Oracle database definition and will vary slightly from database to database. A few of the column values are stored as keys in the database for space optimization. For key definitions, see the section titled *Log Database Keys*.

- **objectType CHAR(2)** – A two-character key that represents the object type that is being audited.
- **action CHAR(2)** – A two-character key that represents the action that was performed.
- **actionStatus CHAR(1)** – A one-character key that represents the result of the action that was performed.
- **reason CHAR(2)** – A two-character database key to describe a `ReasonDenied` object if there was a failure. `ReasonDenied` is a class that wraps a message catalog entry and is used for common failures such as invalid credentials and insufficient privileges.
- **actionDateTime VARCHAR(21)** – The date and time in which the above action took place. This value is stored in GMT time.
- **objectName VARCHAR(128)** – The name of the object that was acted on during an operation.
- **resourceName VARCHAR(128)** – The resource name that was used during an operation, if applicable. Some events do not reference resources; however, in many situations it gives greater detail to log the resource where an operation has performed.
- **accountName VARCHAR(255)** – The account ID being acted on, if applicable.
- **server VARCHAR(128)** – The server where the action was performed (automatically assigned by the event logger).
- **message VARCHAR(255)** – Any localized messages associated with an action including things like error messages. The text is stored localized so it will not be internationalized.
- **interface VARCHAR(50)** – The Identity Manager interface (such as the Administrator, User, IVR, or SOAP interface) from which the operation was performed.

- **acctAttrChanges VARCHAR(4000)** – Stores the account attributes that have changed during a create and update. The attributes changes field is always populated during a create or update for a resource account or Identity Manager account object. All of the attributes changed during an action are stored in this field as a string. The data is in NAME=VALUE NAME2=VALUE2 format. This field can be queried by executing "contains" SQL statements against the name or value.

  The following example shows what a value in the acctAttrChanges column looks like:

  ```
  COMPANY="COMPANY" DEPARTMENT="DEPT" DESCRIPTION="DSMITH
  DESCRIPTION" FAX NUMBER="5122222222" HOME ADDRESS="12282
  MOCKINGBIRD LANE" HOME CITY="AUSTIN" HOME PHONE="5122495555" HOME
  STATE="TX" HOME ZIP="78729" JOB TITLE="DEVELOPER" MOBILE
  PHONE="5125551212" WORK PHONE="5126855555"
  EMAIL="someone@somecompany.COM" EXPIREPASSWORD="TRUE"
  FIRSTNAME="DANIEL" FULLNAME="DANIEL SMITH" LASTNAME="SMITH"
  ```

- **acctAttr01label-acctAttr05label VARCHAR(50)** – These five additional NAME slots are columns that can promote up to five attributes to be stored in their own column instead of in the big blob. You can promote an attribute from the Resource Schema Configuration page using the "audit?" setting, and the attribute will be available for data mining.
- **acctAttr01value-acctAttr05value VARCHAR(128)** – Five additional VALUE slots that can promote up to five attributes to be stored in a separate column instead of in the blob column.
- **parm01label-parm05label VARCHAR(50)** – Five slots used to store parameters associated with an event. Examples of these are Client IP and Session ID.
- **parm01value-parm05value VARCHAR(128)** – Five slots used to store parameters associated with an event. Examples of these are Client IP and Session ID.
- **id VARCHAR(50)** – Unique ID assigned to each record by the repository referenced in the waveset.logattr table.
- **name VARCHAR(128)** – Generated name assigned to each record.

waveset.logattr

The `waveset.logattr` table is used to store IDs of the organizational membership for each event, which is used to scope the audit log by organization.

- **id VARCHAR(50)** – ID of the `waveset.log` record.
- **attrname VARCHAR(50)** – Currently, always `MEMBEROBJECTGROUPS`.
- **attrval VARCHAR(255)** – ID of the `MemberObject` group where the event belongs.

Log Database Keys

The objectType, action, actionStatus, and reason columns are stored in the database as keys to conserve space.

ObjectTypes, Actions, and Results

The following table describes the objectTypes, actions, and results that are stored in the database as keys:

ObjectType Name	DbKey	Action Name	DbKey
Account	AN	Approve	AP
Administrator	AD	Bypass Verify	BV
AdminGroup	AG	Cancel Reconcile	CR
Attribute Definition	AF	Change Password	CP
Application	AP	Create	CT
Capability	US	Connect	CO
Configuration	CN	Delete	DL
Discovery	DS	Deprovision	DP
EmailTemplate	ET	Disable	DS
Extract	ER	Disconnect	DC
ExtractTask	EX	Enable	EN
Identity Manager Account	LA	Execute	LN

ObjectType Name	DbKey	Action Name	DbKey
LoadConfig	LD	Export	EP
LoadTask	LT	Import	IM
LoginConfig	LC	List	LI
Policy	PO	Load	LD
Provisioning Task	PT	Login	LG
Resource	RS	Update	MO
Resource Account	RA	Logout	LO
Resource Form	RF	Native Changes	NC
Resource Object	RE	Provision	PV
RiskReportTask	RR	Reset Password	RP
Role	RL	Reprovision	RV
Rule	RU	Reject	RJ
User	US	Terminate	TR
TaskDefinition	TD	View	VW
TaskInstance	TI	**Results Name**	**DbKey**
TaskSchedule	TS	Success	S
TaskTemplate	TT	Failure	F
TaskResult	TR		
UserForm	UF		
WorkItem	WI		
XMLDATA	XD		

Reasons

The following table describes the reasons that are stored in the database as keys:

Reason Name	English Text	DbKey
PolicyViolation	Violation of policy {0}: {1}	PV
InvalidCredentials	Invalid credentials	CR
InsufficientPrivileges	Insufficient privileges	IP
DatabaseAccessFailed	Database access failed	DA
AccountDisabled	Account disabled	DI

Preventing Audit Log Tampering

You can configure Identity Manager to prevent the following forms of audit log tampering:

- Adding or inserting audit log records
- Modifying existing audit logs records
- Deleting audit log records or the entire audit log
- Truncating audit logs

All Identity Manager audit log records have unique, per-server sequence numbers and encrypted hash of records and sequence numbers. When you create a Tamper Detection Report, it scans the audit logs per server for:

- Gaps in the sequence number (indicating a deleted record)
- Hash mismatches (indicating a modified record)
- Duplicate sequence numbers (indicating a copied record)
- Last sequence number that is less than expected (indicating a truncated log)

To configure tamper-resistant logging:

1. Create a tampering report by selecting **Reports > New > Audit Log Tampering Report**.
2. When the Define a Tampering Report page displays (as shown in the following figure), enter a title for the report and then **Save** it.

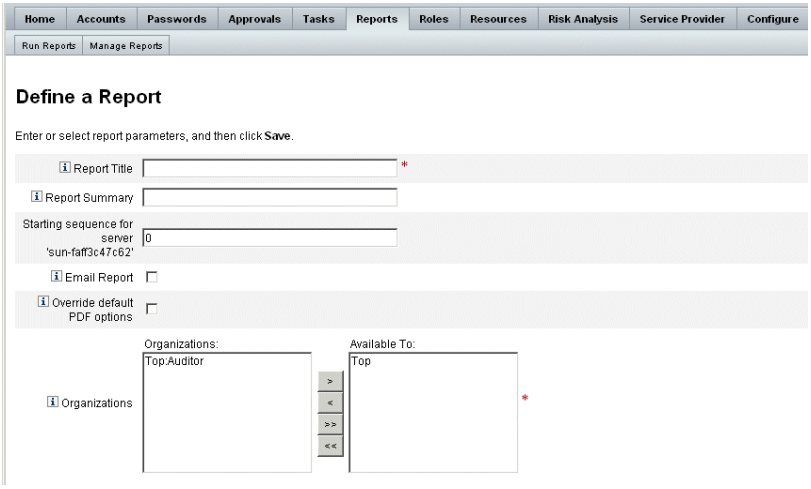

Figure 1. Configuring an Audit Log Tampering Report

Note You can also specify the following optional parameters:

- **Report Summary** – Enter a descriptive summary of the report.
- **Starting sequence for server '<*server_name*>'** – Enter the starting sequence number for the server.

  This option enables you to delete old log entries without having them flagged as tampering and limits the report's scope for performance reasons.
- **Email Report** – Enable to email report results to a specified email address.

  When you select this option, the page refreshes and prompts you for email addresses. However, keep in mind that email is not safe for text content — sensitive information (such as account IDs or account history) may be exposed.
- **Override default PDF options** – Select to override the default PDF options for this report.
- **Organizations** – Select organizations that should have access to this report.

3. Next, select **Configure > Audit Events** to open the Audit Events page (shown in the following figure).

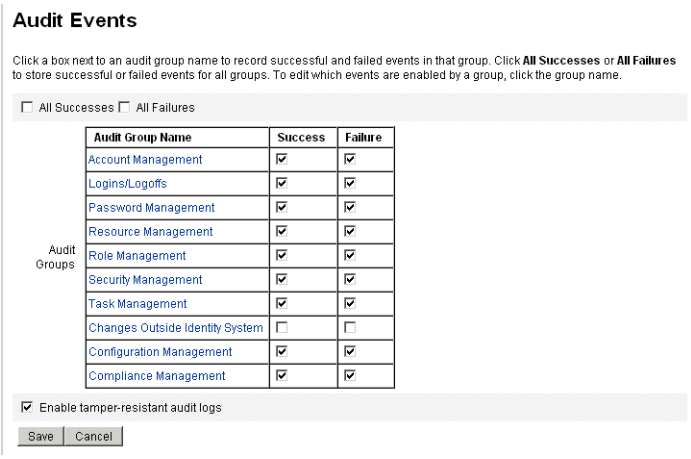

Figure 2. Tamper-Resistant Audit Logs

4. Select the **Enable tamper-resistant audit logs** box, and then click **Save**.

Note You can turn this option off again, but unsigned entries will be flagged as such in the Audit Log Tampering Report, and you must reconfigure the report to ignore these entries.

2 Audit Log Database Schema

The tables provided in this chapter list the audit data schema values for the following database types:

- Oracle
- DB2
- MySQL
- Sybase

Oracle

The following table provides a list of data schema values for the Oracle database type:

Database Column	Value
id	VARCHAR(50) NOT NULL
name	VARCHAR(128) NOT NULL
resourceName	VARCHAR(128)
accountName	VARCHAR(50)
objectType	CHAR(2)
objectName	VARCHAR(128)
action	CHAR(2)
actionDate	CHAR(8)
actionTime	CHAR(12)
acctAttrChanges	VARCHAR(4000)
acctAttr01label	VARCHAR(50)
acctAttr01value	VARCHAR(128)
acctAttr02label	VARCHAR(50)
acctAttr02value	VARCHAR(128)
acctAttr03label	VARCHAR(50)
acctAttr03value	VARCHAR(128)

Database Column	Value
acctAttr04label	VARCHAR(50)
acctAttr04value	VARCHAR(128)
acctAttr05label	VARCHAR(50)
acctAttr05value	VARCHAR(128)
parm01label	VARCHAR(50)
parm01value	VARCHAR(128)
parm02label	VARCHAR(50)
parm02value	VARCHAR(128)
parm03label	VARCHAR(50)
parm03value	VARCHAR(128)
parm04label	VARCHAR(50)
parm04value	VARCHAR(128)
parm05label	VARCHAR(50)
parm05value	VARCHAR(128)

DB2

The following table provides a list of data schema values for the DB2 database type:

Database Column	Value
id	VARCHAR(50) NOT NULL
name	VARCHAR(128) NOT NULL
resourceName	VARCHAR(128)
accountName	VARCHAR(50)
objectType	CHAR(2)
objectName	VARCHAR(128)
action	CHAR(2)
actionDate	CHAR(8)
actionTime	CHAR(12)
actionStatus	CHAR(1)
interface	VARCHAR(50)
server	VARCHAR(128)
subject	VARCHAR(128)
reason	CHAR(2)
message	VARCHAR(255)
acctAttrChanges	CLOB(16M)
acctAttr01label	VARCHAR(50)
acctAttr01value	VARCHAR(128)
acctAttr02label	VARCHAR(50)
acctAttr02value	VARCHAR(128)
acctAttr03label	VARCHAR(50)
acctAttr03value	VARCHAR(128)
acctAttr04label	VARCHAR(50)
acctAttr04value	VARCHAR(128)

Database Column	Value
acctAttr05label	VARCHAR(50)
acctAttr05value	VARCHAR(128)
parm01label	VARCHAR(50)
parm01value	VARCHAR(128)
parm02label	VARCHAR(50)
parm02value	VARCHAR(128)
parm03label	VARCHAR(50)
parm03value	VARCHAR(128)
parm04label	VARCHAR(50)
parm04value	VARCHAR(128)
parm05label	VARCHAR(50)
parm05value	VARCHAR(128)

MySQL

The following table provides a list of data schema values for the MySQL database type:

Database Column	Value
id	VARCHAR(50) BINARY NOT NULL
name	VARCHAR(128) BINARY NOT NULL
resourceName	VARCHAR(128)
accountName	VARCHAR(50)
objectType	CHAR(2)
objectName	VARCHAR(128)
action	CHAR(2)
actionDate	CHAR(8)
actionTime	CHAR(12)

Database Column	Value
actionStatus	CHAR(1)
interface	VARCHAR(50)
server	VARCHAR(128)
subject	VARCHAR(128)
reason	CHAR(2)
message	VARCHAR(255)
acctAttrChanges	BLOB
acctAttr01label	VARCHAR(50)
acctAttr01value	VARCHAR(128)
acctAttr02label	VARCHAR(50)
acctAttr02value	VARCHAR(128)
acctAttr03label	VARCHAR(50)
acctAttr03value	VARCHAR(128)
acctAttr04label	VARCHAR(50)
acctAttr04value	VARCHAR(128)
acctAttr05label	VARCHAR(50)
acctAttr05value	VARCHAR(128)
parm01label	VARCHAR(50)
parm01value	VARCHAR(128)
parm02label	VARCHAR(50)
parm02value	VARCHAR(128)
parm03label	VARCHAR(50)
parm03value	VARCHAR(128)
parm04label	VARCHAR(50)
parm04value	VARCHAR(128)
parm05label	VARCHAR(50)
parm05value	VARCHAR(128)

Sybase

The following table provides a list of data schema values for the Sybase database type:

Database Column	Value
id	VARCHAR(50) NOT NULL
name	VARCHAR(128) NOT NULL
resourceName	VARCHAR(128)
accountName	VARCHAR(50)
objectType	CHAR(2)
objectName	VARCHAR(128)
action	CHAR(2)
actionDate	CHAR(8)
actionTime	CHAR(12)
actionStatus	CHAR(1)
interface	VARCHAR(50)
server	VARCHAR(128)
subject	VARCHAR(128)
reason	CHAR(2)
message	VARCHAR(255)
acctAttrChanges	TEXT
acctAttr01label	VARCHAR(50)
acctAttr01value	VARCHAR(128)
acctAttr02label	VARCHAR(50)
acctAttr02value	VARCHAR(128)
acctAttr03label	VARCHAR(50)
acctAttr03value	VARCHAR(128)
acctAttr04label	VARCHAR(50)

Database Column	Value
acctAttr04value	VARCHAR(128)
acctAttr05label	VARCHAR(50)
acctAttr05value	VARCHAR(128)
parm01label	VARCHAR(50)
parm01value	VARCHAR(128)
parm02label	VARCHAR(50)
parm02value	VARCHAR(128)
parm03label	VARCHAR(50)
parm03value	VARCHAR(128)
parm04label	VARCHAR(50)
parm04value	VARCHAR(128)
parm05label	VARCHAR(50)
parm05value	VARCHAR(128)

Audit Log Database Mappings

The following tables contain the mappings between stored audit log database keys and the display string to which they map in the audit report output. Identity Manager stores items that are used as constants as short database keys to save space in the repository. The product interface does not display these mappings. Instead, you see them only when examining the output of a dump of the audit report results.

The three tables supplied in this section list:

- Object key types
- Action keys
- Action status keys

Audit Object Type	DB Key
Administrator	AD
Admin Group	AG
Application	AP
Audit Config	AC
Audit Log	AL
Email Template	ET
Lighthouse Account	LA
Login Config	LC
Notify	NT
Object Group	OG
Policy	PO
Remedy Config	RC
Resource Account	RA
Resource	RS
Resource Object	RE
Role	RL
Role Attribute	RT
Task Definition	TD
Task Instance	TI
Task Schedule	TS
User	US
Workflow Case	WC
Workflow Process	WP
Workflow Task	WT

Action	DBKey
Approve	AP
Change Password	CP
Change Resource Password	CR
Configure	CG
Connect	CN
Create	CT
Credentials Expired	CE
Delete	DL
Delete Account	DA
Deprovision	DP
Disable	DS
Disconnect	DC
Enable	EN
Launch	LN
Load	LD
Login	LG
Logout	LO
Native Change	NC
Protect Resource Password	PT
Provision	PV
Reject	RJ
Reprovision	RV
Reset Password	RP
Terminate	TR
Update	MO
View	VW

Action Status	DB Key
Failure	F
Success	S

Index

P

R

S

T

V

W