Administration Guide

XDASv2 for eDirectory, Identity Manager, and NMAS V1

April 2013

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.

Copyright © 2013 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

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 5

1 Overview 7

1.1 Key Benefits . . . 7
1.2 XDASv2 Server Architecture . . . 7

2 Configuring XDASv2 9

2.1 Installing eDirectory and XDASv2 . . . 9
2.1.1 XDASv2 Files Installed with eDirectory . . . 9
2.1.2 Installing XDASv2 on OES Platforms . . . 9
2.2 Configuring the XDASv2 Property File . . . 10
2.3 Configuring XDAS Events . . . 14
2.4 Loading and Unloading the Modules . . . 15
2.5 Enabling XDAS Event Caching . . . 15

3 Configuring XDASv2 in eDirectory 8.8.7 Patch 3 and Later Versions 17

3.1 Installing eDirectory and XDASv2 . . . 17
3.1.1 XDASv2 Files Installed with eDirectory . . . 17
3.1.2 Installing XDASv2 on OES Platforms . . . 18
3.2 Upgrading iManager Plugins For XDASv2 . . . 19
3.3 Configuring the XDASv2 Property File . . . 19
3.4 Configuring XDAS Events . . . 23
3.5 Loading and Unloading the Modules . . . 24
3.6 Enabling XDAS Event Caching . . . 24

4 iManager Plug-In for XDASv2 25

4.1 System Requirements . . . 25
4.2 Installing the iManager Plug-In for XDASv2 . . . 25
4.3 Using the iManager Plug-In Console for XDASv2 . . . 25
4.4 Configuring XDASv2 Events for Auditing . . . 26
4.4.1 Configuring Events . . . 26
4.4.2 Configuring XDASv2 Roles . . . 28
4.4.3 Configuring XDASv2 Accounts . . . 29

5 iManager Plug-In for XDASv2 in eDirectory 8.8.7 Patch 3 and Later Versions 31

5.1 System Requirements . . . 31
5.2 Installing the iManager Plug-In for XDASv2 . . . 31
5.3 Using the iManager Plug-In Console for XDASv2 . . . 31
5.4 Configuring XDASv2 Events for Auditing . . . 32
5.4.1 Configuring Events . . . 32
5.4.2 Configuring Filters and Event Notifications . . . 34
5.4.3 Configuring XDASv2 Roles . . . 34
5.4.4 Configuring XDASv2 Accounts . . . 35

6 Troubleshooting **37**

A XDASv2 Events **39**

A.1 Account Management Events . . . 39
A.2 Session Management Events . . . 40
A.3 Data Item and Resource Element Management Events . . . 41
A.4 Service or Application Management Events . . . 42
A.5 Service or Application Utilization Events . . . 43
A.6 Peer Association Management Events . . . 43
A.7 Data Item or Resource Element Content Access Events . . . 44
A.8 Work Flow Management Events . . . 45
A.9 Role Management Events . . . 45
A.10 Exceptional Events . . . 46
A.11 Audit Service Management Events . . . 47
A.12 Authentication Event . . . 48

B XDASv2 Schema **51**

B.1 XDAS V2 JSON Schema . . . 51
B.2 XDAS Field Definitions . . . 54
B.3 Outcome Codes . . . 56
B.4 Example of an Event . . . 56

About This Guide

This guide describes how to configure and use XDASv2 to audit Novell eDirectory 8.8 and Novell Identity Manager.

- Chapter 1, “Overview,” on page 7
- Chapter 2, “Configuring XDASv2,” on page 9
- Chapter 3, “Configuring XDASv2 in eDirectory 8.8.7 Patch 3 and Later Versions,” on page 17
- Chapter 4, “iManager Plug-In for XDASv2,” on page 25
- Chapter 5, “iManager Plug-In for XDASv2 in eDirectory 8.8.7 Patch 3 and Later Versions,” on page 31
- Chapter 6, “Troubleshooting,” on page 37
- Appendix A, “XDASv2 Events,” on page 39
- Appendix B, “XDASv2 Schema,” on page 51

Audience

This guide is intended for 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 and enter your comments there.

Documentation Updates

For the most recent version of the *Novell XDASv2 Administration Guide*, see the Novell eDirectory online documentation (http://www.novell.com/documentation/edir88/) Web site.

Additional Documentation

For documentation on eDirectory documentation, see the following:

- *Novell eDirectory 8.8 SP7 Administration Guide* (http://www.novell.com/documentation/edir88/edir88/data/a2iii88.html).
- *Novell eDirectory 8.8 SP7 Installation Guide* (http://www.novell.com/documentation/edir88/edirin88/data/a2iii88.html).
- *Novell iManager 2.7 Administration Guide* (http://www.novell.com/documentation/imanager27/index.html).

1 Overview

The XDASv2 specification provides a standardized classification for audit events. It defines a set of generic events at a global distributed system level. XDASv2 provides a common portable audit record format to facilitate the merging and analysis of audit information from multiple components at the distributed system level. The XDASv2 events are encapsulated within a hierarchical notational system that helps to extend the standard or existing event identifier set.The XDASv2 taxonomy defines a set of fields, of these the primary fields are observer, initiator and target. XDASv2 events helps you easily understand the audit trails of heterogeneous applications

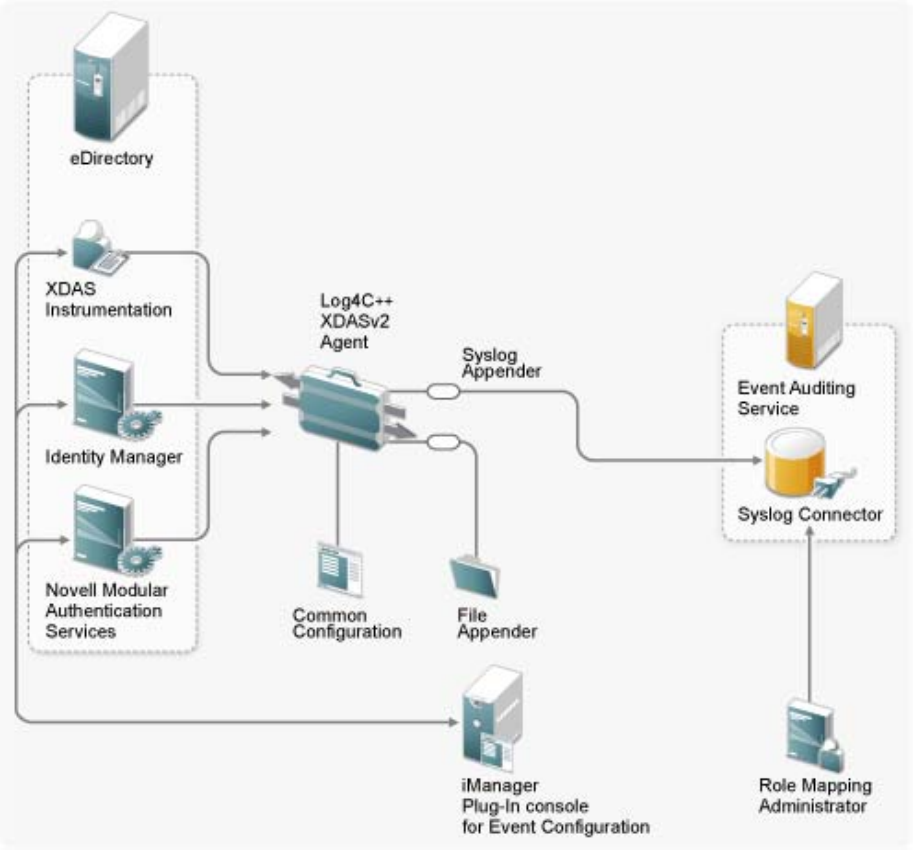
- Section 1.1, “Key Benefits,” on page 7
- Section 1.2, “XDASv2 Server Architecture,” on page 7

1.1 Key Benefits

- Provides secured audit services for a distributed system.
- Defines a set of generic events at a global distributed system level.
- Defines a common portable audit record format to help merge and analyze the audit information from multiple components of a distributed system.
- Defines a common format for audit events that analysis applications can use.
- Records XDASv2 audit trail.
- Configures event preselection criteria and event disposition actions.
- Provides a common audit format regardless of the platform on which the XDASv2 service is running.
- Supports heterogeneous environments without the necessity to re-engineer the current operating system or application-specific audit service implementations.
- Supports adequate separation of duties for users.
- Protects the audit log by making it accessible only to principals acting in specific administrative or security roles.
- Optionally caches audit events locally on the agent in case of communication failure between the agent and the auditing server and re-sends events when communication is re-established.

1.2 XDASv2 Server Architecture

Figure 1-1 *XDASv2 Server Architecture*



2 Configuring XDASv2

This chapter contains the following information:

- Section 2.1, “Installing eDirectory and XDASv2,” on page 9
- Section 2.2, “Configuring the XDASv2 Property File,” on page 10
- Section 2.3, “Configuring XDAS Events,” on page 14
- Section 2.4, “Loading and Unloading the Modules,” on page 15
- Section 2.5, “Enabling XDAS Event Caching,” on page 15

2.1 Installing eDirectory and XDASv2

2.1.1 XDASv2 Files Installed with eDirectory

The following eDirectory XDASv2 files are, by default, installed as part of eDirectory.

- Linux:
  - `novell-edirectory-xdaslog`
  - `novell-edirectory-xdaslog-conf`
  - `novell-edirectory-xdasinstrument`
- Solaris:
  - `NOVLlog`
  - `NOVLedirxdasin`
- Windows
  - `xdasauditds.dlm`
  - `xdaslog.dll`

2.1.2 Installing XDASv2 on OES Platforms

1. To install XDAS RPMs, download eDirectory 8.8 SP7 from the Novell Downloads Web site (http://download.novell.com/).
2. Extract the downloaded tar file and install the following RPMs from the `<eDirectory build extracted folder>/eDirectory/setup` folder.

   For 32-bit RPMs:

```
#rpm -Uvh <eDirectory build extracted folder>/eDirectory/setup/novell-edirectory-xdasinstrument-8.8.7-1.i586.rpm --nodeps

#rpm -Uvh <eDirectory build extracted folder>/eDirectory/setup/novell-edirectory-xdaslog-8.8.7-1.i586.rpm --nodeps
```

```
#rpm -Uvh <eDirectory build extracted folder>/eDirectory/setup/novell-edirectory-xdaslog-conf-8.8.7-1.noarch.rpm --nodeps
```

For 64-bit RPMs:

```
#rpm -Uvh <eDirectory build extracted folder>/eDirectory/setup/novell-edirectory-xdasinstrument-8.8.7-1.x86_64.rpm --nodeps

#rpm -Uvh <eDirectory build extracted folder>/eDirectory/setup/novell-edirectory-xdaslog-8.8.7-1.x86_64.rpm --nodeps

#rpm -Uvh <eDirectory build extracted folder>/eDirectory/setup/novell-edirectory-xdaslog-conf-8.8.7-1.noarch.rpm --nodeps
```

3. Rename the `xdasconfig.properties.template` file to `xdasconfig.properties`.

   The `xdasconfig.properties.template` file is automatically created when the XDAS RPM is installed.

4. Create symbolic links for expat libraries.

   For 32-bit RPMs:

   - `ln -s /usr/lib/libexpat.so.1.5.0 /opt/novell/eDirectory/lib/libexpat.so.0.1.0`
   - `ln -s /opt/novell/eDirectory/lib/libexpat.so.0.1.0 /opt/novell/eDirectory/lib/libexpat.so.0`
   - `ln -s /opt/novell/eDirectory/lib/libexpat.so.0.1.0 /opt/novell/eDirectory/lib/libexpat.so`

   For 64-bit RPMs:

   - `ln -s /lib64/libexpat.so.1.5.0 /opt/novell/eDirectory/lib64/libexpat.so.0.1.0`
   - `ln -s /opt/novell/eDirectory/lib64/libexpat.so.0.1.0 /opt/novell/eDirectory/lib64/libexpat.so.0`
   - `ln -s /opt/novell/eDirectory/lib64/libexpat.so.0.1.0 /opt/novell/eDirectory/lib64/libexpat.so`

2.2 Configuring the XDASv2 Property File

The XDASv2 property file is located at `/etc/opt/novell/eDirectory/conf/xdasconfig.properties` by default. For non-root installations, the XDASv2 property file is located in the `conf` directory. You can customize the file according to your requirements.

NOTE: When you install eDirectory, the installer creates the `xdasconfig.properties.template` file. To configure the property file, first rename `xdasconfig.properties.template` to `xdasconfig.properties`.

If you configure the property file and then upgrade your environment to eDirectory 8.8 SP7, the installer does not replace the existing `xdasconfig.properties` file. Instead, the installer updates the `xdasconfig.properties.template` file, in order to retain any customizations in the `xdasconfig.properties` file itself.

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 2-1 *XDAS Configuration File*

Operating System	File
Linux/Solaris	`/etc/opt/novell/eDirectory/conf/xdasconfig.properties`
Windows	`<Install Path>/novell/nds/xdasconfig` The property file is usually in the eDirectory installation directory.

The following is the content of the XDASv2 property file:

```
# Set the level of the root logger to DEBUG and attach appenders.
#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

# Defines caching for SyslogAppender.
# Inputs should be yes/no
#log4j.appender.S.CacheEnabled=no

# Cache location directory
# Directory should be available for creating cache files
#log4j.appender.S.CacheDir=/var/opt/novell/eDirectory

# Cache File Size
# Cache File Size should be in the range of 50MB to 4000MB
#log4j.appender.S.CacheMaxFileSize=500MB

# 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 2-2 *XDASv2 Property File*

Options	ID
Syslog Appender	S
Rolling File Appender	R

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 2-3 *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.
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.
log4j.appender.S.Port=*port*	The port at which the XDAS connects to the Syslog server. The port supports values from `1` to `65535`. If you specify an invalid value, the port defaults to `514`. 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. Currently, the INFO log level is supported.

Setting	Description
log4j.appender.S.Facility=USER	Specifies the type of facility. The facility is used to try to classify the message.Currently, USER facility is supported. These values may be specified as upper or lower case characters.
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. For information about the conversion patters and their descriptions, see logging.apache.org.
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. See Table 2-4 on page 13 for simple date format patterns. For information about the conversion patters and their descriptions, see logging.apache.org

The following examples illustrate the date and time patterns interpreted in the U.S. The given date and time are 2012-07-04 12:08:56 local time in the U.S. Pacific Time time zone.

Table 2-4 *Date and Time Pattern Example*

Date and Time Pattern	Result
"yyyy.MM.dd G 'at' HH:mm:ss z"	2012.07.04 AD at 12:08:56 PDT
"EEE, MMM d, "yy"	Wed, Jul 4, '01
"h:mm a"	12:08 PM
"hh 'o''clock' a, zzzz"	12 o'clock PM, Pacific Daylight Time
"K:mm a, z"	0:08 PM, PDT
"yyyyy.MMMMM.dd GGG hh:mm aaa"	02012.July.24 AD 12:08 PM
"EEE, d MMM yyyy HH:mm:ss Z"	Wed, 24 Jul 2012 12:08:56 -0700
"yyMMddHHmmssZ"	120724120856-0700
"yyyy-MM-dd'T'HH:mm:ss.SSSZ"	2012-07-04T12:08:56.235-0700

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 information about configuring XDAS Events, see Section 2.3, "Configuring XDAS Events," on page 14.

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. Select the desired event from iManager.

   For information about configuring XDAS Events, see Section 2.3, "Configuring XDAS Events," on page 14.

2.3 Configuring XDAS Events

For information on configuring XDASv2 events for eDirectory, see Chapter 4, "iManager Plug-In for XDASv2," on page 25 .

NOTE: After you select the event, it takes up to 3 minutes for the configuration changes to take effect on the NCP Server. If you want the configuration changes to be implemented immediately on the NCP server, you can unload and load the xdasauditds module.

2.4 Loading and Unloading the Modules

After you have configured the XDASv2 events, run the following commands to load and unload the XDASv2 modules:

- **Linux/Solaris** : Run the following command to load the XDASv2 modules:

  ```
  ndstrace -c "load xdasauditds"
  ```

  Run the following command to unload the XDASv2 modules:

  ```
  ndstrace -c "unload xdasauditds"
  ```

- **Windows:** To load the module, run `ndscons.exe`, select the *xdasauditds* option from the list of available modules, then click *Start*.

  To unload the module, run `ndscons.exe`, select the *xdasauditds* option from the list of available modules, then click *Stop*.

If you have installed Novell Modular Authentication Service (NMAS) and enabled NMAS auditing, the NMAS server automatically loads the XDASv2 library.

2.5 Enabling XDAS Event Caching

eDirectory 8.8 SP7 allows you to optionally store XDAS events locally on the agent in a Syslog Appender cache. With events cached, if the agent cannot communicate with the auditing server, the audit events generated are retained, ensuring that audit data is not lost. The agent then attempts to re-send the cached events when the agent computer can once again communicate with the auditing server.

XDAS event caching is disabled by default. To enable event caching, complete the steps below.

1. On the agent computer, navigate to the location of the XDASv2 property file. The xdasconfig.properties file is located at `/etc/opt/novell/eDirectory/conf/xdasconfig.properties` by default. For non-root installations, the XDASv2 property file is located in the `conf` directory by default.
2. Use a text editor to open the `xdasconfig.properties` file.
3. Within the property file, navigate to the `log4j.appender.S.CacheEnabled` property and change the property value to `yes`.
4. If you want to cache events in a specific directory, modify the value of the `log4j.appender.S.CacheDir` property to specify the directory path. The default path is `/var/opt/novell/eDirectory`. If you specify a directory, ensure that the directory path is a valid location on the server. If the specified path does not exist, the Syslog Appender logs events to the default location.
5. If you want to specify a custom file size for the cache, modify the value of the `log4j.appender.S.CacheMaxFileSize` property. The default value is 500 MB. The minimum value should be 50 MB, with a maximum value of 4 GB.
6. Save and close the `xdasconfig.properties` file.

3 Configuring XDASv2 in eDirectory 8.8.7 Patch 3 and Later Versions

This chapter contains the following information:

- Section 3.1, “Installing eDirectory and XDASv2,” on page 17
- Section 3.2, “Upgrading iManager Plugins For XDASv2,” on page 19
- Section 3.3, “Configuring the XDASv2 Property File,” on page 19
- Section 3.4, “Configuring XDAS Events,” on page 23
- Section 3.5, “Loading and Unloading the Modules,” on page 24
- Section 3.6, “Enabling XDAS Event Caching,” on page 24

3.1 Installing eDirectory and XDASv2

3.1.1 XDASv2 Files Installed with eDirectory

The following eDirectory XDASv2 files are, by default, installed as part of eDirectory.

- Linux:
  - `novell-edirectory-xdaslog`
  - `novell-edirectory-xdaslog-conf`
  - `novell-edirectory-xdasinstrument`
- Solaris:
  - `NOVLlog`
  - `NOVLedirxdasin`
- Windows
  - `xdasauditds.dlm`
  - `xdaslog.dll`

NOTE: Extend the schema file `edirxdas.sch` after installing the patch.

- Schema Extension on Linux, Solaris, or AIX Systems:

  For information about extending the schema on Linux, Solaris, or AIX Systems, see “Extending the Schema on Linux, Solaris, or AIX Systems” in Novell eDirectory 8.8 SP7 Administration Guide.
- Schema Extension on Windows:

  For information about extending the schema on Windows, see “Extending the Schema on Windows” in Novell eDirectory 8.8 SP7 Administration Guide.

3.1.2 Installing XDASv2 on OES Platforms

1. To install XDAS RPMs, download eDirectory 8.8 SP7 from the Novell Downloads Web site (http://download.novell.com/).
2. Extract the downloaded tar file and install the following RPMs from the *<eDirectory build extracted folder>*/eDirectory/setup folder.

   For 32-bit RPMs:

```
#rpm -Uvh <eDirectory build extracted folder>/eDirectory/setup/novell-edirectory-xdasinstrument-8.8.7-1.i586.rpm --nodeps

#rpm -Uvh <eDirectory build extracted folder>/eDirectory/setup/novell-edirectory-xdaslog-8.8.7-1.i586.rpm --nodeps

#rpm -Uvh <eDirectory build extracted folder>/eDirectory/setup/novell-edirectory-xdaslog-conf-8.8.7-1.noarch.rpm --nodeps
```

   For 64-bit RPMs:

```
#rpm -Uvh <eDirectory build extracted folder>/eDirectory/setup/novell-edirectory-xdasinstrument-8.8.7-1.x86_64.rpm --nodeps

#rpm -Uvh <eDirectory build extracted folder>/eDirectory/setup/novell-edirectory-xdaslog-8.8.7-1.x86_64.rpm --nodeps

#rpm -Uvh <eDirectory build extracted folder>/eDirectory/setup/novell-edirectory-xdaslog-conf-8.8.7-1.noarch.rpm --nodeps
```

3. Rename the `xdasconfig.properties.template` file to `xdasconfig.properties`.

   The `xdasconfig.properties.template` file is automatically created when the XDAS RPM is installed.

4. Create symbolic links for expat libraries.

   For 32-bit RPMs:

   - `ln -s /usr/lib/libexpat.so.1.5.0 /opt/novell/eDirectory/lib/libexpat.so.0.1.0`
   - `ln -s /opt/novell/eDirectory/lib/libexpat.so.0.1.0 /opt/novell/eDirectory/lib/libexpat.so.0`
   - `ln -s /opt/novell/eDirectory/lib/libexpat.so.0.1.0 /opt/novell/eDirectory/lib/libexpat.so`

   For 64-bit RPMs:

   - `ln -s /lib64/libexpat.so.1.5.0 /opt/novell/eDirectory/lib64/libexpat.so.0.1.0`
   - `ln -s /opt/novell/eDirectory/lib64/libexpat.so.0.1.0 /opt/novell/eDirectory/lib64/libexpat.so.0`
   - `ln -s /opt/novell/eDirectory/lib64/libexpat.so.0.1.0 /opt/novell/eDirectory/lib64/libexpat.so`

3.2 Upgrading iManager Plugins For XDASv2

You can upgrade iManager Audit plugins to latest version only if you have installed eDirectory 8.8 SP7 patch 3 or later version.

1. Log in to the iManager console.
   - **1a** Open iManager from a Web browser, using the following URL:

     `https://ip_address_or_DNS/nps/iManager.html`

     where *ip_address_or_DNS* is the IP address or DNS name of your iManager server.

     For example:

     `http://192.168.0.5/nps/iManager.html`
   - **1b** Log in using your username and password.

     In iManager, you have access only to those roles for which you have assigned rights. To have full access to all Novell iManager features, you must log in as a user with Admin rights to the tree.

     For more information, see "Accessing iManager" in the *Novell iManager 2.7 Administration Guide* (http://www.novell.com/documentation/imanager27/imanager_admin_275/data/bsxrjzp.html)
2. Select *Audit Configuration* from *Roles and Tasks*.
3. Click the *Upgrade XDAS Configuration* link.

   An alert message about the upgrade process is displayed.
4. Click Ok.

   During upgrade, new iManager files are installed and they cause configuration changes. After the upgrade completes, a message is displayed stating the success or failure status of the installation.

3.3 Configuring the XDASv2 Property File

The XDASv2 property file is located at `/etc/opt/novell/eDirectory/conf/xdasconfig.properties` by default. For non-root installations, the XDASv2 property file is located in the `conf` directory. You can customize the file according to your requirements.

NOTE: When you install eDirectory, the installer creates the `xdasconfig.properties.template` file. To configure the property file, first rename `xdasconfig.properties.template` to `xdasconfig.properties`.

If you configure the property file and then upgrade your environment to eDirectory 8.8 SP7, the installer does not replace the existing `xdasconfig.properties` file. Instead, the installer updates the `xdasconfig.properties.template` file, in order to retain any customizations in the `xdasconfig.properties` file itself.

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 3-1 *XDAS Configuration File*

Operating System	File
Linux/Solaris	`/etc/opt/novell/eDirectory/conf/xdasconfig.properties`
Windows	`<Install Path>/novell/nds/xdasconfig` The property file is usually in the eDirectory installation directory.

The following is the content of the XDASv2 property file:

```
# Set the level of the root logger to DEBUG and attach appenders.
#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

# Defines caching for SyslogAppender.
# Inputs should be yes/no
#log4j.appender.S.CacheEnabled=no

# Cache location directory
# Directory should be available for creating cache files
#log4j.appender.S.CacheDir=/var/opt/novell/eDirectory

# Cache File Size
# Cache File Size should be in the range of 50MB to 4000MB
#log4j.appender.S.CacheMaxFileSize=500MB

# 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 3-2 *XDASv2 Property File*

Options	ID
Syslog Appender	S
Rolling File Appender	R

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 3-3 *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.
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. IFor example, log4j.appender.S.Host=192.168.0.1
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. Currently, the INFO log level is supported.
log4j.appender.S.Facility=USER	Specifies the type of facility. The facility is used to try to classify the message.Currently, USER facility is supported. These values may be specified as upper or lower case characters.

Setting	Description
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. For information about the conversion patters and their descriptions, see logging.apache.org.
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. See Table 3-4 on page 22 for simple date format patterns. For information about the conversion patters and their descriptions, see logging.apache.org

The following examples illustrate the date and time patterns interpreted in the U.S. The given date and time are 2012-07-04 12:08:56 local time in the U.S. Pacific Time time zone.

Table 3-4 *Date and Time Pattern Example*

Date and Time Pattern	Result
"yyyy.MM.dd G 'at' HH:mm:ss z"	2012.07.04 AD at 12:08:56 PDT
"EEE, MMM d, "yy"	Wed, Jul 4, '01
"h:mm a"	12:08 PM
"hh 'o''clock' a, zzzz"	12 o'clock PM, Pacific Daylight Time
"K:mm a, z"	0:08 PM, PDT
"yyyyy.MMMMM.dd GGG hh:mm aaa"	02012.July.24 AD 12:08 PM
"EEE, d MMM yyyy HH:mm:ss Z"	Wed, 24 Jul 2012 12:08:56 -0700
"yyMMddHHmmssZ"	120724120856-0700
"yyyy-MM-dd'T'HH:mm:ss.SSSZ"	2012-07-04T12:08:56.235-0700

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 information about configuring XDAS Events, see Section 3.4, "Configuring XDAS Events," on page 23.

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 Select the desired event from iManager.

For information about configuring XDAS Events, see Section 3.4, "Configuring XDAS Events," on page 23.

3.4 Configuring XDAS Events

For information about configuring XDASv2 events for eDirectory, see Chapter 4, "iManager Plug-In for XDASv2," on page 25 Chapter 4, "iManager Plug-In for XDASv2," on page 25.

NOTE: After you select the event, it takes up to 3 minutes for the configuration changes to take effect on the NCP Server. If you want the configuration changes to be implemented immediately on the NCP server, you can unload and load the xdasauditds module.

3.5 Loading and Unloading the Modules

After you have configured the XDASv2 events, run the following commands to load and unload the XDASv2 modules:

- **Linux/Solaris** : Run the following command to load the XDASv2 modules:

  ```
  ndstrace -c "load xdasauditds"
  ```

  Run the following command to unload the XDASv2 modules:

  ```
  ndstrace -c "unload xdasauditds"
  ```

- **Windows:** To load the module, run `ndscons.exe`, select the *xdasauditds* option from the list of available modules, then click *Start*.

  To unload the module, run `ndscons.exe`, select the *xdasauditds* option from the list of available modules, then click *Stop*.

If you have installed Novell Modular Authentication Service (NMAS) and enabled NMAS auditing, the NMAS server automatically loads the XDASv2 library.

3.6 Enabling XDAS Event Caching

eDirectory 8.8 SP7 allows you to optionally store XDAS events locally on the agent in a Syslog Appender cache. With events cached, if the agent cannot communicate with the auditing server, the audit events generated are retained, ensuring that audit data is not lost. The agent then attempts to re-send the cached events when the agent computer can once again communicate with the auditing server.

XDAS event caching is disabled by default. To enable event caching, complete the steps below.

1. On the agent computer, navigate to the location of the XDASv2 property file. The xdasconfig.properties file is located at `/etc/opt/novell/eDirectory/conf/xdasconfig.properties` by default. For non-root installations, the XDASv2 property file is located in the `conf` directory by default.
2. Use a text editor to open the `xdasconfig.properties` file.
3. Within the property file, navigate to the `log4j.appender.S.CacheEnabled` property and change the property value to `yes`.
4. If you want to cache events in a specific directory, modify the value of the `log4j.appender.S.CacheDir` property to specify the directory path. The default path is `/var/opt/novell/eDirectory`. If you specify a directory, ensure that the directory path is a valid location on the server. If the specified path does not exist, the Syslog Appender logs events to the default location.
5. If you want to specify a custom file size for the cache, modify the value of the `log4j.appender.S.CacheMaxFileSize` property. The default value is 500 MB. The minimum value should be 50 MB, with a maximum value of 4 GB.
6. Save and close the `xdasconfig.properties` file.

4 iManager Plug-In for XDASv2

You can manage and configure eDirectory for XDASv2 auditing by using Novell iManager. Novell iManager is a Web-based application that is used to manage, maintain, and monitor Novell eDirectory through wired and wireless devices.The XDASv2 iManager plug-in is included with the eDirectory installation.

NOTE: iManager plug-in is only for enabling and configuring XDASv2 events for eDirectory. This plug-in does not apply to NMAS and Identity Manager.

- Section 4.1, "System Requirements," on page 25
- Section 4.2, "Installing the iManager Plug-In for XDASv2," on page 25
- Section 4.3, "Using the iManager Plug-In Console for XDASv2," on page 25
- Section 4.4, "Configuring XDASv2 Events for Auditing," on page 26

4.1 System Requirements

Installing and using the Novell Audit iManager Plug-in requires iManager 2.7.4 or later. See Novell iManager Product Page (http://www.novell.com/products/consoles/) for requirements and download instructions.

4.2 Installing the iManager Plug-In for XDASv2

The iManager plug-in for XDASv2 is bundled with eDirectory 8.8 SP6 plug-ins. eDirectory 8.8 SP6 plug-ins can be downloaded from the Novell download site (http://download.novell.com/SummaryFree.jsp?buildid=EKamexBB_F4~).

The installation instructions are available at eDirectory 8.8 Plug-ins for iManager 2.7 download page (http://www.novell.com/documentation/edir88/esd/ii_edir886_iman_27_plugins.html).

4.3 Using the iManager Plug-In Console for XDASv2

1. Log in to the iManager console.
   - **1a** Open iManager from a Web browser, using the following URL:

     `https://ip_address_or_DNS/nps/iManager.html`

     where *ip_address_or_DNS* is the IP address or DNS name of your iManager server.

     For example:

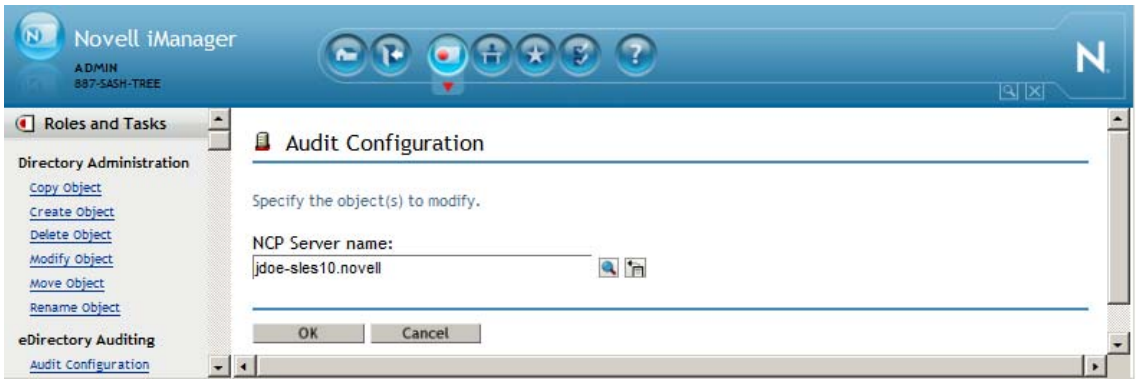
     `http://192.168.0.5/nps/iManager.html`
   - **1b** Log in using your username and password.

In iManager, you have access only to those roles for which you have assigned rights. To have full access to all Novell iManager features, you must log in as a user with Admin rights to the tree.

For more information, see "Accessing iManager" in the *Novell iManager 2.7 Administration Guide* (http://www.novell.com/documentation/imanager27/imanager_admin_275/data/bsxrjzp.html)

2 Select *Audit Configuration* from *Roles and Tasks*.

3 Specify the name of your eDirectory server in *NCP Server*.



Click the *Object Selector* icon to browse for the eDirectory server.

4 Click *OK*.

The XDASv2 Audit page is displayed. Continue with Section 4.4.1, "Configuring Events," on page 26.

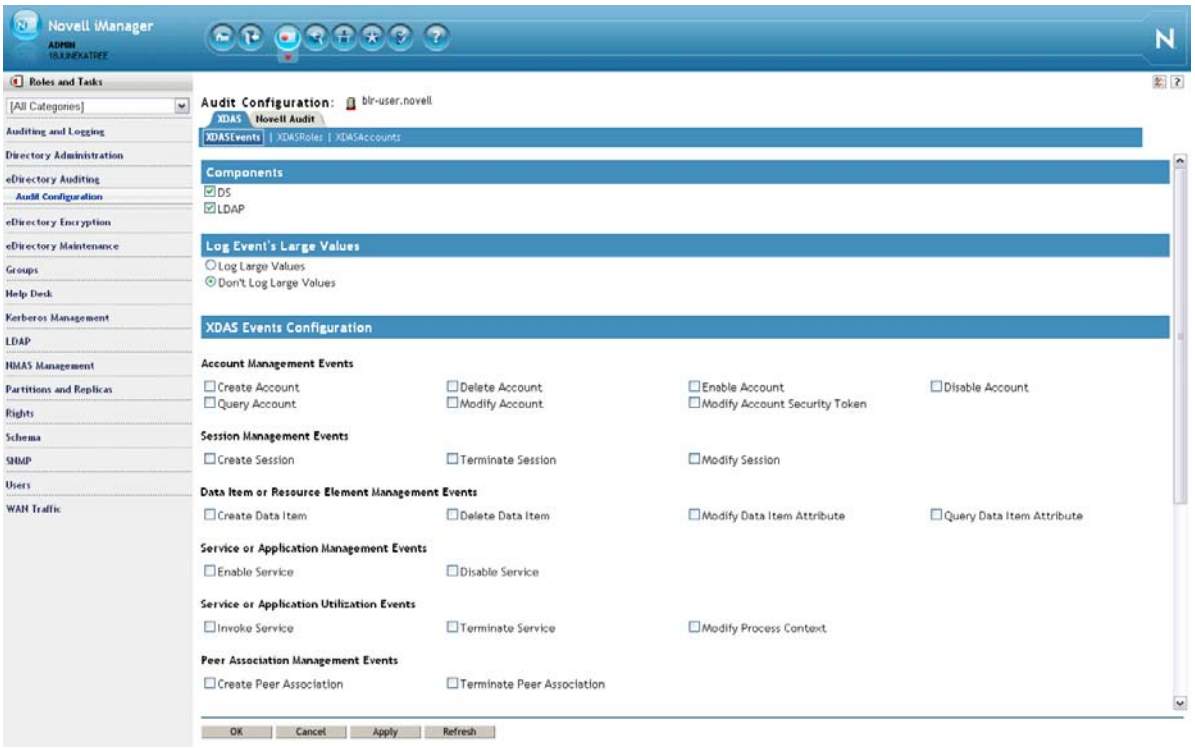
4.4 Configuring XDASv2 Events for Auditing

- Section 4.4.1, "Configuring Events," on page 26
- Section 4.4.2, "Configuring XDASv2 Roles," on page 28
- Section 4.4.3, "Configuring XDASv2 Accounts," on page 29

4.4.1 Configuring Events

Use this page to configure XDASv2 events.

Figure 4-1 *XDASv2 Events*



1 You can select both or either of the following components for XDASv2 event settings:

DS: Specifies an eDirectory™ object. For each DS object, a corresponding LDAP object exists.

LDAP: Specifies an LDAP object.

2 Log event values:

The events are logged into a text file. Event values with more than 768 bytes in size are considered “large values.” You can log events of any size.

Log Large Values: Select this option to log events that are more than 768 bytes in size.

Don’t Log Large Values: Select this option to log events that are less than 768 Byte in size. If the event size is more, the event value is truncated and saved to the log file.

3 Specify the following based on your requirements:

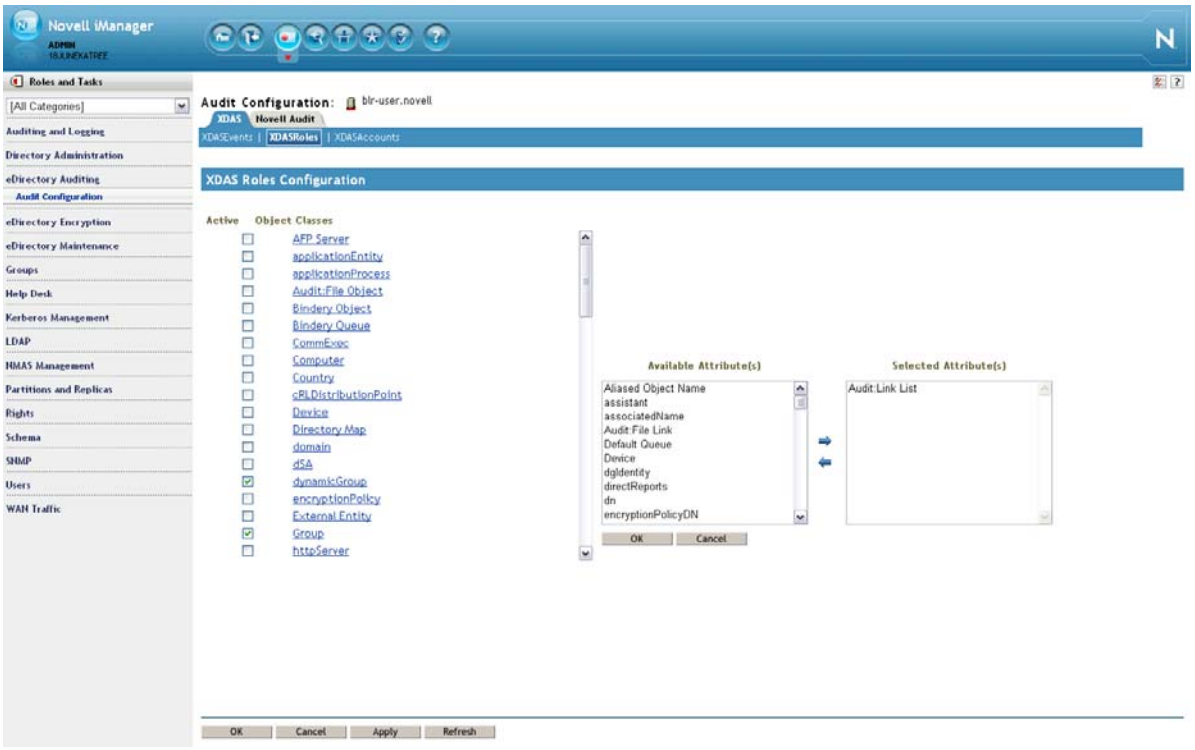
Options	Description
Account Management Events	Select the account management events for which you want to log events. You can log events to create, delete, enable, disable, and query accounts, and also to modify account security token.
Session Management Events	Select the session management events for which you want to log events. You can log events to create, terminate, and modify sessions.
Data Item or Resource Element Management Events	Select the data item or resource element management events for which you want to log events. You can log events to create and delete data items and to modify and query data item attributes.
Service or Application Management Events	Select the service or application management events for which you want to log events. You can log events for enabling and disabling services.
Service or Application Utilization Events	Select the service or application utilization events for which you want to log events. You can log events to start and terminate services, and to modify process contexts.
Peer Association Management Events	Select the peer association events for which you want to log events. You can log events for creating and terminating peer associations.
Data Item or Resource Element Content Access Events	Select the data item or resource element content access events for which you want to log events. You can log events to create, terminate, and modify data item associations.
Role Management Events	Select the role management events for which you want to log events. You can log events to create, delete, query, and modify roles.
Exceptional Management Events	Select the exceptional management events for which you want to log events. You can log events to start and shut down systems and also to back up and recover data stores.
Authentication Management Events	Select the authentication management events for which you want to log events. You can log events to authenticate sessions and create access tokens.
Operational Events	Select the operational management events for which you want to log events. You can log events to generate eDirectory operation IDs.

For more information on events, see Chapter A, "XDASv2 Events," on page 39.

4.4.2 Configuring XDASv2 Roles

Configure XDASv2 roles for the objects for which you want to collect XDASv2 events. You can select object classes and set attributes for them.

Figure 4-2 *XDASv2 Roles*



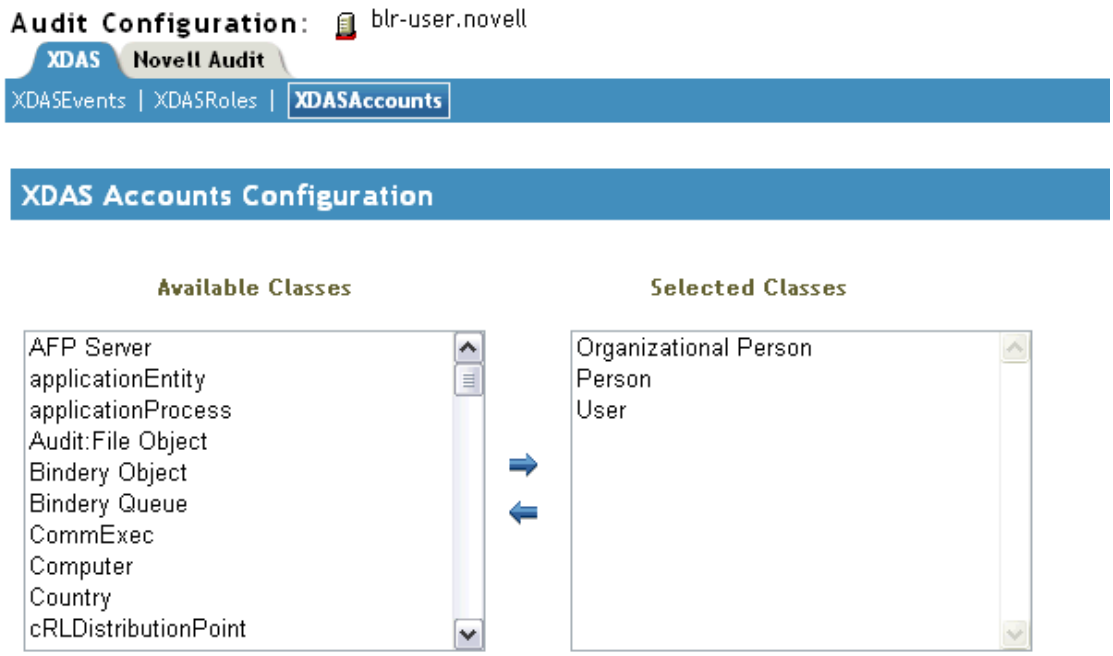
1. Select object classes for which you want to collect events.
2. Set any number of attributes for the object classes you have selected. Click the attribute and click the arrow to add the attribute to the selected list of attributes.
3. Click *OK* after you add the attributes. Click *Apply* to confirm the modifications.

   The selected attributes appear in this list.

4.4.3 Configuring XDASv2 Accounts

Configure XDASv2 accounts by selecting available object classes. Click *OK* to exit the application.

Figure 4-3 *XDASv2 Accounts*



1. Select object classes from the list for which you want to collect events.
2. Selected object classes appear in this list.
3. Click *Apply* after adding the object classes.

5 iManager Plug-In for XDASv2 in eDirectory 8.8.7 Patch 3 and Later Versions

This chapter contains the following information.

- Section 5.1, "System Requirements," on page 31
- Section 5.2, "Installing the iManager Plug-In for XDASv2," on page 31
- Section 5.3, "Using the iManager Plug-In Console for XDASv2," on page 31
- Section 5.4, "Configuring XDASv2 Events for Auditing," on page 32

5.1 System Requirements

Installing and using the Novell Audit iManager Plug-in requires iManager 2.7.4 or later. See Novell iManager Product Page (http://www.novell.com/products/consoles/) for requirements and download instructions.

5.2 Installing the iManager Plug-In for XDASv2

The iManager plug-in for XDASv2 is bundled with eDirectory 8.8 SP6 plug-ins. eDirectory 8.8 SP6 plug-ins can be downloaded from the Novell download site (http://download.novell.com/SummaryFree.jsp?buildid=EKamexBB_F4~).

The installation instructions are available at eDirectory 8.8 Plug-ins for iManager 2.7 download page (http://www.novell.com/documentation/edir88/esd/ii_edir886_iman_27_plugins.html).

5.3 Using the iManager Plug-In Console for XDASv2

1. Log in to the iManager console.
   - **1a** Open iManager from a Web browser, using the following URL:

     `https://ip_address_or_DNS/nps/iManager.html`

     where *ip_address_or_DNS* is the IP address or DNS name of your iManager server.

     For example:

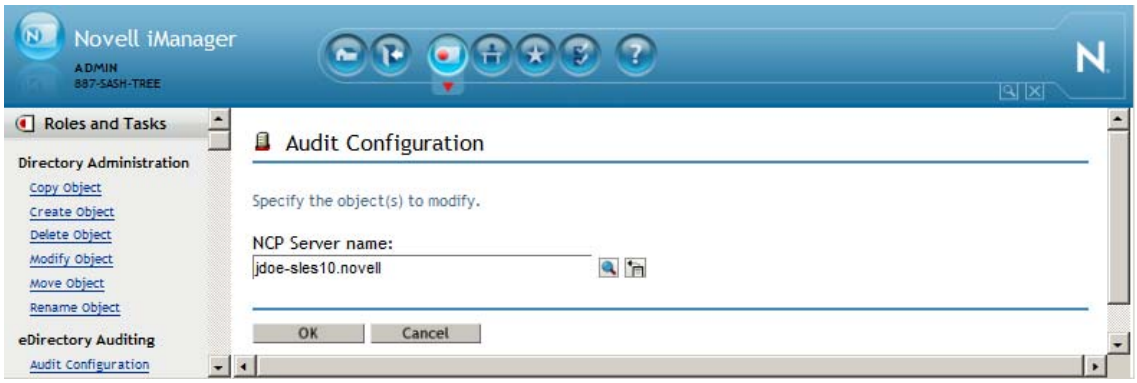
     `http://192.168.0.5/nps/iManager.html`
   - **1b** Log in using your username and password.

In iManager, you have access only to those roles for which you have assigned rights. To have full access to all Novell iManager features, you must log in as a user with Admin rights to the tree.

For more information, see "Accessing iManager" in the *Novell iManager 2.7 Administration Guide* (http://www.novell.com/documentation/imanager27/imanager_admin_275/data/bsxrjzp.html)

2 Select *Audit Configuration* from *Roles and Tasks*.

3 Specify the name of your eDirectory server in *NCP Server*.



Click the *Object Selector* icon to browse for the eDirectory server.

4 Click *OK*.

The XDASv2 Audit page is displayed. Continue with Section 5.4.1, "Configuring Events," on page 32.

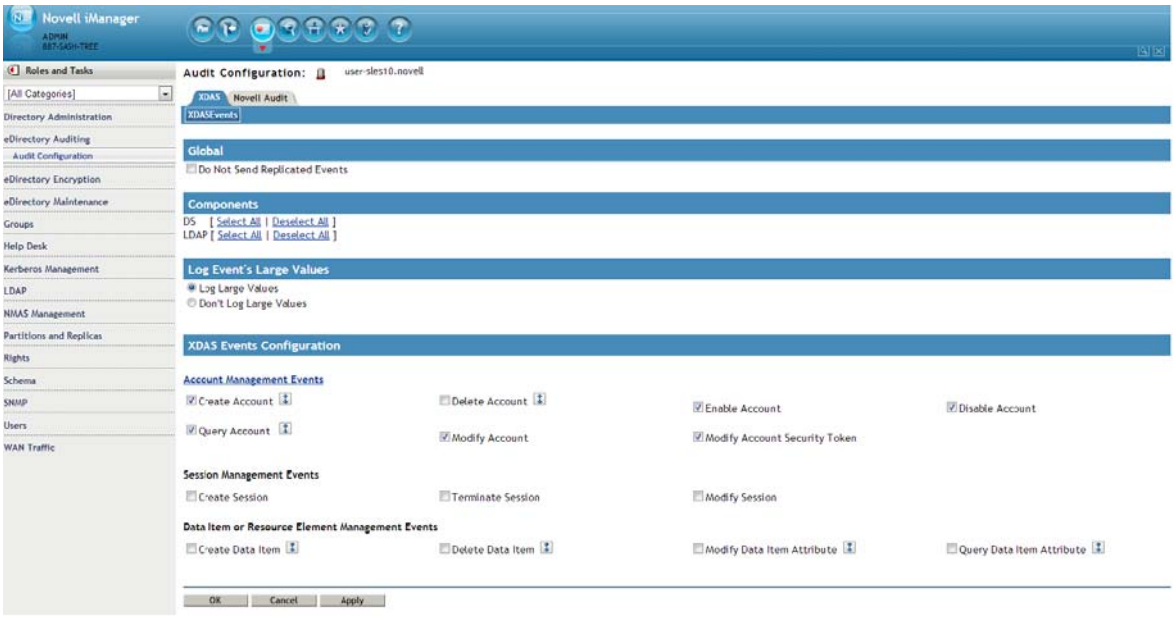
5.4 Configuring XDASv2 Events for Auditing

- Section 5.4.1, "Configuring Events," on page 32
- Section 5.4.2, "Configuring Filters and Event Notifications," on page 34
- Section 5.4.3, "Configuring XDASv2 Roles," on page 34
- Section 5.4.4, "Configuring XDASv2 Accounts," on page 35

5.4.1 Configuring Events

Use this page to configure XDASv2 events.

Figure 5-1 *XDASv2 Events*



1 Global:

You can select or clear the global settings for duplicate entries.

Do Not Send Replicated Events: Select this option to stop receiving duplicate entries, such as logins, for eDirectory.

2 You can select both or either of the following components for XDASv2 event settings:

DS: Specifies an eDirectory™ object. For each DS object, a corresponding LDAP object exists.

LDAP: Specifies an LDAP object.

NOTE: You can select the DS and LDAP components at the granular level for the XDAS events. Based on the event you select, the appropriate components that are supported for that event are selected. For example, if you select the *Delete Account* event, the *DS* and *LDAP* components are selected.

---

3 Log event values:

The events are logged into a text file. Event values with more than 768 bytes in size are considered "large values." You can log events of any size.

Log Large Values: Select this option to log events that are more than 768 bytes in size.

Don't Log Large Values: Select this option to log events that are less than 768 Byte in size. If the event size is more, the event value is truncated and saved to the log file.

4 Specify the following based on your requirements:

Options	Description
Account Management Events	Select the account management events for which you want to log events. You can log events to create, delete, enable, disable, and query accounts, and also to modify account security token.
Session Management Events	Select the session management events for which you want to log events. You can log events to create, terminate, and modify sessions.

Options	Description
Data Item or Resource Element Management Events	Select the data item or resource element management events for which you want to log events. You can log events to create and delete data items and to modify and query data item attributes.
Service or Application Management Events	Select the service or application management events for which you want to log events. You can log events for enabling and disabling services.
Service or Application Utilization Events	Select the service or application utilization events for which you want to log events. You can log events to start and terminate services, and to modify process contexts.
Peer Association Management Events	Select the peer association events for which you want to log events. You can log events for creating and terminating peer associations.
Data Item or Resource Element Content Access Events	Select the data item or resource element content access events for which you want to log events. You can log events to create, terminate, and modify data item associations.
Role Management Events	Select the role management events for which you want to log events. You can log events to create, delete, query, and modify roles.
Exceptional Management Events	Select the exceptional management events for which you want to log events. You can log events to start and shut down systems and also to back up and recover data stores.
Authentication Management Events	Select the authentication management events for which you want to log events. You can log events to authenticate sessions and create access tokens.
Operational Events	Select the operational management events for which you want to log events. You can log events to generate eDirectory operation IDs.

5.4.2 Configuring Filters and Event Notifications

Using filters and event notifications, XDASv2 is capable of reporting when a specific type of event occurs, or when it does not occur. This section provides the information you need to configure your system filters and notifications. You can configure filters and events notifications for XDASv2 Roles and XDASv2 Accounts.

5.4.3 Configuring XDASv2 Roles

Click the *Role Management Events* link to configure the XDASv2 roles. Configure XDASv2 roles for the objects for which you want to collect XDASv2 events. You can select object classes and set attributes for them.

1. Select object classes for which you want to collect events.
2. Set any number of attributes for the object classes you have selected. Click the attribute and click the arrow to add the attribute to the selected list of attributes.
3. Click *OK* after you add the attributes. Click *Apply* to confirm the modifications.

   The selected attributes appear in this list.

5.4.4 Configuring XDASv2 Accounts

Click the *Account Management Events* link to configure the XDASv2 accounts. Configure XDASv2 accounts by selecting available object classes. Click *OK* to exit the application.

1. Select object classes from the list for which you want to collect events.
2. Selected object classes appear in this list.
3. Click *Apply* after adding the object classes.

6 Troubleshooting

Keep in mind the following information when you install Novell XDASv2:

Initializing XDAS module error

Possible Cause: You cannot connect to the server IP or the port number mentioned in `xdasconfig.properties` file when you initialize the XDASv2 module. It displays the following message:

```
log4cxx: Could not instantiate TCP Socket to <IP>. All logging
will FAIL.

log4cxx: IO Exception : status code = 111
```

Action: To work around this issue,

1. Check whether the sever IP or the port number given in the `xdasconfig.properties` file is correct.
2. Check whether the remote server is reachable and is accepting the connection on the given port.
3. Reload the xdasauditds module.

The TCP connection is lost

Possible Cause: If the remote server is not reachable or does not accept connection on the given port, the following error is displayed:

```
log4cxx: Detected problem with TCP connection to <IP>. All logging
will FAIL.

log4cxx: IO Exception : status code = 32
```

Action: To work around this issue:

1. Check whether remote server is reachable and is accepting the connection on the given port.
2. Reload the xdasauditds module.

The SSL certificate file issue

Possible Cause: The SSL certificate file is either not valid or not present at the given location in the `xdasconfig.properties` file. The following error is displayed:

```
log4cxx: could not load verify locations for SSL
```

Action: To work around this issue,

1. Specify the absolute path to a valid certificate file.
2. Reload the xdasauditds module.

The network connection to the remote server is lost

Source: The following error is displayed:

```
log4cxx: SSL write failed for <IP>. All logging will FAIL.
```

Action: To work around this issue,

1. Check whether remote server is reachable and is accepting the connection on the given port.
2. Reload the xdasauditds module.

The SSL connection has failed

Possible Cause: The SSL connection fails because either the TLS/SSL handshake fails or a connection failure occurs. The following error message is displayed:

```
log4cxx: SSL Connect Failed to <IP>
```

Action: To work around this issue,

1. Check whether remote server is reachable and is listening on the given port.
2. Check whether the certificate is valid.
3. Reload the xdasauditds module.

A XDASv2 Events

The XDASv2 events are classified into the following categories:

- Section A.1, “Account Management Events,” on page 39
- Section A.2, “Session Management Events,” on page 40
- Section A.3, “Data Item and Resource Element Management Events,” on page 41
- Section A.4, “Service or Application Management Events,” on page 42
- Section A.5, “Service or Application Utilization Events,” on page 43
- Section A.6, “Peer Association Management Events,” on page 43
- Section A.7, “Data Item or Resource Element Content Access Events,” on page 44
- Section A.8, “Work Flow Management Events,” on page 45
- Section A.9, “Role Management Events,” on page 45
- Section A.10, “Exceptional Events,” on page 46
- Section A.11, “Audit Service Management Events,” on page 47
- Section A.12, “Authentication Event,” on page 48

A.1 Account Management Events

An identity is a token used to represent a particular user or entity. Blame or credit for an action goes to the identity for a set of activities within a system. Accounts exist in the application domains to associate attributes with the set of identifiers typically associated with identities. Identities can be a a human being or an automated identity, such as another service, which is acting on behalf of a human or a regularly scheduled system activity. In both the cases, account management is considered as persistent account creation, wherein an identity with some limited or unlimited set of system rights is associated with attributes.

NOTE: The `Modify Account Security Token` event could have been defined in terms of `Modify Account`, but modification of account security tokens is considered critical to audit security, and is thus given its own event.

Table A-1 *Account Management Event Taxonomy*

Event Name	Event Identifier	Corresponding eDir Event	Description	Use
Create Account	0.0.0.0	DSE_CREATE_ ENTRY	Create a new account	Consider this event as appropriate for any situation wherein an account, as defined above, is to be created.

Event Name	Event Identifier	Corresponding eDir Event	Description	Use
Delete Account	0.0.0.1	DSE_DELETE_ ENTRY	Delete an existing account	This event has the opposite semantic meaning of account creation. Use this event wherever such an account, as described above, is to be deleted.
Disable Account	0.0.0.2	DSE_LOGIN	Disable an existing account	Consider this event relevant for any situation where a particular record in an identifier database is disabled by an administrator or an automated security process such that it can no longer be used until it is re-enabled
Enable Account	0.0.0.3		Enable an existing account	This is the counterpart event to the disable account event defined above.
Query Account	0.0.0.4	DSE_SEARCH	Query an existing account	Consider the Query account events whenever a request for the attribute information of a particular account is made.
Modify Account	0.0.0.5	DSE_MERGE_ ENTRIES	Modify an existing account	Consider the Modify account events whenever a request to change attribute information of a particular account is made.
Modify Account Security Token	0.0.0.6	DSE_CHGPASS	Modify an existing account security token	An account security token may be a password, or any other type of authentication materials associated with a user account. Here, a user account means any type of account by which a user, application, or system service may authenticate, and then act with the rights of that account.
Query Account Security Token				
Delete Account Security Token				

A.2 Session Management Events

A session is the association of an initiator with a stream of communication. A session may represent a user's connection to server, as in the case of logging into a UNIX or Windows host, or a set of related transactions in a connection-less environment, as in the case of using a cookie to maintain persistent transactions between a browser client and a Web server.

Table A-2 *Session Management Event Taxonomy*

Event Name	Event Identifier	Corresponding eDir Event	Description	Use
Create Session	0.0.1.0		Create a new session	This event should be reported whenever a new session (as defined above) is created.
Terminate Session	0.0.1.1		Terminate an existing session	This event should be reported whenever an existing session (as defined above) is terminated.
Query Session	0.0.1.2		Query user session attributes	This event should be reported whenever attribute information is requested on an existing session.
Modify Session	0.0.1.3	DSE_CHANGE_CONN_STATE	Modify user session attributes	This event should be reported whenever attribute information is modified on an existing session.

A.3 Data Item and Resource Element Management Events

This set of events relate to the creation and management of data items and resource elements within a domain. The type of data item or resource element is dependent upon the domain. For example, files and directories, device special files, and shared memory segments within an operating system, tables and records within a database, messages within an email system. The term data item is used in this context to refer to any type of resource element.

Table A-3 *Data Item and Resource Element Management Event Taxonomy*

Event Name	Event Identifier	Corresponding eDir Event	Description	Use
Create Data Item	0.0.2.0	DSE_CREATE_ENTRY	Create a data item	This event is reported whenever a security-relevant data item or resource element is created.
Delete Data Item	0.0.2.1	DSE_DELETE_ENTRY	Delete a data item	This event is reported whenever a security-relevant data item or resource element is deleted
Query Data Item Attribute	0.0.2.2	DSE_COMPARE_ATTR_VALUE	Query data item attributes	This event is reported whenever a security-relevant data item or resource element is queried – either for value, or for an attribute of the data item.

Event Name	Event Identifier	Corresponding eDir Event	Description	Use
Modify Data Item Attribute	0.0.2.3	DSE_DEFINE_ATTR_DEF DSE_REMOVE_ATTR_DEF DSE_REMOVE_CLASS_DEF DSE_DEFINE_CLASS_DEF DSE_MODIFY_CLASS_DEF	Modify data item attributes	This event is reported whenever a security-relevant data item or resource element is modified – either the value, or an attribute of the data item

A.4 Service or Application Management Events

This set of events relates to the management of services or applications. For example, the RPM package manager might throw these events as packages are installed or removed from a Linux system. Windows 32 Service Control Manager (SCM) events sent to the Windows 32 System Event Log may be translated into these events as they are imported into OpenXDASv2. This set of events could also be much more domain-specific, including concepts such as installing, removing, or configuring installable executable-modules within a single application domain. The key idea is to ensure that reported events have security significance.

Table A-4 *Service or Application Management Event Taxonomy*

Event Name	Event Identifier	Corresponding eDir Event	Description	Use
Install Service	0.0.3.0	DSE_CHANGE_MODULE_STATE	Install a service or application	This event is reported when a service or application is installed
Remove Service	0.0.3.1	DSE_CHANGE_MODULE_STATE	Remove a service or application	This event is reported when a service or application is removed.
Query Service Configuration	0.0.3.2		Query the configuration of a service or application	This event is reported when service or application configuration information is requested.
Modify Service Configuration	0.0.3.3		Modify configuration of a service or application	This event is reported when service or application configuration information is modified.
Disable Service	0.0.3.4	DSE_CLOSE_BINDERY	Disable a service or application	This event is reported when a service, operation or function is disabled.
Enable Service	0.0.3.5	DSE_OPEN_BINDERY	Enable a service or application	This event ise reported when a service, operation or function is enabled.

A.5 Service or Application Utilization Events

This class of events relates to the use of services and applications. They typically map to the execution of a program or a procedure and manipulation of the processing environment.

Table A-5 *Service or Application Utilization Events Taxonomy*

Event Name	Event Identifier	Corresponding eDir Event	Description	Use
Invoke Service	0.0.4.0	DSE_START_UPDATE_SCHEMA	Invoke a service or application	This event is reported when a security-relevant service is invoked.
Terminate Service	0.0.4.1	DSE_END_UPDATE_SCHEMA	Terminate a service or application	This event is reported when a service is terminated.
Query Process Context	0.0.4.2		Query a processing context	This event is reported when any attributes of a process context are queried – this event is somewhat specific to operating systems, but some use can be found in other domain-specific applications.
Modify Process Context	0.0.4.3	DSE_SERVER_RENAME DSE_SYNTHETIC_TIME DSE_SERVER_ADDRESS_CHANGE	Modify processing context	This event is reported when any attributes of a process context are modified – this event is somewhat specific to operating systems, but some use can be found in other domain-specific applications.

A.6 Peer Association Management Events

Peer association events are related to the association of a user or identity with a group, or the association of two users in some domain-specific context. For example, adding an LDAP user to a group, or associating two users for a domain-specific purpose in an application's identity association database.These events are also related to the association of identities within disparate authentication domains for purposes of federation.

For example, when an identity in domain A makes a request to a service governed by domain B, then a peer association is required between these domains – often this is called a trust relationship. From an implementation perspective, setting up a trust relationship is often done by establishing an identity in domain B, which is used as a proxy for any request coming from any identity in domain A. Trust relationships can be much more complex, however, as individual identities in domain A can have individual associations with specific domain B identities.

Table A-6 *Peer Association Management Events Taxonomy*

Event Name	Event Identifier	Corresponding eDir Event	Description	Use
Create Peer Association	0.0.5.0		Create an association with a peer	This event is reported when a new peer association is created.
Terminate Peer Association	0.0.5.1		Terminate an association with a peer	This event is reported when an existing peer association is destroyed.
Query Association Context	0.0.5.2		Query an association context	This event is reported when the attributes of a peer association are queried.
Modify Association Context	0.0.5.3		Modify an association context	This event is reported when the attributes of a peer association are modified.
Receive Data Via Association	0.0.5.4		Receive data via an association	This event is reported when data is received from a service in an authentication domain specifically via a trust relationship or peer association.
Send Data Via Association	0.0.5.5		Send data via an association	This event is reported when data is sent to a service in an authentication domain specifically via a trust relationship or peer association.

A.7 Data Item or Resource Element Content Access Events

Resource content-access events are related to access of any data files protected by an authentication domain. This could be file system files, database records, Web pages etc. While instrumenting applications, consider securing access to the resources. Resource access can be a high-bandwidth process. Therefore, only security-relevant events should be reported. Such instrumentation should be configurable at the application level by the application administrator, thus must be policy driven. This implies that such applications add additional infrastructure and user interface to allow administrators to manage the resource-access events that has to be audited, and determine the unimportant events within the security context.

Table A-7 *Data Item or Resource Element Content Access Events Taxonomy*

Event Name	Event Identifier	Corresponding eDir Event	Description	Use
Create Data Item Association	0.0.6.0		Create association with a data item	This event is reported when rights are granted by an identity to a specific data item – when a trust relationship is established between an identity and a data item.

Event Name	Event Identifier	Corresponding eDir Event	Description	Use
Terminate Data Item Association	0.0.6.1		Terminate association with a data item	This event is reported when rights are revoked from an identity to a specific data item – when a trust relationship is revoked between an identity and a data item.
Query Data Item Association	0.0.6.2		Query context of association with a data item	This event is reported when rights are queried for an identity on a specific data item – when trust relationship attributes are queried for a specific identity and data item.
Modify Data Item Association	0.0.6.3		Modify context of association with data item	This event is reported when rights are modified on the previously established relationship between an identity and specific data item.
Query Data Item Contents	0.0.6.4		Query data item contents	This event is reported when a data item is read on behalf of an identity.
Modify Data Item Contents	0.0.6.5		Modify data item contents	This event is reported when a data item is written on behalf of an identity.

A.8 Work Flow Management Events

Even though work flow events can be classified in terms of data items, work flow has its own category of events within the XDASv2 taxonomy. Work flow events is used whenever a work flow process is instrumented for audit events.

Table A-8 *Workflow Management Event Taxonomy*

Event Name	Event Identifier	Corresponding eDir Event	Description	Use
Request Work Flow Approval	0.0.7.0		Initiate a workflow approval request	Approval for a work flow item has been requested.
Receive Work Flow Approval	0.0.7.1		Receive a work flow approval notice	Approval for a work flow item has been received by appropriate authority.
Escalate Work Flow Request	0.0.7.2		A work flow item was escalated	A work flow request has been escalated.
Send Work Flow Notification	0.0.7.3		A work flow notification was sent	Sent a work flow change notification.

A.9 Role Management Events

Role management event may also be classified in terms of data items, but role management is key to systems that manage identity, so these were also given their own category within the XDASv2 taxonomy.

Table A-9 *Role Management Event Taxonomy*

Event Name	Event Identifier	Corresponding eDir Event	Description	Use
Create Role	0.0.8.0		Create a new role	Creates a new role, or an attempt is made to create a new role.
Delete Role	0.0.8.1		Delete an existing role	An existing role is deleted, or an attempt is made to delete an existing role.
Disable Role	0.0.8.2		Disable an existing role	An existing role is disabled, or an attempt is made to disable an existing role.
Enable Role	0.0.8.3		Enable an existing role	A previously disabled role is re-enabled, or an attempt is made to enable a previously disabled role.
Query Role	0.0.8.4		Query role attributes	Role attributes are queried, or an attempt is made to query role attributes.
Modify Role	0.0.8.5		Modify a role attribute	Role attributes are modified, or an attempt is made to modify role attributes.

A.10 Exceptional Events

Exceptional events are generated very rarely, and are considered important because they are generated. For instance, shutting down an enterprise-critical server is exceptional because it can't happen without someone's permission.

Table A-10 *Exceptional Event Taxonomy*

Event Name	Event Identifier	Corresponding eDir Event	Description	Use
Start System	0.0.9.0		Start a system	This event is reported when a server, system, or mission-critical application starts up.
Shutdown System	0.0.9.1		Shutdown a system	This event is reported when a server, system, or mission critical application shuts down.
Resource Exhaustion	0.0.9.2		Resource exhaustion	This event is reported when a server, system, or mission critical application runs out of some critical resource, like memory or disk space. It is often difficult to report such events because often the critical resource in question is required in order to report the event.

Event Name	Event Identifier	Corresponding eDir Event	Description	Use
Resource Corruption	0.0.9.3		Resource Corruption	This event is reported when a server, system, or mission critical application detects a resource corruption (memory, disk file, etc).
Resource Unavailable	0.0.9.4		Resource Unavailable	This event is reported when a server, system, or mission critical application becomes unavailable.
Resource Available	0.0.9.5		Resource Available	This event is reported when a server, system, or mission critical application becomes available. This event is usually reported if the resource has been unavailable for a period of time.
Back up Data Store	0.0.9.6		Back up Data Store	This event is reported when a server, system, or mission critical application backs up a critical data store.
Recover Data Store	0.0.9.7		Recover Data Store	This event is reported when a server, system, or mission critical application restores a critical data store.

A.11 Audit Service Management Events

Audit services have traditionally been classified by themselves because auditing represents a lower level activity than the security events which themselves are being audited. By classifying audit events separately, an entire category of endless-loop defects can be avoided. Developers instrumenting applications will probably never need to report these events, as they are generally reported by systems like OpenXDAS itself. However, they are documented here for the sake of completeness.

Table A-11 *Audit Service Management Event Taxonomy*

Event Name	Event Identifier	Correspondin g eDir Event	Description	Use
Configure Audit Service	0.0.10.0		Configure audit service	Configuration data has been changed for an audit subsystem. OpenXDAS reports this event when a SIGHUP is received, indicating that the xdasd configuration file has been modified and should be re-read.
Audit Data Store Full	0.0.10.1		Audit datastore is full	This event is reported by OpenXDAS when an audit log is full, and can no longer accept additional audit records. Where possible, space is reserved for this event, in case it must be reported.

Event Name	Event Identifier	Corresponding eDir Event	Description	Use
Audit Data Store Corrupted	0.0.10.2		Audit datastore is corrupted	This event is reported by OpenXDAS when the data store reports that an audit log has been corrupted. Generally, this condition is not detected unless a request is made to read an audit stream, and the audit log reports that it cannot be read due to corruption.

A.12 Authentication Event

XDASv1 specified authentication as a modification of session attributes. XDASv2 makes authentication a first class event because authentication is critical to an audit.

Table A-12 *Authentication Events Taxonomy*

Event Names	Event Identifier	eDirectory Events	Description	Use
Authenticate Session	0.0.11.0		A new identity is associated with a session	When a user authenticates a session, a new identity is associated with that session. This identity is then used to authorize requests for protected resources.
Unauthenticate Session	0.0.11.1		A user has actively disassociated his identity from an existing authenticate session.	When a user clicks the “Logout” button on his or her web browser, the previously authenticated identity is removed from an existing authenticated session.
Federate Identity	0.0.11.2		A remote identity is associated with a local identity.	An identity relationship is established between a user at XYZ.COM and the local identity provider.
Unfederate Identity	0.0.11.3		A remote identity is disassociated from a local identity.	An existing identity relationship between a user at an external identity provider and the local identity provider is removed.

Event Names	Event Identifier	eDirectory Events	Description	Use
Create Access Token	0.0.11.4		A SAMLv2, WS-*, OAuth, or other access token was provided upon request.	A resource access token was created by a service (or identity) provider to send to a service consumer. Access is limited by time frame, specifically requested resources, or other limiting criteria, in terms of a contract specified by previously agreed upon name/value pairs in the token. The act of creating and sending an access token is the start of a new pseudo-identity with limited and specific rights to protected resources. This pseudo-identity can be used as a correlation identifier between this and future authorization events. The actually identity of the system user behind the access token may or may not be hidden from the consumer.
Destroy Access Token	0.0.11.5		An existing SAMLv2, WS-*, OAuth, or other access token was destroyed or decommissioned.	A previously created access token was decommissioned such that it is no longer allowed to be used for access to protected resources. Future requests for access to protected resources, based on this access token should be denied.

B XDASv2 Schema

- Section B.1, “XDAS V2 JSON Schema,” on page 51
- Section B.2, “XDAS Field Definitions,” on page 54
- Section B.3, “Outcome Codes,” on page 56
- Section B.4, “Example of an Event,” on page 56

The XDAS schema is defined as follows:

B.1 XDAS V2 JSON Schema

```
{
    "id":"XDASv2",
    "title":"XDAS Version 2 JSON Schema",
    "description":"A JSON representation of an XDASv2 event record.",
    "type":"objectr",
    "properties":{
      "Source":{
        "description":"The original source of the event, if applicable.",
        "type":"string",
        "optional":true
      },
      "Observer":{
        "description":"The recorder (ie., the XDASv2 service) of the event.",
        "type":"object",
        "optional":false,
        "properties":{
          "Account":{"$ref":"account"},
          "Entity":{"$ref":"entity"}
        }
      },
      "Initiator":{
        "description":"The authenticated entity or access token that causes an
event.",
        "type":"object",
        "optional":false,
        "properties":{
          "Account":{"$ref":"account","optional":true},
          "Entity":{"$ref":"entity"},
          "Assertions":{
            "description":"Attribute/value assertions about an identity.",
            "type":"object",
            "optional":true
          }
        }
      },
      "Target":{
        "description":"The target object, account, data item, etc of the event.",
        "type":"object",
        "optional":true,
        "properties":{
          "Account":{"$ref":"account"},
          "Entity":{"$ref":"entity"},
          "Data":{
```

```
            "description":"A set attribute/value pairs describing the target
object.",        *
            "type":"object",
            "optional":true
          }
        }
      },
      "Action":{
        "description":"The action describes the event in a uniform manner.",
        "type":"object",
        "optional":false,
        "properties":{
          "Event":{
            "description":"The event identifier in standard XDASv2 taxonomy.",
            "type":"object",
            "optional":false,
            "properties":{
              "Id":{
                "description":"The XDASv2 taxonomy event identifier.",
                "type":"string",
                "optional":false,
                "pattern":"/^[0-9]+(\.[0-9]+)*$/"
              },
              "Name":{
                "description":"A short descriptive name for the specific event.",
eg. a new replica is added
                "type":"string",
                "optional":true
              },
      "CorrelationID":{
          "description":"Correlation ID, source#uniqueID#connID",
                 "type":"string",
                 "optional":true
      }
     },
     "SubEvent":{
      "type":object
      "description": "Describes the actual domain specific event that has
occured.",
      "optional":true,
      "properties":{
        "Name"":{
                    "description":"A short descriptive name for this event.",
                    "type":"string",
                    "optional":true
                  },
      }
            }
          }
          "Log":{
            "description":"Client-specified logging attributes.",
            "optional":true,
            "properties":{
              "Severity":{"type":"integer", "optional":true},
              "Priority":{"type":"integer", "optional":true},
              "Facility":{"type":"integer", "optional":true}
            }
          }
          "Outcome":{
            "description":"The XDASv2 taxonomy outcome identifier.",
            "type":"string",
            "optional":false,
            "pattern":"/^[0-9]+(\.[0-9]+)*$/"
          }
          "Time":{
            "description":"The time the event occurred.",
            "type":"object",
            "optional":false,
            "properties":{
              "Offset":{
```

```
                "description":"Seconds since Jan 1, 1970.",
                "type":"integer"
              },
              "Sequence":{
                "description":"Milliseconds since last integral second.",
                "type":"integer",
                "optional":true
              },
              "Tolerance":{
                "description":"A tolerance value in milliseconds.",
                "type":"integer",
                "optional":true
              },
              "Certainty":{
                "description":"Percentage certainty of tolerance.",
                "type":"integer",
                "optional":true,
                "minimum":0,
                "maximum":100,
                "default":100,
              },
              "Source":{
                "description":"The time source (eg., ntp://time.nist.gov).",
                "type":"string",
                "optional":true
              },
              "Zone":{
                "description":"A valid timezone symbol (eg., MST/MDT).",
                "type":"string",
                "optional":true
              }
            }
      "ExtendedOutcome":{
            "description":"The XDASv2 taxonomy outcome identifier.",
            "type":"string",
            "optional":false,
            "pattern":"/^[0-9]+(\.[0-9]+)*$/"
          }
        }
      }
    }
  },
  {
    "id":"account",
    "description":"A representation of an XDAS account.",
    "type":"object",
    "properties":{
      "Domain":{
       "description":"A (URL) reference to the authority managing this account.",
/* lets take it as the partition?
        "type":"string"
      },
      "Name":{
        "description":"A human-readable account name.",        - DN
        "type":"string",
        "optional":true
      },
      "Id":{
        "description":"A machine-readable unique account identifier value.",  -
EntryID
        "type":"integer"
      }
```

```
    }
  },
  {
    "id":"entity",                    - Server details for Target, client address
details for the initiator
    "description":"A representation of an addressable entity.",
    "type":"object",
    "properties":{
      "SysAddr":{"type":"string","optional":true},
      "SysName":{"type":"string","optional":true},
      "SvcName":{"type":"string","optional":true},
      "SvcComp":{"type":"string","optional":true},
    }
  }
```

B.2 XDAS Field Definitions

These fields in the schema are the XDASv2 fields defined specifically for audit events. Some or all of these fields may also be relevant to other types of event, but information of this sort is required for auditing services. The XDASv2 JSON record format is open. By that, we mean that any additional fields may be added to the record at any place, as long as they don't conflict with the field values defined for audit by the XDASv2 standard.Thus, if there is a particular type of correlation data, such as a workflow identifier, or a session identifier that can be used as correlation data points between events within a particular workflow or client session, you may add these fields. Simply choose a non-conflicting name for your field.

Table B-1 *XDAS Field Definitions*

XDAS Field	Description
Source (Optional)	The source of an event identifies the event service of another system from which this event was originally defined and converted to an XDAS event. Since many events are generated directly by XDAS clients, the source field is optional.
Initiator	The initiator of an event is the authenticated entity that initially provoked creation of the event. Note that an initiator need not be identified. If the entity can't be identified - perhaps an entity is attempting to login, thus provoking the generation of a login event by an observer - then as much information about the origin of the event as possible should be specified. NOTE: In the special case of a login event, the authenticated identity of the initiator is not yet known until after the login attempt has succeeded. Therefore a failed login event should not give the identity of the target account as the identity of the initiator. An initiator is described in terms of an account and an entity (described below), as well as an optional set of assertions. These assertions describe, in terms of a set of name/value pairs, the attributes of the initiator identity. Some initiators are not known by a specific account, but are known only by a set of assertions (SAML2, for instance) that describe the rights of the actor. The schema is not defined for these assertions, as they will be different for each class and potentially for each individual object.
Action	The action identifies the event that is being recorded. This field provides the XDASv2 event identifier, as well as an outcome code (success, or failure class), and the time the event occurred, with as much accuracy as possible.
Event	The event field is the key to XDAS events. Event encapsulates a taxonomical identifier and a short descriptive name for human readability.
Id	The event Id code represents the event identifier, defined by the XDASv2 standard event taxonomy, and extensions defined by the Novell CSS product.

XDAS Field	Description
Name	The event name is a human readable representation of the event identifier. The event name is optional, but recommended for readability.
Data	The event data provides additional descriptive information about the event.
Log	The log field contains standard syslog-like log-level values, in terms of Severity and Facility numeric identifiers. The log field is optional, as well as every sub-field within the log field. These values should only be used when necessary, as they generally represent judgment calls on the part of the instrumentor. Such judgment calls are best left to analysis software or engineers once the event data is collected.
Outcome	For details on outcome codes, see Section B.3, “Outcome Codes,” on page 56.
Time	The event time is the time recorded by the observer at the point the event was committed to the event service. Time values are gathered by the XDAS client helper library. Thus, there is no reason to be concerned about values stored in this field, as the helper library will attempt to be as accurate as possible when generating time information.
Offset	The offset field contains a value representing the number of seconds since midnight, January 1, 1970 - otherwise known as the UNIX epoch.
Sequence	The sequence field contains a unique numeric value identifying this event from another event which may have been recorded within the same second. For the most part, this value should be taken as a monotonically increasing numeric value that begins at zero and continues until the next second boundary, at which point, it begins again at zero.
Tolerance	The tolerance value is a value between 0 and 100, indicating the tolerance of the clock used to record the time in offset. Values of zero indicate the clock is very accurate. Values of 100 indicate that the clock should not be trusted.
Certainty	The certainty value is a value between 0 and 100, indicating the percentage certainty of the tolerance value. Zero means there is no certainty of the tolerance, and thus, it shouldn't be trusted to any degree of accuracy. A value of 100 indicates that the tolerance value is very accurate.
Source	The time source is information indicating the source of time for the observer system. This may be a URL for a time server, or simply a local time source, such as a hardware clock.
Zone	The time zone is the new time zone string representing the time zone of this clock.
Target (Optional)	The target of an event is the account or protected resource upon which the initiator is attempting to act, thereby provoking the generation of an event. A target is described in terms of an account and an entity (described below), as well as an optional and unspecified Data object. The Data object is a set of name/value pairs describing class-specific attributes of the actor. The schema does not define the actual fields, as different classes will have a unique set of data attributes (if any).
Observer	The observer of an event is the authenticated identity of an entity (service) that is monitoring the system, and generating events based on initiator actions. An observer is described in terms of an account and an entity (described below).
Referenced Classes	The observer, initiator, and target fields contain references to the account and entity classes defined separately within the schema. These other classes identify key attributes of the three primary actors within an audit event.

XDAS Field	Description
Account Class	The account class represents the identity of the actor. This identity is relative to an authentication realm or Domain. Both an account name and an account Id are provides, although only the Id is really required. The Name is for human readability.
Account Domain	The account Domain defines the authentication authority of the actor. Account identifiers mean very little without an authentication authority.
Account Name	The account Name is optional, providing human readability.
Account Id	The account Id is a unique identifier of the account within the authentication Domain.
Entity Class	The entity class describes the location of the actor. This location is defined in terms of a system access end point (IP network) address and a system access end point (host/domain) name. Additional fields are also available to describe the service and component names within the software that manages the above end points.
Entity SysAddr	An IP address describing the access end point of the software actor.
Entity SysName	A host/domain name describing the access end point of the software actor.
Entity SvcName	A service name further describing the service that manages the above end point.
Entity SvcComp	A service component name describing the component within the above service.

B.3 Outcome Codes

The outcome code is a hierarchical numeric value much like the event code. Outcome codes indicate success or a failure class and reason. The success hierarchy is encapsulated by the 0.x sub-arc. Failure classes are represented by the 1.x hierarchy. Denial codes are represented by the 2.x hierarchy.

B.4 Example of an Event

An example event is given below:

```
{"Source" : "eDirectory#DS",
             "Observer" : {"Account" : {"Domain" : "TREE_NAME","Name" :
"CN=server1,O=novell"},
"Entity" : {"SysAddr" : "164.99.90.129","SysName" : "blr-edir-phoenix"}},

"Initiator" : {"Account" : {"Domain" : "TREE_NAME","Name" :
"CN=server1,O=novell"},"Assertions" : {"NetAddress" : "164.99.90.129"}},"Target" :
{"Data" : {"Name" : "CN=server1,O=novell","newFlags" : "262178","oldFlags" :
"35"}},"Action" : {"Event" : {"Id" : "0.0.1.3","Name" :
"MODIFY_SESSION","CorrelationID" : "eDirectory#-1#","SubEvent" :
"DSE_CHANGE_CONN_STATE"},"Time" : {"Offset" : 1286430957},"Log" : {"Severity" :
7},"Outcome" : "0","ExtendedOutcome" : "0"}}
```