Reporting Guide for Novell Sentinel

Identity Manager 4.0.1

April 15, 2011

Novell

Legal Notices

Novell, Inc., makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc., makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. See the Novell International Trade Services Web page (http://www.novell.com/info/exports/) for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

Novell, Inc.
1800 South Novell Place
Provo, UT 84606
U.S.A.
www.novell.com

Online Documentation: To access the latest online documentation for this and other Novell products, see the Novell Documentation Web page (http://www.novell.com/documentation).

Novell Trademarks

For Novell trademarks, see the Novell Trademark and Service Mark list (http://www.novell.com/company/legal/trademarks/tmlist.html).

Third-Party Materials

All third-party trademarks are the property of their respective owners.

Contents

About This Guide

Welcome to the *Identity Manager Integration Guide for Novell Sentinel*. This guide provides the information necessary to integrate Novell Sentinel with Identity Manager to provide auditing and reporting services.

Audience

This guide is intended for network administrators.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation.

Documentation Updates

For the most recent version of the *Identity Manager Integration Guide for Novell Sentinel*, visit the Identity Manager Documentation Web site (http://www.novell.com/documentation/idm401/).

Additional Documentation

For the current Sentinel documentation, see the Sentinel Documentation Web site (http://www.novell.com/documentation/sentinel61/index.html).

1 Overview

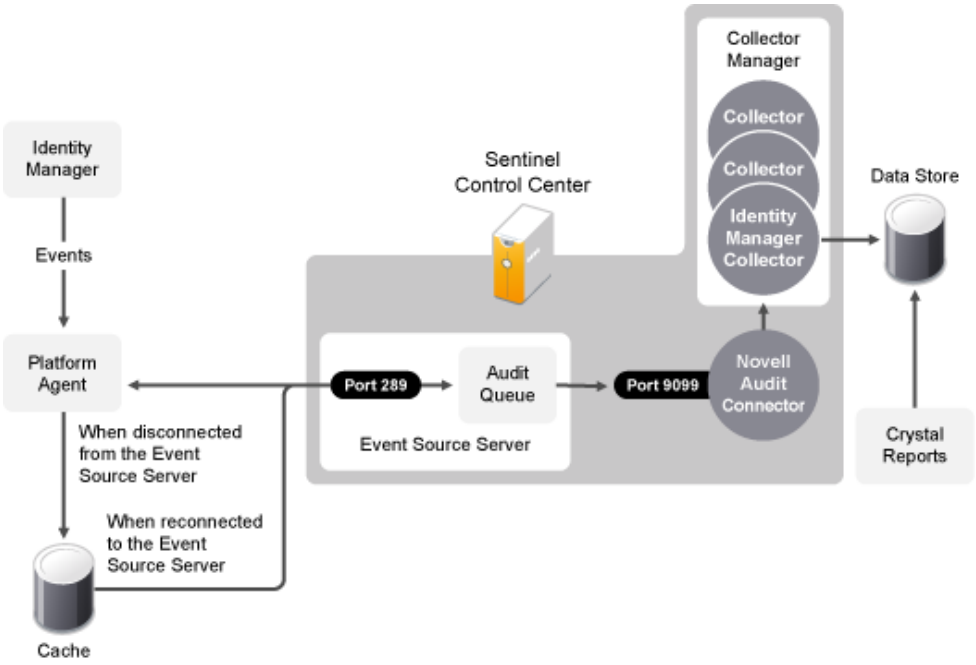
Adding Novell Sentinel to your Identity Manager solution provides a reporting services. By adding reporting, you can demonstrate that the business policies are enforced within your Identity Manager solution. This is the last component to add to your Identity Manager solution.

1.1 Sentinel Integrated Architecture

Sentinel is a security information management and compliance monitoring solution that monitors, responds to, and reports on security and compliance events. Sentinel easily integrates with Novell Identity Manager so you get automated, real-time security management and compliance monitoring across all systems and networks. The Sentinel-Identity Manager framework provides automatic documenting and reporting of security, systems, and access events across the enterprise; built-in incident management and remediation; and the ability to demonstrate and monitor compliance with internal policies and government regulations.

The following diagram illustrates the Identity Manager logging and reporting architecture when integrated with Sentinel.

Figure 1-1 *Identity Manager and Sentinel Integrated Architecture*



1. An Identity Manager event occurs and it is sent to the Platform Agent. To capture all Identity Manager events, the Platform Agent must be installed and configured on each Identity Manager server.

2. (Conditional) If the Platform Agent cannot connect to the Event Source Server, the events are stored in cache until the connection is reestablished.
3. The Platform Agent sends the events to the Event Source Sever, which stores the events in the audit queue.
4. The events in the audit queue are sent to the Novell Audit Connector.
5. The Novell Audit Connector sends the events to the Identity Manager Collector, which parses the information and then stores the parsed events in the data store.
6. The stored events are displayed through Crystal Reports.

For a thorough discussion of the Sentinel architecture, see "Appendix A Sentinel Architecture" in the *Novell Sentinel User's Guide* (http://www.novell.com/documentation/sentinel61/pdfdoc/sentinel_61_user_guide.pdf).

2 Configuring Novell Sentinel with Identity Manager

Use the following checklist to verify that all of the steps are completed to install and configure Sentinel with Identity Manager.

- ❒ Install and configure the Sentinel components. The Sentinel components should be a different server from the Identity Manager server. For more information, see the *Novell Sentinel Installation Guide* (http://www.novell.com/documentation/sentinel61/pdfdoc/sentinel_61_installation_guide.pdf).
- ❒ Install and Configure the Novell Sentinel Identity Manager Collector. For more information, see Chapter 3, "Installing and Configuring the Identity Manager Collector," on page 11.
- ❒ Install and configure the Novell Audit Connector. For more information, see Chapter 4, "Installing and Configuring the Novell Audit Connector," on page 15.
- ❒ Install and configure the Platform Agent.

  The Platform Agent (`logevent`) is the client piece of the Novell auditing architecture.It is automatically installed if either the *Novell Identity Manager Metadirectory Server* or *Novell Identity Manager Connected System* option is selected during the Identity Manager install. It is also installed during the installation of the User Application.

  For more information on installing and configuring the Platform Agent, see Chapter 5, "Installing and Configuring the Platform Agent," on page 19.
- ❒ Install and configure the Novell XDAS.

  The XDAS provides a standardized classification for audit events.

  For more information, see Chapter 6, "Installing and Configuring the Novell XDASv2," on page 23.
- ❒ (Optional) Secure the connection between Identity Manager and the Platform Agent.

  For more information, see Chapter 7, "Securing the Logging System," on page 29.
- ❒ Configure the Sentinel Control Center to access the Crystal Enterprise server for the predefined reports for Identity Manager. For more information, see Chapter 10, "Querying and Reporting," on page 47.

3 Installing and Configuring the Identity Manager Collector

The Identity Manager Collector parses and normalizes the raw data passed to it by the Novell Audit Connector and converts the data into a Sentinel event. The Sentinel event can be visualized in the Active View, processed by the correlation engine, queried in a report, and added to an incident response workflow.

The Identity Manager Collector can also parse non-event data and transform the raw scan data into a format understood by Sentinel. Sentinel then stores the vulnerability data in the database and includes it in the Exploit Detection map. For more detailed information about Sentinel collectors, see the *Sentinel Collector Script User's Guide* (http://www.novell.com/documentation/sentinel6/pdfdoc/sentinel60_collectorguide.pdf).

3.1 Installing the Identity Manager Collector

The Identity Manager Collector must be added to the Event Source Manager to be installed. This step is only done once. The Identity Manager Collector is then displayed as a collector to select during configuration. To install the Identity Manager Collector:

1. Download the Identity Manager Collector (`Novell_Identity-Manager_6.1r3.clz.zip`) from the Sentinel 6.1 Connectors Web site (http://support.novell.com/products/sentinel/secure/sentinel61.html) to the server where the Sentinel Control Center is running.

   The Identity Manager Collector is located under the *Collectors* tab.
2. Log in to the Sentinel Control Center.
3. Select the *Event Source Management* > *Live View*, then select *Tools* > *Import plugin*.
4. Browse to and select the `Novell_Identity-Manager_6.1r3.clz.zip` file, then click *Next*.
5. Follow the remaining prompts, then click *Finish*.
6. Continue with Section 3.2, "Configuring the Identity Manager Collector," on page 11. The Identity Manager Collect must be configured to work.

3.2 Configuring the Identity Manager Collector

1. In the Event Source Management live view, right-click the Collection Manager, then click *Add Collector*.
2. Select *Novell* in the *Vendor* column.
3. Select *Identity Manager* in the *Name* column, then click *Next*.
4. From the *Installed Scripts* column, select *Novell_Identity-Manager_6.1r3*, then click *Next*.
5. Configure the Identity Manager Collector for your needs by using the following information.

Configuration Parameter	Default Value	Description
Alert Unsupported Events	no	Generates an event for the event source data not handled by the Identity Manager Collector.
Default Reporter Name	DEFAULT_RN	Populates the Reporter Name event tag with this text, if not handled in the collector script.
Default Sensor Name	DEFAULT_SN	Populates the Sensor Name event tag with this text, if not handled in the collector script.
Default Severity	3 Medium (3)	The default severity assigned to the events, if the severity mapping is not defined in the collector script.
Event Source Missing Year	yes	Select whether to use the current year if year is not reported in the event source timestamp.
Event Source Time Zone	+0000	Sets the time zone offset UTC (+0000) of the event source data timestamps. This is used if the source data is reported only in local time with no time zone indicated. The format is + or - followed by a two digit hour and minute offset.
Event Source Time uses 24 Hour Clock	yes	Select whether the time reported in the event source data is in the 24 hour format.
Execution Mode	release	Sets the executions mode for the collector. There are three options: ◆ **release:** Use this mode for normal operation. ◆ **custom:** Use this mode if the Identity Manager Collector is customized. ◆ **debug:** Use this mode when troubleshooting issues. It generates debug trace files.
IP To Country Mapping	off	Select whether to determine the source country from the Source IP.
MSSP Customer Name		
Script Error Severity	5 Severe (5)	Sets the severity for a script error event.
Send Script Error Message	yes	Sends a script error event when there is an error with the collector script.
Taxonomy Filename	`tx_novl_idm_3x.csv`	The name of the taxonomy CSV file used by the collector script.

Configuration Parameter	Default Value	Description
Translate IP and hostname	no	Translates the IP address to the hostname and the hostname to the IP address for the source and destination, if it is missing. This parameter uses the packages with the collector. These files must be pre-filled with the host information if name resolution is desired.
Unsupported Events Severity	1 Trivial (1)	Assigned severity for unhandled events generated by the collector script.

6 Click *Next*.

7 Complete the configuration of the Identity Manager Collector with the following information:

- **Name:** Specify a name for this connector.
- **Run:** Select whether the connector is started whenever the Collector Manager is started.
- **Alert if no data received in specified time period:** (Optional) Select this option to send the No Data Alert event to Sentinel if data is not received by the Connector in the specified time period.
- **Limit Data Rate:** (Optional) Select this option to set a maximum limit on the rate of data the connector sends to Sentinel. If the data rate limit is reached, Sentinel throttles back on the source in order to limit the flow of data.
- **Set Filter:** (Optional) Specify a filter on the raw data passing through the connector.
- **Trust Event Source Time:** (Optional) Select this option if you trust the Event Source server's time.

8 Click *Finish*.

The next step is to proceed to Chapter 4, "Installing and Configuring the Novell Audit Connector," on page 15.

4 Installing and Configuring the Novell Audit Connector

The Novell Audit Connector facilitates integration between Identity Manager and Sentinel. Identity Manager is instrumented to send all events to the Platform Agent for logging purposes. The Novell Audit Connector allows Sentinel to connect to Identity Manager via the Platform Agent. For more detailed information about the Novell Audit Connector, see the Novell Audit Connector documentation (http://support.novell.com/products/sentinel/doc/connectors/audit_connector.pdf).

You must have the Identity Manager Collector installed and configured before proceeding with the installation and configuration of the Novell Audit Connector.

4.1 Installing the Novell Audit Connector

1. Download the `audit_connector.zip` file from the Sentinel 6.1 Connectors Web site (http://support.novell.com/products/sentinel/secure/sentinel61.html) to the server where the Sentinel Control Center is running.

   The Novell Audit connector is located under the *Connectors* tab.
2. Log in to the Sentinel Control Center.
3. Select *Event Source Management* > *Live View*, then select *Tools* > *Import plugin*.
4. Select *Import Collector Script or Connector plugin package file (.zip)* option, then click *Next*.
5. Browse to and select the `audit_connector.zip` file, then click *Next*.
6. Follow the remaining prompts, then click *Finish*.
7. Continue with Section 4.2, "Configuring the Novell Audit Connector," on page 15. you must configure the Novell Audit connector for it to work.

4.2 Configuring the Novell Audit Connector

The Novell Audit Connector is configured to receive messages sent from Identity Manager to the Platform Agent. These events are then processed by the Identity Manager Collector.

There are multiple ways to configure the Novell Audit Connector. These instructions use the right-click menu items on the Event Source Management Graph view.

1. Right-click the Identity Manager Collector, then click *Add Connector*.
2. Select *View Compatible Connection Methods Only*.
3. Select *Audit* from the list of installed connectors, then click *Next*.
4. Click *Add* to add an Event Source server.

   The Event Source server is the server that is running the Platform Agent and Identity Manager.

5 Select the network interface setting for the server running the Platform Agent and Identity Manager.

- **All network interfaces:** Binds the port on all the IP addresses of the server, including the loopback address.
- **Internal loopback interface:** Only binds the local loopback address.
- **Network interface with this IP:** Binds the port only to the specified IP address.

6 In the *Port Number* field, specify the SLS port, then click *Next*.

The default port is 289.

7 Select the option for the client authentication type.

- **Open:** Allows all SSL connections from the Platform Agent. It does not perform any client certificate validation or authentication.
- **Loose:** Validates a client certificates to be a valid X.509 certificate, but does not check if the certificate is signed by a Certificate Authority.
- **Normal:** Validates the certificate to be a valid X.509 certificate and also checks to see that the client certificate is signed by a Certificate Authority.

  This option requires a trust store to be imported. The trust store must have the client's certificate and the Certificate Authority's certificate. Click the *Import* button to import the trust store.

8 Select whether you want to use the built-in server key pair or import server key pair, then click *Next*.

The Novell Audit connector comes with a built-in certificate. You can use it or overwrite it with your own certificate.

9 Select the behavior of the Event Source Server if it receives move events than the Collector can parse. The options are:

- **Drop connections:** The Event Source Server drops existing connections and stops accepting new connections until the buffer has space for the new messages. This is the default behavior, because the Platform Agent performs caching when a connection is dropped.
- **Drop messages:** The Event Source Server drops the oldest message in order to accept the new message. These dropped messages are lost and cannot be recovered.

10 Select whether the Event Source Server disconnects an SSL connection with the Platform Agent if the connection is idle and does not send any data within the set number of minutes.

If you select this option, you must specify the number of minutes to wait before it disconnects. The default value is 15 minutes.

11 Select whether you want the Event Source Server to request the Platform Agent to send the signature of the event with the event, then click *Next*.

12 Select *Run* to have the Event Source Server automatically start whenever the Collector Manager is restarted, then click *Finish*.

13 Repeat Step 4 through Step 12 for each Identity Manager server.

To capture all events in your environment, you must have an Event Source server for each Identity Manager, and the Identity Manager server must have the Platform Agent installed on it.

14 Select the Event Source server to add to the Novell Audit Connector, then click *Next*.

15 Use the default policy or create a custom policy to automatically add or exclude individual source devices, then click *Next*.

For more information, see "Auto Configuring Event Sources" in the *Novell Audit Connector Guide* (http://support.novell.com/products/sentinel/doc/connectors/audit_connector.pdf)

16 Finish the configuration of the connector with the following information, then click *Finish*.

- **Name:** Specify a name for this connector.
- **Run:** Select whether the connector is started whenever the Collector Manager is started.
- **Alert if no data received in specified time period:** (Optional) Select this option to send the No Data Alert event to Sentinel if not data is received by the connector in the specified time period.
- **Limit Data Rate:** (Optional) Set a maximum limit on the rate of data the connector sends to Sentinel. If the data rate limit is reached, Sentinel throttles back on the source in order to limit the flow of data.
- **Set Filter:** (Optional) Specify a filter on the raw data passing through the connector.
- **Save Raw Data to a File:** (Optional) Save the raw data passing through this connector to a file for further analysis.

Proceed to Chapter 5, "Installing and Configuring the Platform Agent," on page 19.

5 Installing and Configuring the Platform Agent

The Platform Agent is the client portion of the Sentinel auditing system for Identity Manager. It receives logging information and system requests from Identity Manager and transmits the information to the Novell Audit Connector for Novell Sentinel.

5.1 Installing the Platform Agent

The Platform Agent is automatically installed if either the Novell *Identity Manager Metadirectory Server* or *Novell Identity Manager Connected System* option is selected during the Identity Manager install. For more information on the Identity Manager installation, see the *Identity Manager 4.0.1 Integrated Installation Guide*.

IMPORTANT: The Platform Agent must be installed on every server running Identity Manager if you want to log Identity Manager events.

5.2 Configuring the Platform Agent Text File

After you install Identity Manager, you can configure the Platform Agent. The Platform Agent's configuration settings are stored in a simple, text-based `logevent` configuration file. By default, `logevent` file is located in the following directories:

Table 5-1 *Platform Agent Configuration File*

Operating System	File
Linux	`/etc/logevent.conf`
Solaris	`/etc/logevent.conf`
Windows	`\windows\logevent.cfg`

The following is a sample `logevent` file.

```
LogHost=127.0.0.1
LogCacheDir=c:\logcache
LogCachePort=1288
LogEnginePort=1289
LogCacheUnload=no
LogCacheSecure=yes
LogReconnectInterval=600
LogDebug=never
LogSigned=always
LogMaxBigData=3072
LogMaxCacheSize=2GB
LogCacheLimitAction=stop logging
ForceServerVersionNumber=1.0.0
LogJavaClassPath=/opt/novell/idm/rbpm/UserApplication/NAuditPA.jar
```

The entries in the `logevent` file are not case sensitive, entries can appear in any order, empty lines are valid, and any line that starts with a hash (#) is commented out.

You must add the following entry into the `logevent` file to log events for the User Application:

```
LogJavaClassPath=/opt/novell/idm/rbpm/UserApplication/NAuditPA.jar
```

The User Application installation copies this file into the correct directory, but the entry must be manually added to the `logevent` file.

The following table provides an explanation of each setting in the `logevent` file. The Platform Agent is used by Sentinel and Novell Audit. The documentation for the Platform Agent is in the *Novell Audit 2.0 Administration Guide* (http://www.novell.com/documentation/novellaudit20/).

IMPORTANT: You must restart the Platform Agent any time you make a change to the configuration.

Table 5-2 *logevent Settings*

Setting	Description
LogHost=*dns_name*	The hostname or IP address of the Event Source Server where the Platform Agent sends events. In an environment where the Platform Agent connects to multiple hosts—for example, to provide load balancing or system redundancy—separate the IP address of each server with commas in the LogHost entry. For example, `LogHost=192.168.0.1,192.168.0.3,192.168.0.4` The Platform Agent connects to the servers in the order specified. If the first logging server goes down, the Platform Agent tries to connect to the second logging server, and so on.
LogCacheDir=*path*	The directory where the Platform Agent stores the cached event information if the Event Source Server becomes unavailable.
LogEnginePort=*port*	The port at which the Platform Agent can connect to the Event Source Server. By default, this is port 1289.

Setting	Description
LogCachePort=*port*	The port at which the Platform Agent connects to the Logging Cache Module. By default, this is port 1288. If the connection between the Platform Agent and the Event Source Server fails, Identity Manager continues to log events to the local Platform Agent. The Platform Agent simply switches into Disconnected Cache mode; that is, it begins sending events to the Logging Cache module (`lcache`). The Logging Cache module writes the events to the Disconnected Mode Cache until the connection is restored. When the connection to the Event Source Server is restored, the Logging Cache Module transmits the cache files to the Event Source Server. To protect the integrity of the data store, the Event Source Server validates the authentication credentials in each cache file before logging its events.
LogCacheUnload=Y|N	Set the parameter to `N` to prevent `lcache` from being unloaded.
LogCacheSecure=Y|N	Set the parameter to `Y` to encrypt the local cache file.
LogReconnectInterval=*seconds*	The interval, in seconds, at which the Platform Agent and the Platform Agent Cache try to reconnect to the Event Source Server if the connection is lost. By default, this is 600.
LogDebug=Never|Always	The Platform Agent debug setting. ◆ Set to `Never` to never log debug events. ◆ Set to `Always` to always log debug events.
LogSigned=Never|Always	The signature setting for Platform Agent events. **IMPORTANT:** Sentinel can receive and map Audit signatures to a Novell Sentinel event field; however, Novell Sentinel does not currently verify event signatures. ◆ Set to `Never` to never sign or chain events. ◆ Set to `Always` to always log events with a digital signature and to sequentially chain events.
LogMaxBigData=*bytes*	The maximum size of the event data field. The default value is 3072 bytes. Set this value to the maximum number of bytes the client allows. Data that exceeds the maximum is truncated or not sent if the application doesn't allow truncated events to be logged.
LogMaxCacheSize=*bytes*	The maximum size, in bytes, of the Platform Agent cache file. By default, the maximum size is 2 GB. If this size is not specified, the log cache file continues to grow till 2 GB.
LogCacheLimitAction=stop logging|drop cache	The action that you want the cache module to take when it reaches the maximum cache size limit. ◆ Set to `stop logging` if you want to stop collecting new events. ◆ Set to `drop cache` if you want to delete the cache and start over with any new events that are generated.

Setting	Description
ForceServerVersionNumber=*version number*	To instruct the Platform Agent to use a particular Secure Log Server protocol version if events are logged to a log server from Nsure Audit version 1.0.x. The valid values are: 1.0.0, 1.0.1, 1.0.2, 1.0.3, 1.0.3.P1, 1.0.3.P2, and and so on. If you are using patches from Nsure Audit 1.0.3, indicate the patch number being used, for example, P1, P2, P3, and so on. With Nsure Audit 1.0.3 Patch 2, the Secure Log Server properly reports the protocol in use and the Novell Audit 2.0.x Platform Agent automatically uses the protocol reported by the Secure Log Server.
LogJavaClassPath	The location of the `NAuditPA.jar` lcache file. For example: `LogJavaClassPath=/opt/novell/idm/rbpm/UserApplication/NAuditPA.jar`

NOTE: Some options might not be available in all versions of Novell Audit.

Proceed to Chapter 7, "Securing the Logging System," on page 29.

6 Installing and Configuring the Novell XDASv2

XDAS provides a standardized classification for audit events. The events are encapsulated within a hierarchical notational system that helps to extend the standard or existing event identifier set. XDAS events helps you understand the audit trails of heterogeneous applications. The audit events are logged to a socket, or a file, or to a Syslog Connector for further processing.The XDAS taxonomy defines a set of fields, of these the primary fields are observer, initiator and target. For more information, refer to the *Novell XDASv2 Administation Guide* (http://www.novell.com/documentation/edir88/edirxdas_admin/data/bookinfo.html).

6.1 Installing the Novell XDASv2

Novell XDAS is automatically installed if either the Novell Identity Manager Metadirectory Server or Novell Identity Manager Connected System option is selected during the Identity Manager install. For more information on the Identity Manager installation, see *Identity Manager 4.0.1 Framework Installation Guide* or the *Identity Manager 4.0.1 Integrated Installation Guide*. The XDAS library (xdaslog.rpm or xdaslog.dll) must be installed on every server running Identity Manager if you want to log Identity Manager events through XDAS.

6.2 Configuring the Novell XDASv2 Text File

After you install Identity Manager, you can configure the XDAS. The XDAS configuration settings are stored in a simple, text-based `xdasconfig.properties` configuration file. By default, xdasconfig.properties file is located in the following directories:

Table 6-1 *XDAS Configuration File*

Operating System	File
Linux/Solaris	`/etc/opt/novell/eDirectory/conf/xdasconfig.properties`
Windows	`/IDM_Install_Directory/xdasconfig.properties` On Windows, it is usually the Identity Manager install directory.

The following is a sample `xdasconfig.properties` file.

```
# Set the level of the root logger to DEBUG and attaches an appender named R.
log4j.rootLogger=debug, S, R

# Defines appender S to be a SyslogAppender.
#log4j.appender.S=org.apache.log4j.net.SyslogAppender
```

```
# Defines location of Syslog server.
#log4j.appender.S.Host=localhost
#log4j.appender.S.Port=port

# Specify protocol to be used (UDP/TCP/SSL)
#log4j.appender.S.Protocol=UDP

# Specify SSL certificate file for SSL connection.
# File path should be given with double backslash.
#log4j.appender.S.SSLCertFile=/etc/opt/novell/mycert.pem

# Minimum log-level allowed in syslog.
#log4j.appender.S.Threshold=INFO

# Defines the type of facility.
#log4j.appender.S.Facility=USER

# Layout definition for appender Syslog S.
#log4j.appender.S.layout=org.apache.log4j.PatternLayout
#log4j.appender.S.layout.ConversionPattern=%c : %p%m%n

# Defines appender R to be a Rolling File Appender.
#log4j.appender.R=org.apache.log4j.RollingFileAppender

# Log file for appender R.
#log4j.appender.R.File=/var/opt/novell/eDirectory/log/xdas-events.log

# Max size of log file for appender R.
#log4j.appender.R.MaxFileSize=100MB

# Set the maximum number of backup files to keep for appender R.
# Max can be 13. If set to zero, then there will be no backup files.
#log4j.appender.R.MaxBackupIndex=10

# Layout definition for appender Rolling log file R.
#log4j.appender.R.layout=org.apache.log4j.PatternLayout
#log4j.appender.R.layout.ConversionPattern=%d{MMM dd HH:mm:ss} %c : %p%m%n
```

Table 6-2 *XDAS Property File*

Options	ID
Syslog Appender	S
Rolling File Appender	R
Socket Appender	Socket Logger

The entries in the `xdasconfig.properties` file are not case sensitive, entries can appear in any order, empty lines are valid, and any line that starts with a hash (#) is commented out.

The following table provides an explanation of each setting in the `xdasconfig.properties` file.

IMPORTANT: You must restart eDirectory any time you make a change to the configuration.

Table 6-3 *XDAS Settings*

Setting	Description
log4j.rootLogger=*debug, S, R*	Sets the level of the root logger to debug and attaches an appender named R or S, where S specifies a Syslog appender and R specifies a Rolling File appender.

Setting	Description
log4j.appender.S=*org.apache.log4j.net.SyslogAppender*	Specifies the appender S to be a Syslog appender.
log4j.appender.S.Host=*localhost*	Specifies the location of the Syslog server where XDAS events are logged. In an environment where XDAS connects to multiple hosts. You need to separate the IP address of each server with commas in the log4j.appender.S.Host entry. For example, log4j.appender.S.Host=192.168.0.1,192.168.0.3,192.168.0.4 XDAS connects to the servers in the order specified. If the first logging server goes down, XDAS tries to connect to the second logging server, and so on.
log4j.appender.S.Port=*port*	The port at which the XDAS connects to the Syslog server. If the connection between XDAS and the Syslog server fails, Identity Manager cannot log events until the connection is restored.
log4j.appender.S.Protocol=UDP	Specifies the protocol to use. For example, UDP, TCP, or SSL.
log4j.appender.S.SSLCertFile=*/etc/opt/novell/mycert.pem*	Specifies the SSL certificate file for the SSL connection. Use double backslashes to specify the path of the file. This is an optional setting.
log4j.appender.S.Threshold=INFO	Specifies the minimum log level allowed in the Syslog appender.
log4j.appender.S.Facility=USER	Specifies the type of facility.
log4j.appender.S.layout=*org.apache.log4j.PatternLayout*	Layout setting for Syslog appender.
log4j.appender.S.layout.ConversionPattern=*%c : %p%m%n*	Layout setting for Syslog appender.
log4j.appender.R=*org.apache.log4j.RollingFileAppender*	Specifies appender R to be a Rolling File appender.
log4j.appender.R.File=*/var/opt/novell/eDirectory/log/xdas-events.log*	The location of the log file for a Rolling File appender.
log4j.appender.R.MaxFileSize=*100MB*	The maximum size, in MBs, of the log file for a Rolling File appender. Set this value to the maximum size that the client allows.
log4j.appender.R.MaxBackupIndex=*10*	Specify the maximum number of backup files for a Rolling File appender. The maximum number of the backup files can be 10. A zero value means no backup files.
log4j.appender.R.layout=*org.apache.log4j.PatternLayout*	Layout setting for Rolling File appender.
log4j.appender.R.layout.ConversionPattern=*%d{MMM dd HH:mm:ss} %c : %p%m%n*	Layout setting for Rolling File appender.

To enable the Syslog appender, make the following changes in the `xdas.properties` file:

1. Change the following entry to S to attach a Syslog appender:

   ```
   log4j.rootLogger=debug, S
   ```

2. Uncomment the following entries:

   ```
   log4j.appender.S=org.apache.log4j.net.SyslogAppender
   log4j.appender.S.Host=localhost
   log4j.appender.S.Port=port
   log4j.appender.S.Protocol=UDP
   log4j.appender.S.SSLCertFile=/etc/opt/novell/mycert.pem
   #log4j.appender.S.Threshold=INFO
   #log4j.appender.S.Facility=USER
   #log4j.appender.S.layout=org.apache.log4j.PatternLayout
   #log4j.appender.S.layout.ConversionPattern=%c : %p%m%n
   ```

3. Log into iManager and change the log events.

   For more information on changing log levels by using iManager, see Section 9.1, "Setting the Log Level and Maximum Log Size," on page 43.

4. Select the XDAS events for the driver set.

   For more information on selecting XDAS events by using iManager, see Section 8.4, "Selecting XDASv2 Events," on page 40.

5. Restart eDirectory.

To enable the Rolling File appender, make the following changes in the `xdas.properties` file:

1. Change the following entry to R to attach a Rolling File appender:

   ```
   log4j.rootLogger=debug, R
   ```

2. Uncomment the following entries:

   ```
   log4j.appender.R=org.apache.log4j.RollingFileAppender
   log4j.appender.R.File=/var/opt/novell/eDirectory/log/xdas-events.log
   log4j.appender.R.MaxFileSize=100MB
   log4j.appender.R.MaxBackupIndex=10
   log4j.appender.R.layout=org.apache.log4j.PatternLayout
   log4j.appender.R.layout.ConversionPattern=%d{MMM dd HH:mm:ss} %c : %p%m%n
   ```

3. Log into iManager and change log levels.

   For more information on changing log levels by using iManager, see Section 9.1, "Setting the Log Level and Maximum Log Size," on page 43.

4. Select the XDAS events for the driver set.

   For more information on selecting XDAS events by using iManager, see Section 8.4, "Selecting XDASv2 Events," on page 40.

5. Restart eDirectory.

Proceed to Chapter 7, "Securing the Logging System," on page 29.

7 Securing the Logging System

The Novell Sentinel server and Identity Manager Instrumentation utilize embedded certificates generated by an internal Certificate Authority (CA). These SSL certificates ensure that communications between the Identity Manager instrumentation and the Sentinel server are secure.

The next step is to define which events to log. Proceed to Chapter 8, “Managing Identity Manager Events,” on page 31.

8 Managing Identity Manager Events

The event information sent to Novell Sentinel is managed through product-specific instrumentations, or plug-ins. The Identity Manager Instrumentation allows you to configure which events are logged to your data store. You can select predefined log levels, or you can individually select the events you want to log. You can also add user-defined events to the Identity Manager schema.

The following sections review how to manage Identity Manager events:

8.1 Selecting Events to Log

The Identity Manager Instrumentation allows you to select events to be logged for the User Application, driver set, or a specific driver.

NOTE: Drivers can inherit logging configuration from the driver set.

8.1.1 Selecting Events for the User Application

The User Application enables you to change the log level settings of individual loggers and enable logging to the Platform Agent:

1. Log in to the User Application as the User Application Administrator.
2. Select the *Administration* tab.
3. Select the *Logging* link.

   The Logging Configuration page appears.

4 Select one of the following log levels for the listed logs.

Log Level	Description
Fatal	Writes Fatal level messages to the log.
Error	Writes Fatal and Error level messages to the log.
Warn	Writes Fatal, Error, and Warn level messages to the log.
Info	Writes Fatal, Error, Warn, and Info level messages to the log.
Debug	Writes Fatal, Error, Warn, Info, and debugging information to the log.
Trace	Writes Fatal, Error, Warn Info, debugging, and tracing information to the log.

5 Select the *Also send logging messages to Novell Audit* check box to send the events to the Platform Agent.

6 (Optional) Select *Also send logging messages to Open XDAS*, if you want to send the messages to Open XDAS.

For this option to work, you must select the open XDAS option during the installation of the User Application. For more information, see the *Identity Manager Roles Based Provisioning Module 4.0.1 User Application: Installation Guide*.

7 To save the changes for any subsequent application server restarts, select *Persist the logging changes*.

8 Click *Submit*.

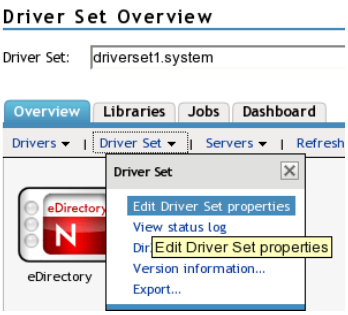
The User Application logging configuration is saved in `installdir/jboss/server/IDMProv/conf/idmuserapp_logging.xml`.

8.1.2 Selecting Events for the Driver Set

1 In iManager, select *Identity Manager* > *Identity Manager Overview*.

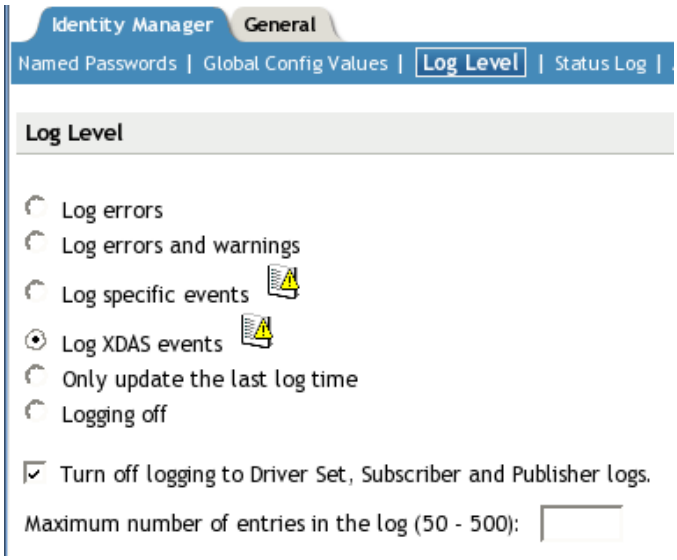
2 Browse to and select the driver set object.

3 Click the driver set object in the list of driver sets, then click *Driver Set* > *Edit Driver Set properties*.



4 Click the *Log Level* tab, then select a log level for the driver set.

For an explanation of each log level, see "Identity Manager Log Levels" on page 35.



5 Enable the *Turn off logging to Driver Set, Subscriber and Publisher logs* option to prevent logging audit events to eDirectory.

Enabling this option improves the performance of the Identity Manager system.

6 Click *Apply* or *OK* to save your changes.

NOTE: Changes to configuration settings are logged by default.

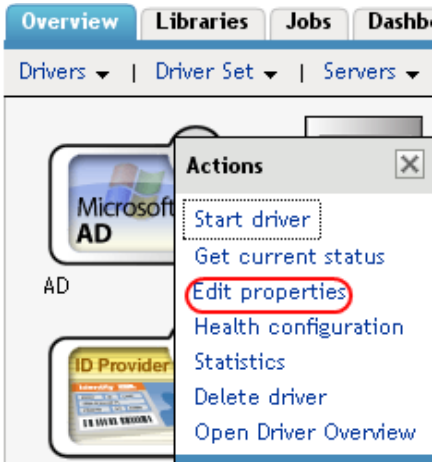
8.1.3 Selecting Events for a Specific Driver

1 In iManager, select *Identity Manager > Identity Manager Overview*.

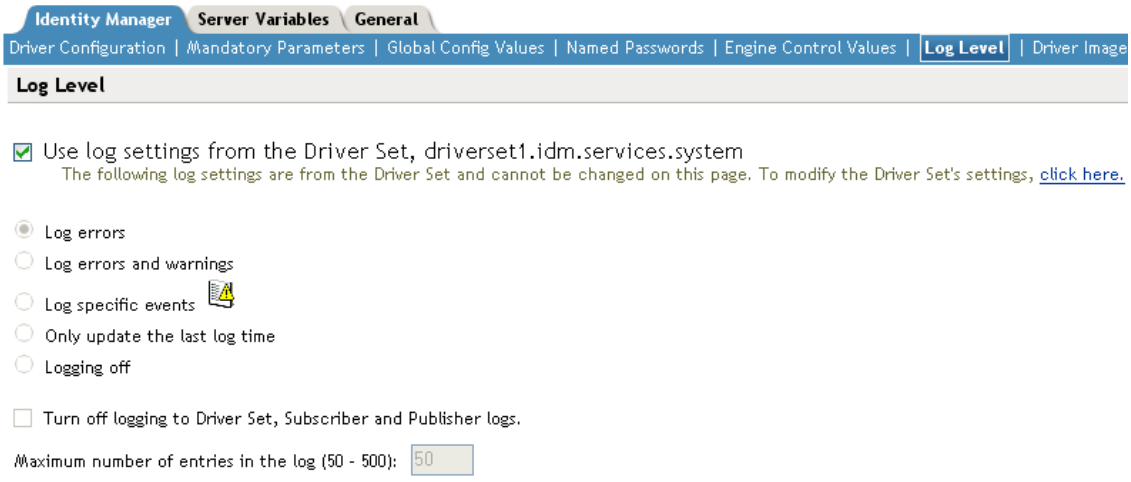
2 Browse to and select the driver set object that contains the driver

3 Select the driver set from the list of driver sets.

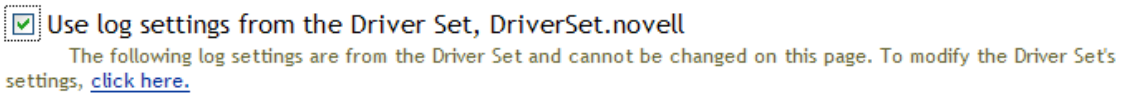
4 Click the upper right corner of the driver icon, then select *Edit properties*.



5 Select the *Log Level* tab.



6 (Optional) By default, the Driver object is configured to inherit log settings from the Driver Set object. To select logged events for this driver only, deselect *Use log settings from the Driver Set*.



7 Enable the *Turn off logging to Driver Set, Subscriber and Publisher logs* option.

Enabling this option improves the performance of the Identity Manager system.

8 Select a log level for the current driver.

For an explanation of each log level, see "Identity Manager Log Levels" on page 35.

9 Click *Apply* or *OK* to save your changes.

NOTE: Changes to configuration settings are logged by default.

8.1.4 Identity Manager Log Levels

The following table provides an explanation of the Identity Manager Instrumentation log levels:

Table 8-1 *Identity Manager Log Levels*

Option	Description
Log errors	This is the default log level. The Identity Manager Instrumentation logs user-defined events and all events with an error status. You receive only events with a decimal ID of 196646 and an error message stored in the Text1 field.
Log errors and warnings	The Identity Manager Instrumentation logs user-defined events and all events with an error or warning status. You receive only events with a decimal ID of 196646 or 196647 and an error or warning message stored in the first text field.
Log specific events	This option allows you to select the Identity Manager events you want to log. Click  to select the specific events you want to log. After you select the events you want to log, click *OK*. **NOTE:** User-defined events are always logged. For a list of all available events, see Appendix A, "Identity Manager Events," on page 51.
Log XDAS events	
Only update the last log time	The Identity Manager Instrumentation logs only user-defined events. When an event occurs, the last log time is updated so you can view the time and date of the last error in the status log.
Logging off	The Identity Manager Instrumentation logs only user-defined events.
Turn off logging to DriverSet, Subscriber and Publisher logs	Turns off logging to the Driver Set object, Subscriber, and Publisher logs.

Option	Description
Maximum Number of Entries in the Log	This setting allows you to specify the maximum number of entries to log in the status logs.

8.2 User-Defined Events

Identity Manager enables you to configure your own events to log to Novell Sentinel. Events can be logged by using an action in the Policy Builder, or within a style sheet. Any information you have access to when defining policies can be logged.

User-defined events are logged any time logging is enabled and are never filtered by the Metadirectory engine. There are two different ways to generate user-defined events:

- Section 8.2.1, "Using Policy Builder to Generate Events," on page 36
- Section 8.2.2, "Using Status Documents to Generate Events," on page 39

8.2.1 Using Policy Builder to Generate Events

1. In the Policy Builder, define the condition that must be met to generate the event, then select the *Generate Event* action.
2. Specify an event ID.

   Event IDs between 1000 and 1999 are allotted for user-defined events. You must specify a value within this range for the event ID when defining your own events. This ID is combined with the Identity Manager application ID of 003.
3. Select a log level.

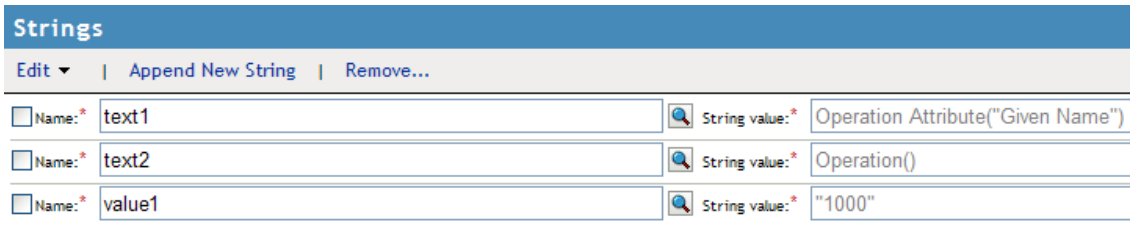
   Log levels enable you to group events based on the type of event being logged. The following predefined log levels are available:

Log Level	Description
log-emergency	Events that cause the Metadirectory engine or driver to shut down.
log-alert	Events that require immediate attention.
log-critical	Events that can cause parts of the Metadirectory engine or driver to malfunction.
log-error	Events describing errors that can be handled by the Metadirectory engine or driver.
log-warning	Negative events not representing a problem.
log-notice	Positive or negative events an administrator can use to understand or improve use and operation.
log-info	Positive events of any importance.
log-debug	Events of relevance for support or for engineers to debug the Metadirectory engine or driver.

4. Click the  icon next to the *Enter Strings* field to launch the Named String Builder.

   In the Named String Builder, you can specify the string, integer, and binary values to include with the event.

5 Use the Named String Builder to define the event values.



The Identity Manager event structure contains a target, a subTarget, three strings (text1, text2, text3), two integers (value1, value3), and a generic field (data). The text fields are limited to 256 bytes, and the data field can contain up to 3 KB of information, unless a larger data field is enabled in your environment.

The following table provides an explanation of the Identity Manager event structure:

Field	Description
target	This field captures the event target. All eDirectory events store the event's object in the *Target* field.
target-type	This field specifies which predefined format the target is represented in. Defined values for this type are as follows: ◆ 0: None ◆ 1: Slash Notation ◆ 2: Dot Notation ◆ 3: LDAP Notation
subTarget	This field captures the subcomponent of the target that was affected by the event. All eDirectory events store the event's attribute in the *SubTarget* field.
text1	The value of this field depends upon the event. It can contain any text string up to 255 characters.
text2	The value of this field depends upon the event. It can contain any text string up to 255 characters.
text3	The value of this field depends upon the event. It can contain any text string up to 255 characters.
*value*1	The value of this field depends upon the event. It can contain any numeric value up to 32 bits.
value3	The value of this field depends upon the event. It can contain any numeric value up to 32 bits.
data	The value of this field depends upon the event. The default size of this field is 3072 characters. You can configure the size of this field in the LogMaxBigData value in `logevent.cfg`. This value does not set the size of the *Data* field, but it does set the maximum size that the Platform Agent can log. For more information, see Chapter 5, "Installing and Configuring the Platform Agent," on page 19. The maximum size of the *Data* field is defined by the database where the data is logged, so the size varies for each database that is used. If the size of the *Data* field logged by the Platform Agent exceeds the maximum size allowed by the database, the channel driver truncates the data in the *Data* field. If an event has more data than can be stored in the *String* and *Numeric* value fields, it is possible to store up to 3 KB of binary data in the *Data* field.

6 Click *OK* to return to the Policy Builder to construct the remainder of your policy.

For more information and examples of the Generate Event action, see "Generate Event" in the *Policies in Designer 4.0.1* guide.

8.2.2 Using Status Documents to Generate Events

Status documents generated through style sheets using the `<xsl:message>` element are sent to Sentinel with an event ID that corresponds to the status document level attribute. The level attributes and corresponding event IDs are defined in the following table:

Table 8-2 *Status Documents*

Status Level	Status Event ID
Success	EV_LOG_STATUS_SUCCESS (1)
Retry	EV_LOG_STATUS_RETRY (2)
Warning	EV_LOG_STATUS_WARNING (3)
Error	EV_LOG_STATUS_ERROR (4)
Fatal	EV_LOG_STATUS_FATAL (5)
User Defined	EV_LOG_STATUS_OTHER (6)

The following example generates an event 0x004 and value1=7777, with a level of EV_LOG_STATUS_ERROR:

```
<xsl:message>
   <status level="error" text1="This would be text1" value1="7777">This data would
be in the blob and in text 2, since no value is specified for text2 in the
attributes.</status>
</xsl:message>
```

The following example generates an event 0x004 and value1=7778, with a level of EV_LOG_STATUS_ERROR:

```
<xsl:message>
   <status level="error" text1="This would be text1" text2="This would be text2"
value1="7778">This data would be in the blob only for this case, since a value for
text2 is specified in the attributes.</status>
</xsl:message>
```

8.3 eDirectory Objects that Store Identity Manager Event Data

The Identity Manager events you want to log are stored in the DirXML-LogEvent attribute on the Driver Set object or Driver object. The attribute is a multi-value integer with each value identifying an event ID to be logged.

You do not need to modify these attributes directly, because these objects are automatically configured based on your selections in iManager.

Before logging an event, the engine checks the current event type against the contents of the DirXML-LogEvent attribute to determine whether the event should be logged.

Drivers can inherit log settings from the driver set. The DirXML-DriverTraceLevel attribute of a Driver object has the highest precedence when determining log settings. If a Driver object does not contain a DirXML-DriverTraceLevel attribute, the engine uses the log settings from the parent driver set.

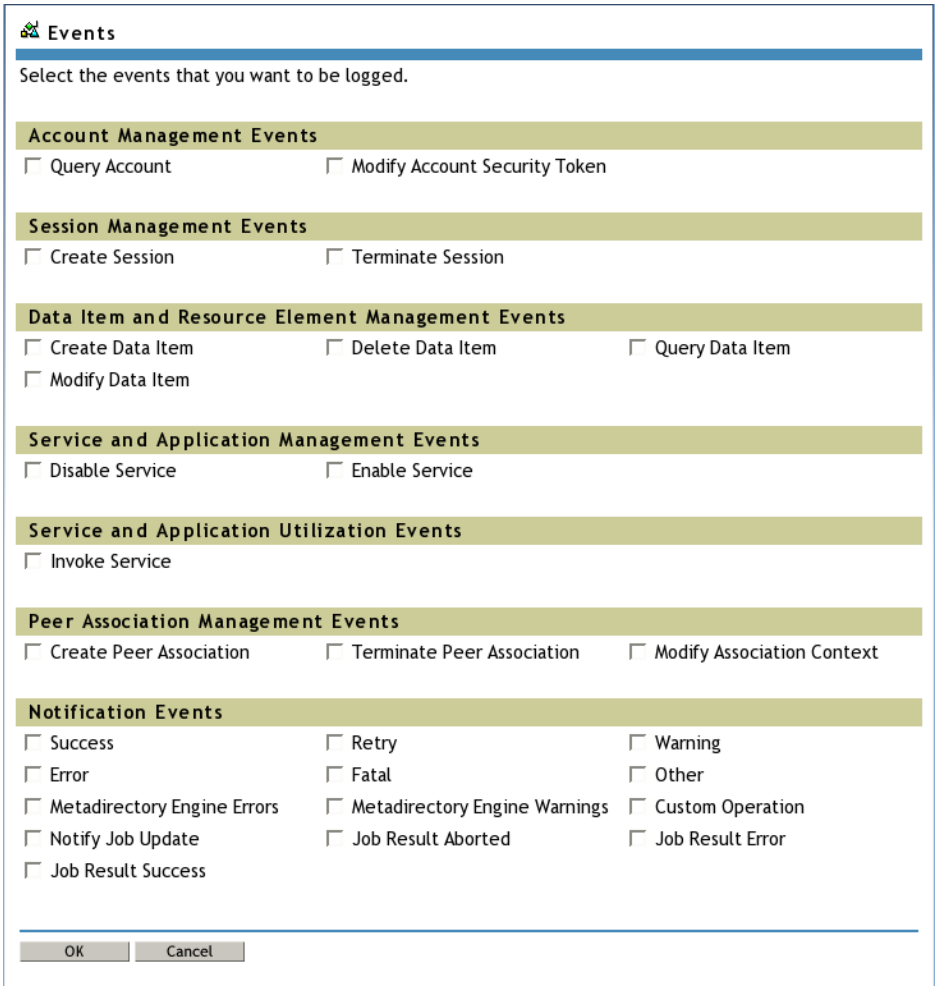
The next step is to generate reports. Proceed to Chapter 10, “Querying and Reporting,” on page 47.

8.4 Selecting XDASv2 Events

1. In iManager, select *Identity Manager* > *Identity Manager Overview*.
2. Browse to and select the driver set object.
3. Click the driver set object in the list of driver sets, then click *Driver Set* > *Edit Driver Set properties*.

4. Click the *Log Level* tab, then select the *Log XDAS Events* for the driver set.

   For an explanation of each log level, see "Identity Manager Log Levels" on page 35.



5 Click *Apply* or *OK* to save your changes.

NOTE: Changes to configuration settings are logged by default.

When an XDAS event is selected, the corresponding subevents as described in the mapping in Table A-12 on page 60 are selected for logging. The set of events configured for logging are same for Novell Audit and Novell XDASv2.

8.5 Correlation ID in XDAS Events

Correlation ID is used to identify related events from Identity Manager and eDirectory. In XDAS log, Correlation ID is logged in the *Action.Event.CorrelationID* field. CorrelationID. For Identity Manager events, the Correlation ID is logged in the following format:

```
Driver Name#Channel Name#UUID
```

UUID is a Universally unique identifier which is a type 4 (pseudo randomly generated) UUID

For example, an event generated in the publisher channel of the Delimited Text driver has an event-id=Delimited Text #Publisher#0:a8b0c1be-01d5-4b55-ad6e-0ff87e31e5d7. The related publisher channel events logged Correlation ID is set to Delimited Text#Publisher#a8b0c1be-01d5-4b55-ad6e-

0ff87e31e5d7. eDirectory and Identity Manager events that are part of the same transaction have this UUID as part of Correlation ID logged with XDAS. This information is not available with logging by using Novell Audit Platform Agent.

9 Using Status Logs

In addition to the functionality provided by Sentinel, Identity Manager logs a specified number of events on the driver set and the driver. These status logs provide a view of recent Identity Manager activity. After the log reaches the set size, the oldest half of the log is permanently removed to clear room for more recent events. Therefore, any events you want to track over time should be logged to Sentinel.

The following sections contain information on the Identity Manager logs:

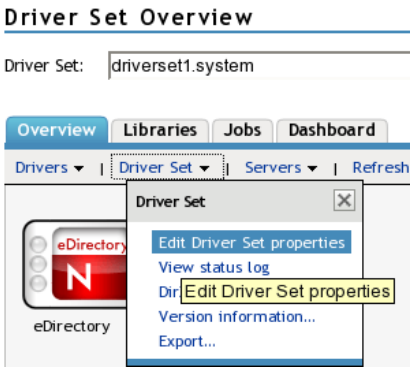
9.1 Setting the Log Level and Maximum Log Size

Status logs can be configured to hold between 50 and 500 events. This setting can be configured for the driver set to be inherited by all drivers in the driver set, or configured for each driver in the driver set. The maximum log size operates independently of the events you have selected to log, so you can configure the events you want to log for the driver set, then specify a different log size for each driver in the set.

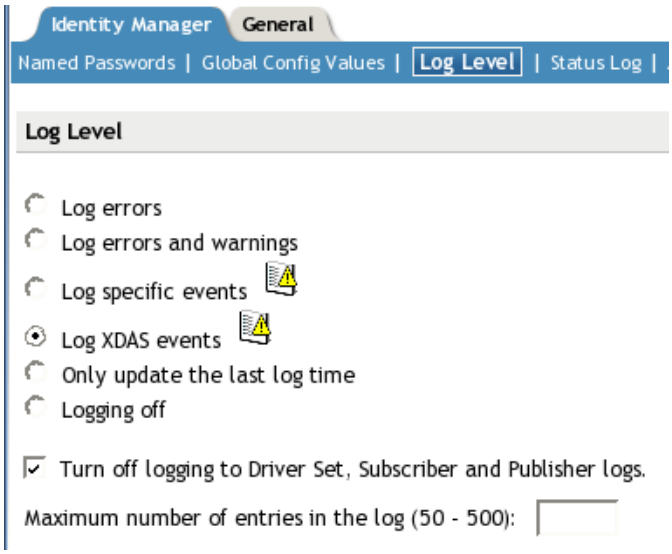
This section reviews how to set the maximum log size on the driver set or an individual driver:

9.1.1 Setting the Log Level and Log Size for the Driver Set

1. In iManager, select *Identity Manager > Identity Manager Overview*.
2. Browse to and select the driver set.
3. Click the driver set name to access the driver set overview page.
4. Select *Driver Set > Edit Driver Set properties*.



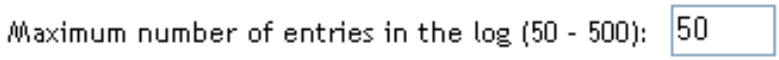
5 Select *Log Level*.



6 Enable the *Turn off logging to Driver Set, Subscriber and Publisher logs* option to prevent logging audit events to eDirectory.

Enabling this option improves the performance of the Identity Manager system.

7 Specify the maximum log size in the *Maximum number of entries in the log* field:



8 After you have specified the maximum number, click *OK*.

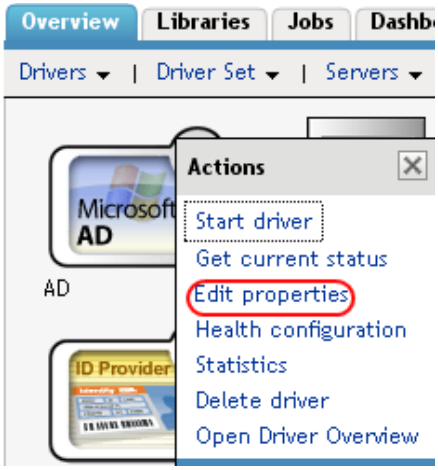
9.1.2 Setting the Log Level and Log Size for the Driver

1 In iManager select *Identity Manager* > *Identity Manager Overview*.

2 Browse to and select the driver set.

3 Click the driver set to access the driver set overview page.

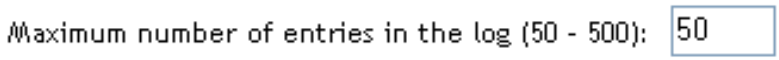
4 Click the upper right corner of the driver icon, then select *Edit properties*.



5 Select *Log Level*.

6 Deselect *Use log settings from the driver set* option, if it is selected.

7 Specify the maximum log size in the *Maximum number of entries in the log* field:



8 After you have specified the maximum number, click *OK*.

9.2 Viewing Status Logs

The status logs are short-term logs for the driver set, the Publisher channel, and the Subscriber channel. They are accessed through different locations in iManager.

- Section 9.2.1, "Accessing the Driver Set Status Log," on page 45
- Section 9.2.2, "Accessing the Publisher Channel and Subscriber Channel Status Logs," on page 46

9.2.1 Accessing the Driver Set Status Log

The status log for the driver set contains only messages generated by the engine, such as state changes for any drivers in the driver set. All engine messages are logged. There are two ways to access the driver set status log:

- "Viewing the Log from the Driver Set Overview Page" on page 45
- "Viewing the Log from the Driver Overview Page" on page 45

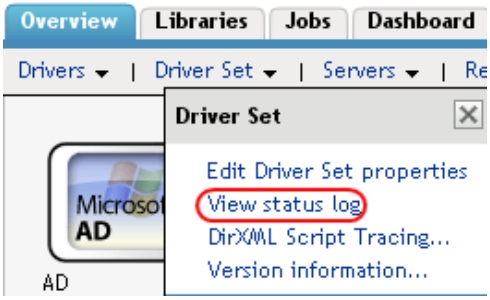
Viewing the Log from the Driver Set Overview Page

1 In iManager, select *Identity Manager* > *Identity Manager Overview*.

2 Browse to and select the driver set.

3 Click the driver set to access the driver set overview page.

4 Select *Driver Set* > *View status log*.



Viewing the Log from the Driver Overview Page

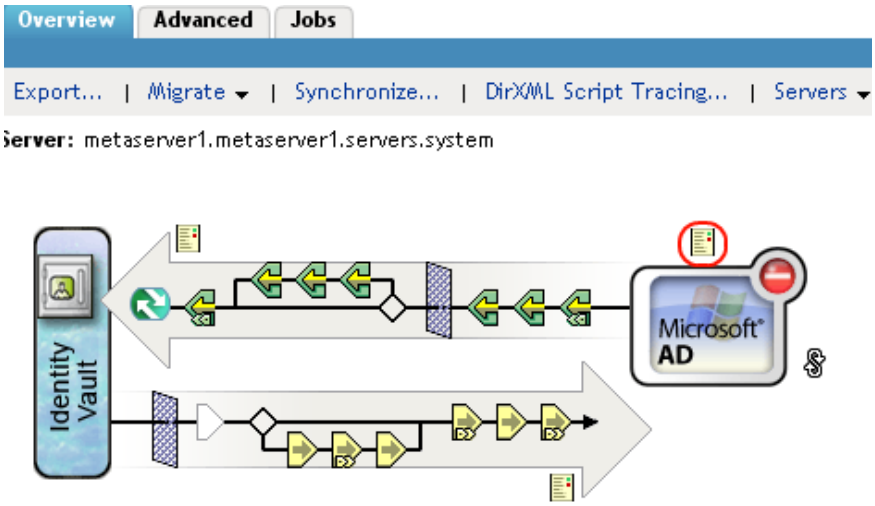
1 In iManager, select *Identity Manager* > *Identity Manager Overview*.

2 Browse to and select the driver set.

3 Click the driver set to access the driver set overview page, then click any driver.

The status log for the driver is stored on the driver overview page for each driver.

4 Click the Driver Set Status Log icon above the driver object.



9.2.2 Accessing the Publisher Channel and Subscriber Channel Status Logs

The status logs for the Publisher and Subscriber channels report channel-specific messages generated by the driver, such as an operation veto for an unassociated object.

To access the Publisher channel and the Subscriber channel logs:

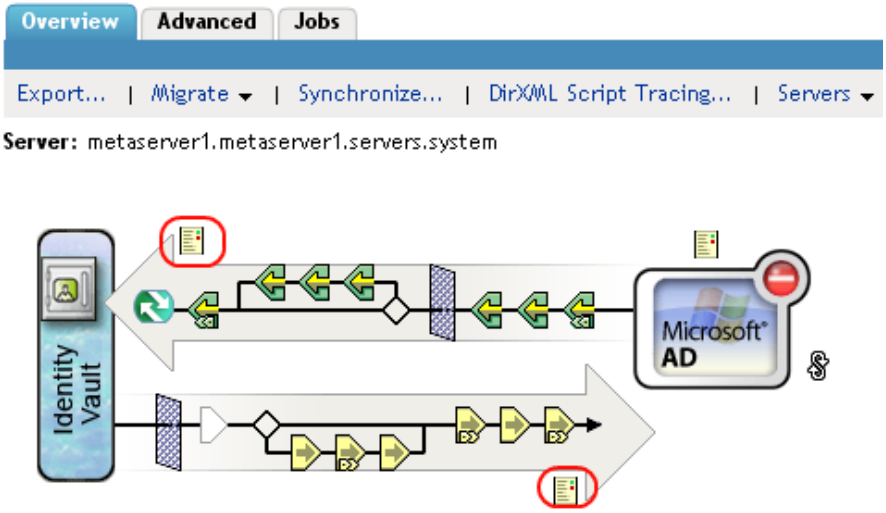
1 In iManager, select *Identity Manager* > *Identity Manager Overview*.

2 Browse to and select the driver set.

3 Click the driver set to access the driver set overview page.

4 Click the desired driver object.

5 Click the Publisher channel or the Subscriber channel status log icon.



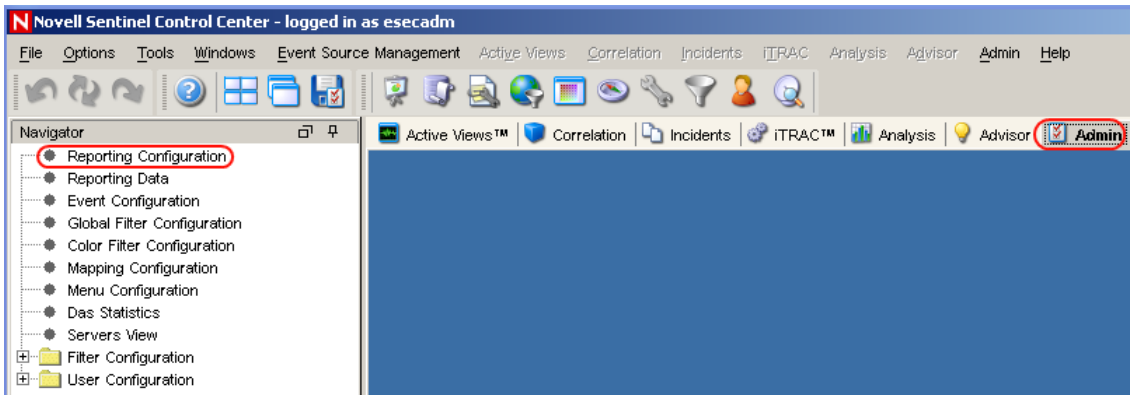
10 Querying and Reporting

After you integrate Identity Manager with Novell Sentinel, you can log system information to a central data store. However, logging information is only half the battle. Obviously, you have to be able to access and understand your log data for the information to be useful. Queries and reports allow you to view and interpret the information in your data store.

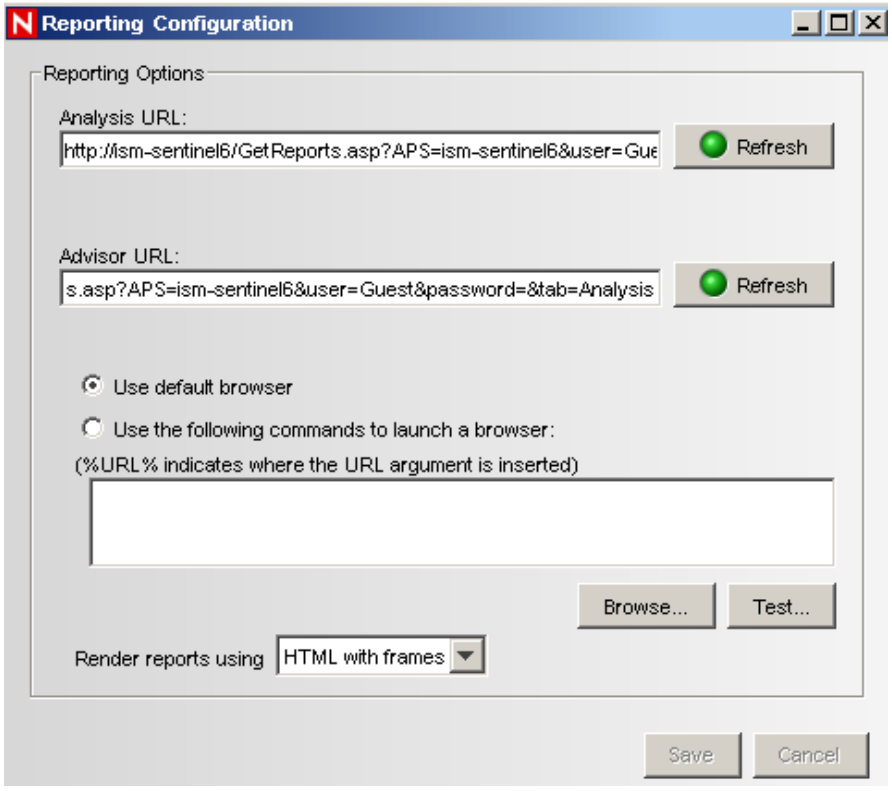
The Identity Manager Collector provides a number of Crystal Decisions reports (`*.rpt`) that simplify gathering information on common operations performed in Identity Manager. The term "reports" refers specifically to Crystal Decisions report template files (`*.rpt`). Crystal Decisions reports graphically summarize specific sets of log data in pie charts, bar charts, and so forth. These reports are included with the current version of the Identity Manager Collector, which can be downloaded from, Sentinel 6.1 Connectors Web site (http://support.novell.com/products/sentinel/secure/sentinel61.html).

Novell Sentinel is integrated with Crystal Reports to generate and display reports. To run the report templates, you must first configure the location of the Crystal Enterprise Server that publishes reports in the General Options window of the Admin page.

1 In the Sentinel Control Center, select the *Admin* tab, then select the *Reporting Configuration* option in the Navigator pane.



2 Specify the location of the Crystal Enterprise server, then click *Save*.



After Novell Sentinel is configured to access the Crystal Enterprise server, the Analysis page allows administrators to run historical reports. Vulnerability reports are available from the Advisor page. These reports are published on a Web server, they run directly against the database, and they then appear on the *Analysis* and *Advisor* tabs under the Navigator pane.

The reports are updated regularly. The following is a list of the categories of reports that are available:

- **Collector Pack Controls:** Contains reports about the Collector Pack setup, dashboard status, and implementation of audit trails.
- **Collector Controls:** Contains reports about event trends and Collector management.
- **Account Management Controls:** Contains reports about user account provisioning, user account management, account access management, and user password management.
- **Trust Management Controls:** Contains reports about trust provisioning, trust management, and trust access management.
- **Object Management Controls:** Contains reports about object provisioning and object management.
- **Authentication Controls:** Contains reports about authentication by servers and users.
- **Workflow Management:** Contains reports that monitor workflows and the resources requested in the workflows.

For more information on running reports in Novell Sentinel, see the "Analysis Tab" and "Advisor Usage and Maintenance" sections in the *Novell Sentinel User's Guide* (http://www.novell.com/documentation/sentinel61/pdfdoc/sentinel_61_user_guide.pdf).

11 Integrating Sentinel with EAS

Event Auditing Service (EAS) is the preferred event destination for the Identity Manager components. EAS provides event forwarding capabilities that links Sentinel with EAS. EAS is configurable through the Identity Reporting application. The event forwarding uses a Sentinel link to push events on to Sentinel. EAS captures log events associated with actions performed in the Identity Reporting Module, the Roles Based Provisioning Module (RBPM), the Role Mapping Administrator (RMA), Identity Manager, and the Identity Vault.

Currenlty, the Identity Manager components cannot send events directly to Sentinel, and then have Sentinel forward events to EAS. The reports that the Identity Reporting application generates are only available on event data present in EAS, same reports cannot be generated against Sentinel at this time. For more information on EAS, refer to the *Identity Reporting Module Guide*.

A Identity Manager Events

This section provides a listing of all events logged by Identity Manager.

A.1 Event Structure

All events logged through Sentinel have a standardized set of fields. This allows Sentinel to log events to a structured database and query events across all logging applications.

Identity Manager events provide information in the following field structure:

EventID, Description, Originator Title, Target Title, Subtarget Title, Text1 Title, Text2 Title, Text3 Title, Value1 Title, Value1 Type, Value2 Title, Value2 Type, Value3 Title, Value3 Type, Group Title, Group Type, Data Title, Data Type, Display Schema.

For a complete explanation of the event structure, see Event Structure (http://www.novell.com/documentation/novellaudit20/novellaudit20/data/al9m381.html) in the *Novell Audit 2.0 Administration Guide*.

A.2 Error and Warning Events

Identity Manager generates an event whenever an error or warning is encountered. The following table lists the Identity Manager error and warning events:

Table A-1 *Error and Warning Events*

Event	Log Level	Information
DirXML_Error	LOG_ERROR	All Identity Manager errors log this event. The actual error code encountered is stored in the event. To log errors, select the *Log Errors* or *Log Errors and Warnings* log level on the driver set or the individual driver. You can also select the *Log Specific Events* option and select this event. For more information, see Section 8.1, "Selecting Events to Log," on page 31.
DirXML_Warning	LOG_WARNING	All Identity Manager warnings log this event. The actual warning code encountered is stored in the event. To log errors, select the *Log Errors* or *Log Errors and Warnings* log level on the driver set or the individual driver. You can also select the *Log Specific Events* option and select this event. For more information, see Section 8.1, "Selecting Events to Log," on page 31.

A.3 Job Events

The following table lists the Job events that can be audited through Sentinel:

Table A-2 *Job Events*

Event ID	Description	Trigger
303E4	Job Result Aborted	Occurs when a running job is aborted by a client.
303E5	Job Result Error	Occurs when a running job reports an error for some operation. (A running job can report status multiple times during the job execution.)
303E6	Job Result Warning	Occurs when a running job reports a warning for some operation.
303E7	Job Result Success	Occurs when a running job reports success for some operation.

See Section A.13, "Log Schema Files," on page 60 for information on understanding the logged events.

A.4 Remote Loader Events

The following table lists the Remote Loader events that can be audited through Sentinel:

Table A-3 *Remote Loader Events*

Event ID	Description	Trigger
30BB8	Remote Loader Start	Occurs when the Remote Loader starts.

Event ID	Description	Trigger
30BB9	Remote Loader Stop	Occurs when the Remote Loader stops.
30BBA	Remote Loader Connection Established	Occurs when the engine establishes a TCP connection with the Remote Loader.
30BBB	Remote Loader Connection Dropped	Occurs when the engine-to-Remote Loader connection is lost.

See Section A.13, "Log Schema Files," on page 60 for information on understanding the logged events.

IMPORTANT: To log these events, you must select the *Log Specific Events* log level and select the events you want to log. For more information, see Section 8.1, "Selecting Events to Log," on page 31.

A.5 Object Events

The following table lists the object events that can be audited through Sentinel:

Table A-4 *Object Events*

Event ID	Description	Trigger
31400	Delete_Entity	Occurs when an object is deleted.
31401	Update_Entity	Occurs when an object is modified.
31440	Create_Entity	Occurs when an object is created.

See Section A.13, "Log Schema Files," on page 60 for information on understanding the logged events.

A.6 Password Events

The following table lists the change password events that can be audited through Novell Sentinel:

Table A-5 *Password Events*

Event ID	Description	Trigger
31410	Change_Password_Failure	Occurs when a password change fails.
31411	Change_Password_Success	Occurs when a password change is successful.
31420	Forgot_Password_Change_Failure	Occurs when the Forgot Password change fails.
31421	Forgot_Password_Change_Success	Occurs when the Forgot Password change is successful.

See Section A.13, “Log Schema Files,” on page 60 for information on understanding the logged events.

A.7 Search List Events

The following table lists the search events that can be audited through Sentinel:

Table A-6 *Search List Events*

Event ID	Description	Trigger
31430	Search_Request	Occurs when a user performs a search request.
31431	Search_Saved	Occurs when the user selects My Saved Searches.

See Section A.13, “Log Schema Files,” on page 60 for information on understanding the logged events.

A.8 Engine Events

The following table lists the engine events that can be audited through Sentinel:

Table A-7 *Engine Events*

Event ID	Description	Trigger
30001	Status Success	Many different events can cause the status success event to occur. It usually signifies that an operation was successfully completed.
30002	Status Retry	Many different events can cause the status retry event to occur. It signifies an operation was not completed and the operation must be tried again later.
30003	Status Warning	Many different events can cause the status warning event to occur. It usually signifies that an operation was completed with minor problems.
30004	Status Error	Many different events can cause the status error event to occur. It usually signifies that an operation was not completed successfully.
30005	Status Fatal	Many different events can cause the status fatal event to occur. It usually signifies that an operation was not completed successfully and the engine or driver could not continue.
30006	Status Other	Any status document processed with a level other than the five previously defined creates a status other event. These events can only be generated within a style sheet or rule.
30007	Search	Occurs when a query document is sent to the Identity Manager engine or driver.
30008	Add Entry	Occurs when an object is added.
30009	Delete Entry	Occurs when an object is deleted.
3000A	Modify Entry	Occurs when an object is modified.

Event ID	Description	Trigger
3000B	Rename Entry	Occurs when an object is renamed.
3000C	Move Entry	Occurs when an object is moved.
3000D	Add Association	Occurs when an association is added. It can happen on an add or a match.
3000E	Remove Association	When an object is deleted, there is no remove association event. The remove association occurs when a User object is deleted in the disparate application, and the delete is then converted into a modify that removes the association.
3000F	Query Schema	Occurs when a query schema operation is sent to the Identity Manager engine or driver.
30010	Check User Password Status	Manual function that is initiated via iManager to check the status of the user’s password.
30011	Check Object Password	Occurs when a request is issued to check an object's password, other than the driver.
30012	Change Password	Occurs when a request is issued to change the driver's password.
30013	Sync	Occurs when a sync event is requested.
30014	Input XML Document	Generated whenever an input document is created by the engine or driver.
30015	Input Transformation Document	Generated after the input transformation policies are processed, allowing the user to view the transformed document.
30016	Output Transformation Document	Generated after the output transformation policies are processed, allowing the user to view the transformed document.
30017	Event Transformation Document	Generated after the event transformation policies are processed, allowing the user to view the transformed document.
30018	Placement Rule Transformation Document	Generated after the Placement rule policies are processed, allowing the user to view the transformed document.
30019	Create Rule Transformation Document	Generated after the Create rule policies are processed, allowing the user to view the transformed document.
3001A	Input Mapping Rule Transformation Document	Generated after the Schema Mapping rules are processed which convert the document to the eDirectory schema.
3001B	Output Mapping Rule Transformation Document	Generated after the Schema Mapping rules are processed which convert the document to the applications schema.
3001C	Matching Rule Transformation Document	Generated after the Matching rule policies are processed, allowing the user to view the transformed document.

Event ID	Description	Trigger
3001D	Command Transformation Document	Generated after the command transformation policies are processed, allowing the user to view the transformed document.
3001E	Publisher Filter Transformation Document	Generated after the processing the notify filter on the Publisher channel, allowing the user to view the transformed document.
3001F	User Agent Request	Occurs when a User Agent XDS command document is sent to the Driver on the Subscriber channel.
30020	Resync Driver	Occurs when a resync request is issued.
30021	Migrate	Occurs when a migrate request is issued.
30022	Driver Start	Occurs when a driver is started.
30023	Driver Stop	Occurs when a driver is stopped.
30024	Password Sync	Generated when setting the distribution or simple password on an object.
30025	Password Reset	Generated when resetting the connected application password after a failed password sync operation.
30026	DirXML Error	Generated whenever the engine throws an internal error.
30027	DirXML Warning	Generated whenever the engine throws an internal warning.
30028	Custom Operation	Occurs when an unknown operation appears in an input document. An example of known operations would be an add, delete, or modify.
30029	Clear Attribute	Occurs when a modify operation contains a remove-all-value element.
3002A	Add Value - Modify Entry	Occurs when a value is added during the modification of an object.
3002B	Remove Value	Occurs when a modify operation contains a remove-value element.
3002C	Merge Entries	Occurs when two objects are being merged.
3002D	Get Named Password	Generated on a Get Named Password operation.
3002E	Reset Attributes	Occurs when a Reset document is issued on the publisher or Subscriber channels.
3002F	Add Value - Add Entry	Occurs when a value is added during the creation of an object.
30030	Set SSO Credential	Occurs when a driver policy executes the do-set-sso-credential action.
30031	Clear SSO Credential	Occurs when a driver policy executes the do-clear-sso-credential action.
30032	Set SSO Passphrase	Occurs when a driver policy executes the do-clear-sso-credential action.

See Section A.13, “Log Schema Files,” on page 60 for information on understanding the logged events.

A.9 Server Events

The following table lists the server events that can be audited through Sentinel:

Table A-8 *Server Events*

Event ID	Description	Trigger
307D0	Config:Log Events	Occurs when the log events attribute is changed on the Driver or Driver Set object.
307D1	Config:Driver Cache Limit	Occurs when the Driver Cache Limit attribute is changed on a Driver object.
307D2	Config:Driver Set	Occurs when the Driver Set/Server association is changed.
307D3	Config:Driver Start Option	Occurs when the Driver Start Option is changed for a Driver object.
307D4	Driver Resync	Occurs when a resynchronization is issued for the driver.
307D5	Migrate Application Server	Occurs when the migration of the application server happens.
307D6	Shim Password Set	Occurs when the Application password is set.
307D7	Keyed Password Set	Occurs when the Identity Manager engine receives a client request to set a named password on an object.
307D8	Remote Loader Password Set	Occurs when the Remote Loader password is set.
307DA	Get Server Certificate	Occurs when the Identity Manager engine receives a client request for the engine's public key certificate (used in encrypting passwords with the Identity Manager verbs).
307DB	Cache Utility	Occurs when the Identity Manager engine receives a client request for the engine's public key certificate (used in encrypting passwords with the Identity Manager verbs).
307DC	Check Object Password	Occurs when the Identity Manager engine receives a client request asking the engine to check if an eDirectory object's nspmDistributionPassword value matches the password value in a connected system.
307DD	Initialize Driver Object	Occurs when the Identity Manager engine receives a client request to initialize a DirXML-Driver object.
307DE	Notify Job Update	Occurs when the Identity Manager engine receives a client request informing the engine that a DirXML-Job object has changed and that the engine needs to update the information it has cached about the job object.
307DF	Open Driver Action	Occurs when the Identity Manager engine receives a client request to submit a command or event document directly to a driver.
307E0	Queue Driver Event	Occurs when the Identity Manager engine receives a client request to submit a command document to a driver's event queue.
307E1	Start Job	Occurs when a job starts.
307E2	Abort Job	Occurs when a job aborts.

See Section A.13, “Log Schema Files,” on page 60 for information on understanding the logged events.

A.10 Security Events

The following table lists the security events that can be audited through Sentinel:

Table A-9 *Security Events*

Event ID	Description	Trigger
31450	Create_Proxy_Definition_Success	Occurs on successful creation of a proxy definition.
31451	Create_Proxy_Definition_Failure	Occurs on failed creation of a proxy definition.
31452	Update_Proxy_Definition_Success	Occurs on successful update of a proxy definition.
31453	Update_Proxy_Definition_Failure	Occurs on failed update of a proxy definition.
31454	Delete_Proxy_Definition_Success	Occurs on successful deletion of a proxy definition.
31455	Delete_Proxy_Definition_Failure	Occurs on failed deletion of a proxy definition.
31456	Create_Delegatee_Definition_Success	Occurs on successful creation of a delegatee definition.
31457	Create_Delegatee_Definition_Failure	Occurs on failed creation of a delegatee definition.
31458	Update_Delegatee_Definition_Success	Occurs on successful update of a delegatee definition.
31459	Update_Delegatee_Definition_Failure	Occurs on failed update of a delegatee definition.
3145A	Delete_Delegatee_Definition_Success	Occurs on successful deletion of a delegatee definition.
3145B	Delete_Delegatee_Definition_Failure	Occurs on failed deletion of a delegatee definition.
3145C	Create_Availability_Success	Occurs on successful creation of the availability status.
3145D	Create_Availability_Failure	Occurs on failed creation of the availability status.
3145E	Delete_Availability_Success	Occurs on successful deletion of the availability status.
3145F	Delete_Availability_Failure	Occurs on failed deletion of the availability status.

See Section A.13, “Log Schema Files,” on page 60 for information on understanding the logged events.

A.11 Workflow Events

The following table lists the User Application events that can be audited through Sentinel:

Table A-10 *Workflow Events*

Event ID	Description	Trigger
31520	Workflow_Error	Occurs when there is a workflow error.
31521	Workflow_Started	Occurs when the workflow starts.

Event ID	Description	Trigger
31522	Workflow_Forwarded	Occurs when the workflow is forwarded.
31523	Workflow_Reassigned	Occurs when the workflow is reassigned.
31524	Workflow_Approved	Occurs when the workflow is approved.
31525	Workflow_Refused	Occurs when the workflow is refused.
31526	Workflow_Ended	Occurs when the workflow ends.
31527	Workflow_Claimed	Occurs when the workflow is claimed.
31528	Workflow_Unclaimed	Occurs when the workflow is not claimed.
31529	Workflow_Denied	Occurs when the workflow is denied.
3152A	Workflow_Completed	Occurs when the workflow is completed.
3152B	Workflow_Timedout	Occurs when the workflow timed out.
3152C	User_Message	This is a user adhoc log message.
3152D	Provision_Error	Occurs when there is an error in the provisioning step.
3152E	Provision_Submitted	Occurs during the provisioning step on submission of entitlements.
3152F	Provision_Success	Occurs during the provisioning step on successful completion of the step.
31530	Provision_Failure	Occurs during the provisioning step upon failure of the step.
31531	Provision_Granted	Occurs during the provisioning step on granting of an entitlement.
31532	Provision_Revoked	Occurs during the provisioning step on the revoking of an entitlement.
31533	Workflow_Retracted	Occurs when the workflow is retracted.
31534	Workflow_Escalated	Occurs when the workflow is escalated.
31535	Workflow_Reminder_Sent	Occurs when reminders are sent to addressees of a workflow task.
31536	Digital_Signature	Occurs whenever a digital signature is passed to the workflow engine.
31470	Digital_Signature_Verification_Request	Occurs when a digital signature request is verified.
31471	Digital_Signature_Verification_Failure	Occurs if a digital signature is invalid.
31472	Digital_Signature_Verification_Success	Occurs upon successful verification of a digital signature.
31537	Workflow_ResetPriority	Occurs when the priority of a workflow task is reset.

See Section A.13, "Log Schema Files," on page 60 for information on understanding the logged events.

A.12 Driver Start and Stop Events

Identity Manager can generate an event whenever a driver starts or stops. The following table lists these events:

Table A-11 Driver Start and Stop Events

Event	Log Level	Information
EV_LOG_DRIVER_START	LOG_INFO	To log driver starts, select the *Log Specific Events* log level and specify this event. For more information, see Section 8.1, "Selecting Events to Log," on page 31
EV_LOG_DRIVER_STOP	LOG_WARNING	To log driver stops, select the *Log Errors and Warnings* log level, or select the *Log Specific Events* log level and specify this event.

A.13 Log Schema Files

Log Schema (LSC) files catalog the events that can be logged for a given application. They also provide event descriptions and field titles, although this is optional. For information on creating Log Schema files, see the Novell Audit SDK (http://developer.novell.com/ndk/naudit.htm).

A.13.1 How LSC Files Are Used

The information stored in the log schema files—specifically Event IDs, Group IDs, Text and Numeric field values—is useful in defining query statements, Notification Filters, and Heartbeat Notifications. For example, if you want to receive a notification when Remote Loader stops, you must first look up the Event ID for the Remote Loader Stop event in the dirxml log schema. You can then configure a Notification Filter that selects events with an Event ID of 00030BB9.

For more information on Log Schema files, refer to Log Schema Files (http://www.novell.com/documentation/novellaudit20/novellaudit20/data/alg2t8z.html) in the *Novell Audit 2.0 Administration Guide.*

A.14 XDAS Events

The following table lists the XDAS events that can be audited through Sentinel:

Table A-12 XDAS Events

XDAS Event Name	Event Identifier	Corresponding LSC Event	Description
Notification	0.0.6.0	00030001	Status Success
	0.0.6.0	00030002	Status Retry
	0.0.6.0	000307DE	Notify Job Update
	0.0.6.0	000303E4	Job Result Aborted
	0.0.6.0	000303E5	Job Result Error
	0.0.6.0	000303E6	Job Result Warning
	0.0.6.0	000303E7	Job Result Success
	0.0.6.0	00030003	Status Warning

XDAS Event Name	Event Identifier	Corresponding LSC Event	Description
	0.0.6.0	00030004	Status Error
	0.0.6.0	00030005	Status Fatal
	0.0.6.0	00030006	Status Other
	0.0.6.0	00030026	DirXML Error
	0.0.6.0	00030027	DirXML Warning
	0.0.6.0	00030028	Custom Operation
Data Item and Resource Element Management Event			
	0.0.2.0	000307DD	Initialize Driver Object
	0.0.2.0	000307D1	Config:Driver Cache Limit
	0.0.2.0	000307D2	Config:Driver Set
	0.0.2.0	000307D0	Config:Log Events
	0.0.2.0	000307D3	Config:Driver Start Option
	0.0.2.0	00030008	Add Entry
	0.0.2.0	0003002F	Add Value - Add Entry
Modify Data Item	0.0.2.3	0003002E	Reset Attributes
	0.0.2.3	0003002A	Add Value - Modify Entry
Delete Data Item	0.0.2.1	0003002B	Remove Value
	0.0.2.1	00030029	Clear Attribute
	0.0.2.1	00030009	Delete Entry
Query Data Item	0.0.2.2	000307DB	Cache Utility
	0.0.2.2	00030007	Search
	0.0.2.2	0003000F	Query Schema
Modify Data Item	0.0.2.3	0003000A	Modify Entry
	0.0.2.3	0003000B	Rename Entry
	0.0.2.3	0003002C	Merge Entries
	0.0.2.3	0003000C	Move Entry
Peer Association Management Events			
Create Peer Association	0.0.5.0	0003000D	Add Association
Terminate Peer Association	0.0.5.2	0003000E	Remove Association
Modify Association Context	0.0.5.3	00030020	Resync Driver

XDAS Event Name	Event Identifier	Corresponding LSC Event	Description
	0.0.5.3	000307D5	Migrate Application
	0.0.5.3	000307D4	Driver Resync
Service or Application Utilization Events			
Invoke Service	0.0.4.0	000307DF	Open Driver Action
	0.0.4.0	000307E0	Queue Driver Event
	0.0.4.0	00030014	Input XML Document
	0.0.4.0	00030015	Input Transformation Document
	0.0.4.0	00030016	Output Transformation Document
	0.0.4.0	00030017	Event Transformation Document
	0.0.4.0	00030018	Placement Rule Transformation Document
	0.0.4.0	00030019	Create Rule Transformation Document
	0.0.4.0	0003001A	Input Mapping Rule Transformation Document
	0.0.4.0	0003001B	Output Mapping Rule Transformation Document
	0.0.4.0	0003001C	Matching Rule Transformation Document
	0.0.4.0	0003001D	Command Transformation Document
	0.0.4.0	0003001E	Publisher Filter Transformation Document
	0.0.4.0	0003001F	User Agent Request
	0.0.5.3	00030021	Migrate
Service or Application Management Events			
Enable Service	0.0.3.5	00030022	Driver Start
	0.0.3.5	000307E1	Start Job
	0.0.3.5	00030BB8	Remote Loader Start
Disable Service	0.0.3.4	000307E2	Abort Job
	0.0.3.4	00030023	Driver Stop
	0.0.3.4	00030BB9	Remote Loader Stop

XDAS Event Name	Event Identifier	Corresponding LSC Event	Description
Account Management Events			
Query Account	0.0.0.4	00030010	Check Password
	0.0.0.4	00030011	Check Object Password
	0.0.5.3	00030013	Sync
	0.0.0.4	0003002D	Get Named Password
	0.0.0.4	000307DA	Get Server Certificate
	0.0.0.4	000307DC	Check Object Password
	0.0.0.6	00030012	Change Password
Modify Account Security Token	0.0.0.6	00030024	Password Sync
	0.0.0.6	00030025	Password Reset
	0.0.0.6	00030030	Set SSO Credential
	0.0.0.6	00030031	Clear SSO Credential
	0.0.0.6	00030032	Set SSO Passphase
	0.0.0.6	000307D6	Shim Password Set
	0.0.0.6	000307D7	Keyed Password Set
	0.0.0.6	000307D8	Remote Loader Password Set
	0.0.0.6	000307D9	Regenerate Key Pair
Session Management Events			
Create Session	0.0.1.0	00030BBA	Remote Loader Connection Established
Terminate Session	0.0.1.1	00030BBB	Remote Loader Connection Dropped