Novell Nsure™ Audit

1.0.3

www.novell.com

ADMINISTRATION GUIDE

July 18, 2006

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.

You may not export or re-export this product in violation of any applicable laws or regulations including, without limitation, U.S. export regulations or the laws of the country in which you reside.

U.S. Patent No. 5,157,663; 5,349,642; 5,455,932; 5,553,139; 5,553,143; 5,572,528; 5,594,863; 5,608,903; 5,633,931; 5,652,854; 5,671,414; 5,677,851; 5,692,129; 5,701,459; 5,717,912; 5,758,069; 5,758,344; 5,781,724; 5,781,733; 5,784,560; 5,787,439; 5,818,936; 5,828,882; 5,832,274; 5,832,275; 5,832,483; 5,832,487; 5,859,978; 5,870,561; 5,870,739; 5,873,079; 5,878,415; 5,884,304; 5,893,118; 5,903,650; 5,903,720; 5,905,860; 5,910,803; 5,913,025; 5,913,209; 5,915,253; 5,925,108; 5,933,503; 5,933,826; 5,946,002; 5,946,467; 5,956,718; 5,956,745; 5,964,872; 5,974,474; 5,983,223; 5,983,234; 5,987,471; 5,991,810; 6,002,398; 6,014,667; 6,016,499; 6,023,586; 6,029,247; 6,052,724; 6,061,726; 6,061,740; 6,061,743; 6,065,017; 6,081,774; 6,081,814; 6,094,672; 6,098,090; 6,105,062; 6,105,069; 6,105,132; 6,115,039; 6,119,122; 6,144,959; 6,151,688; 6,157,925; 6,167,393; 6,173,289; 6,216,123; 6,219,652; 6,233,859; 6,247,149; 6,269,391; 6,286,010; 6,308,181; 6,314,520; 6,324,670; 6,338,112; 6,345,266; 6,353,898; 6,424,976; 6,466,944; 6,477,583; 6,477,648; 6,484,186; 6,496,865; 6,510,450; 6,516,325; 6,519,610; 6,532,451; 6,532,491; 6,539,381; RE37,178. Patents Pending.

Novell, Inc.
404 Wyman Street, Suite 500
Waltham, MA 02451
U.S.A.
www.novell.com

Online Documentation: To access the online documentation for this and other Novell products, and to get updates, see www.novell.com/documentation.

Novell Trademarks

BorderManager is a registered trademark of Novell, Inc.

ConsoleOne is a registered trademark of Novell, Inc. in the United States and other countries.

DirXML is a registered trademark of Novell, Inc. in the United States and other countries.

eDirectory is a registered trademark of Novell, Inc.

iChain is a trademark of Novell, Inc.

NCP is a trademark of Novell, Inc.

NDS is a registered trademark of Novell, Inc. in the United States and other countries.

NetMail is a trademark of Novell, Inc.

NetWare is a registered trademark of Novell, Inc. in the United States and other countries.

NetWare Core Protocol is a trademark of Novell, Inc.

NLM is a trademark of Novell, Inc.

Novell is a registered trademark of Novell, Inc., in the United States and other countries.

Novell Certificate Server is a trademark of Novell, Inc.

Novell Storage Services is a trademark of Novell, Inc.

Nsure is a trademark of Novell, Inc.

Novell is a registered trademark of Novell, Inc.

SUSE is a registered trademark of SUSE LINUX AG, a Novell company.

Third-Party Materials

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

Contents

8 Configuring Filters and Event Notifications 101

9 Generating Queries and Reports 107

10 Security and Non-Repudiation 159

11 Troubleshooting NSure Audit 167

A Event Structure 175

B Application Events 191

C Using MySQL with Nsure Audit 211

D Using Oracle with Nsure Audit 217

About This Guide

Welcome to Novell® Nsure™ Audit. This guide provides the information required to configure and manage Nsure Audit.

Audience

This guide is intended for network administrators.

Feedback

We want to hear your comments and suggestions about this manual. Please use the Feedback option at the bottom of each page of the Nsure Audit online documentation.

Documentation Updates

For the most recent version of the Novell Nsure Audit 1.0.3 Administration Guide, see the Novell Documentation Web site, (http://www.novell.com/documentation/lg/nsureaudit/index.html).

Additional Documentation

For information on installing Novell Nsure Audit, see the *Novell Nsure Audit Installation Guide*.

Documentation Conventions

In this documentation, a greater-than symbol (>) is used to separate actions within a step and items within a cross-reference path.

A trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark.When a single pathname can be written with a backslash for some platforms or a forward slash for other platforms, the pathname is presented with a backslash. Users of platforms that require a forward slash, such as UNIX*, should use forward slashes as required by your software.

Overview

1

This section provides an overview of the Novell® Nsure™ Audit Report auditing system and reviews auditing fundamentals.

- Section 1.2, "Product Overview," on page 14
- Section 1.3, "Auditing Background and Fundamentals," on page 15

1.1 New Features

This section contains a listing of the new features available in the latest version of Nsure Audit.

1.1.1 eDirectory Instrumentation Enhancements

Several enhancements were made to the eDirectory™ instrumentation. These include the ability to:

- Choose between inline and journal logging of events. Inline mode logs events during the actual eDirectory process as they occur. Journal mode logs events in a separate thread, so the actual eDirectory process is not interrupted. Journal mode does not incur as much performance overhead, however, if the eDirectory server goes down, events in the journal that have not been processed are lost.
- Use an advanced grouping mechanism to group events related to an operation, providing easier searching and browsing of events. Events are now grouped by eDirectory transaction, which enables you to use the drill down feature of the Nsure Audit Report Application (LReport) to view all events associated with a transaction.
- Store the previous value of an eDirectory change. For example, when an attribute is deleted in eDirectory, Nsure Audit logs a delete attribute event in which the previous attribute value is stored in the data field of the event.
- Show a previous state of eDirectory before a change, based on logged information. For example, when a user is deleted from eDirectory, Nsure Audit logs a series of delete attribute events and a delete object event. Each of these events is grouped using the advanced grouping mechanism, making it easy to drill-down and view all events relating to this transaction. If it was later determined that this user was erroneously deleted, all removed attributes and their values could be retrieved from Nsure Audit, and the object could be reconstructed.
- Present attribute data in human-readable form. In previous versions, this information was in a binary format that was more difficult to access.

1.1.2 New Event Fields

Several additional fields were added to the Nsure Audit event structure to enhance querying and reporting. The Nsure Audit event structure now contains fields to report the originator of an event, the target and subcomponent affected by the event, as well as additional text and value fields.

1.1.3 Installation Enhancements

The installation has been enhanced on all supported platforms to provide more flexible, integrated installs. You now have the option of installing components individually, configuring the Platform Agent during install, and installing the Nsure Audit iManager plug-in during the process.

1.1.4 Microsoft SQL Server Support

The Nsure Audit Secure Logging Server now has the ability to store events in the Microsoft* SQL Server database. The Nsure Audit iManager plug-in has been updated to support establishing this connection.

1.1.5 Additional Supported Platforms

The Secure Logging Server now supports Windows* 2003 Server, RedHat* AS and ES.

1.1.6 JDBC Log Channel

The JDBC* Log Channel Driver has been enhanced to support any JDBC-enabled database, enabling you to log events to a number of different data stores supporting JDBC.

1.1.7 iManager Query and Verification Builder

The Query and Verification Builder interface in iManager has received several enhancements, including the ability to use custom macros in SQL queries. These custom macros make is simple to convert from hex to decimal, use IP addresses in queries, and reference the Nsure Audit table name using a keyword.

You can now also perform lightweight event verifications using the iManager interface.

1.1.8 Event Cache Limit

The Platform Agent now has additional parameters, contained in logevent.cfg, enabling you to specify the maximum size of the Nsure Audit event cache, and specify the action Nsure Audit takes when this limit is reached (stop logging, drop cache, or generate a warning).

1.1.9 Optional Expanded Event Data Field

The Platform agent has an additional parameter, contained in logevent.cfg, enabling you to specify the maximum size of the data field for each event. This option provides additional flexibility for applications logging events to Nsure Audit, as they can use an expanded event data field to increase the amount of information that can be stored with each event.

This increased size is optional, so applications not requiring this functionality can leave this off for increased performance.

1.2 Product Overview

Novell Nsure Audit is a centralized, cross-platform auditing service. It collects event data from multiple applications across multiple platforms and writes the data to a single, non-repudiable data store. Nsure Audit is also capable of creating filtered data stores. Based on criteria you define, Nsure Audit captures specific types of events and writes those events to secondary data stores.

After you have collected the data, the next challenge is making sense of it. Using the query and report generating tools included with Nsure Audit Report, you can evaluate the information in your data stores to determine resource access, usage patterns, and overall compliance with organizational policies and regulations.

Although queries and reports are invaluable in reviewing system activity, sometimes you need to know what is happening on your system as it happens. Therefore, Nsure Audit provides real-time notifications and real-time monitoring so you can assess and act on events as they occur.

To some extent, Nsure Audit can even automate the process of responding to events in real time. The Critical Value Reset (CVR) channel allows you to flag Directory attributes with reset policies. If the value of a given attribute is changed, the CVR channel resets the value as per the policy defined in the CVR Channel object. For example, if your organization has a policy prohibiting security equivalence, you can create a CVR Channel object that automatically resets the Security Equals attribute to a null value if it is ever reset by an administrator.

We understand that security standards are becoming increasingly rigorous. In order to manage liability and protect assets, organizations need to be able to provide a record of all their electronic proceedings and to identify when business policies are being violated. With real-time monitoring, notifications, and historical reporting capabilities, Novell Nsure Audit can give you the facts you need to make informed decisions and ensure the safety of your most valuable corporate asset—its information.

1.3 Auditing Background and Fundamentals

Novell Nsure Audit provides the tools you need to audit your organization's compliance with internal and external policies and regulations; however, the use of secure logging technology such as Novell Nsure Audit does not, in itself, provide a complete auditing solution. Auditing is actually a human-driven process and Novell Nsure Audit is simply a tool to facilitate that process.

Therefore, a complete auditing strategy requires that you:

1. Define your organization's security and usage policies. That is, determine what resources your users are allowed to access, what rights they have to those resources, and so forth.
2. Log the events relevant to those policies.
3. Configure Notification Filters to notify you in real time when a policy violation occurs. You can also use Notification Filters to route the events to the Critical Value Reset (CVR) channel to trigger an automated response to the violation.
4. Perform regular compliance audits. This entails querying the data store for events relevant to your policies and then manually reviewing those events to determine if there are any violations of your corporate policies, when the violations occurred, and who was responsible.

After you have implemented your auditing strategy, Novell Nsure Audit provides the information you need to assess overall compliance with organizational policies and to respond to policy violations in a timely manner.

For example, in a secure environment, you might have a policy that prohibits assigning user rights using the Security Equals attribute because it makes it difficult to track and manage user rights. To audit this policy, you first configure Novell Nsure Audit to log the Change Security Equals event.

To facilitate a timely response to policy violations, you configure a Notification Filter to send a message to your mailbox any time the Change Security Equals event occurs. You also have the Notification Filter route the event to the CVR channel, which is configured to automatically reset the Security Equals attribute on User objects to a null value.

You can monitor your organization's compliance with this policy by using iManager or Nsure Audit Report to query the data store for Change Security Equals events. You then review the query results to determine when violations occurred and who was responsible.

System Architecture 2

As a system administrator, you need a clear understanding of Novell® Nsure™ Audit architecture and functionality so you can better design, configure, and maintain your auditing system. This section provides a basic explanation of the components that make up Novell Nsure Audit.

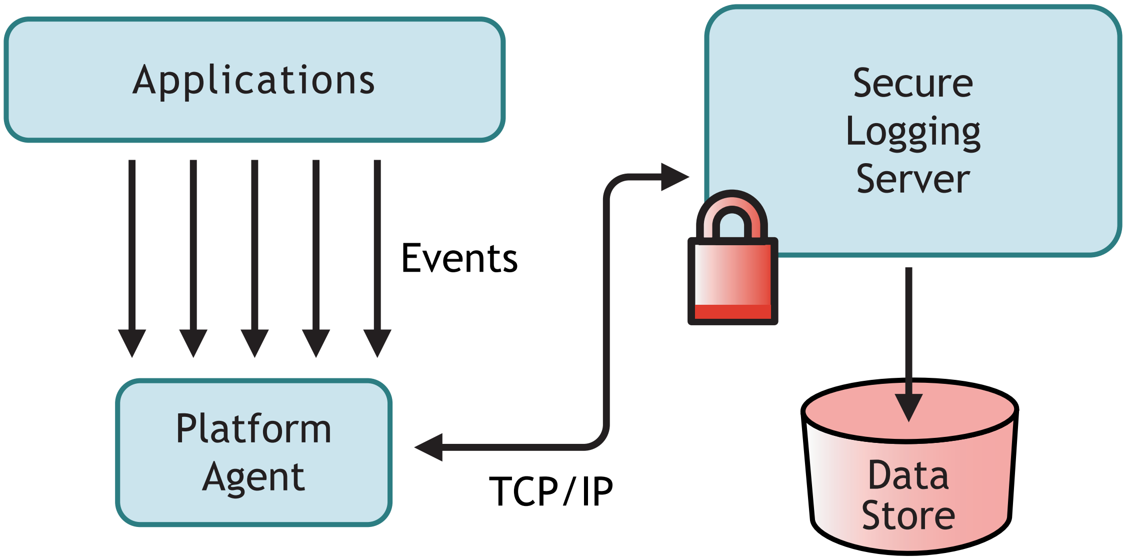
2.1 Nsure Audit System Components

Novell Nsure Audit has a highly modular architecture. Product functions are strategically divided among several different components to protect data integrity, optimize system performance, and provide maximum flexibility. Depending on usage and system resources, these components can be located on a single server or distributed across multiple servers.

The full auditing system consists of the following components:

- Platform Agent
- Secure Logging Server
- Data Store
- Reporting Applications

The Platform Agent, Secure Logging Server, and data store are the central components in this structure. To log events from system applications to the data store, Novell Nsure Audit uses a client/server model. The Platform Agent, as the client piece, receives all log data from the system applications. It securely transmits this data to the Secure Logging Server, which then writes the information to the data store.



Separating the Platform Agent from the actual logging function provides the following advantages:

- You can run Platform Agents on multiple servers throughout the network while still maintaining a single data store.
- Applications are not slowed down trying to commit an event to disk. Applications simply relay their log events to the Platform Agent, which then transmits the information the Logging Service. The Secure Logging Server assumes the full load of writing events to disk.
- You can off-load the system's logging overhead by running the Secure Logging Server on a dedicated server.

The remaining Nsure Audit component includes two reporting applications: iManager and Nsure Audit Report. These supplementary tools allow you, as the administrator, to tap vital data from the system.

The following sections provide a discussion of each component in the Nsure Audit architecture.

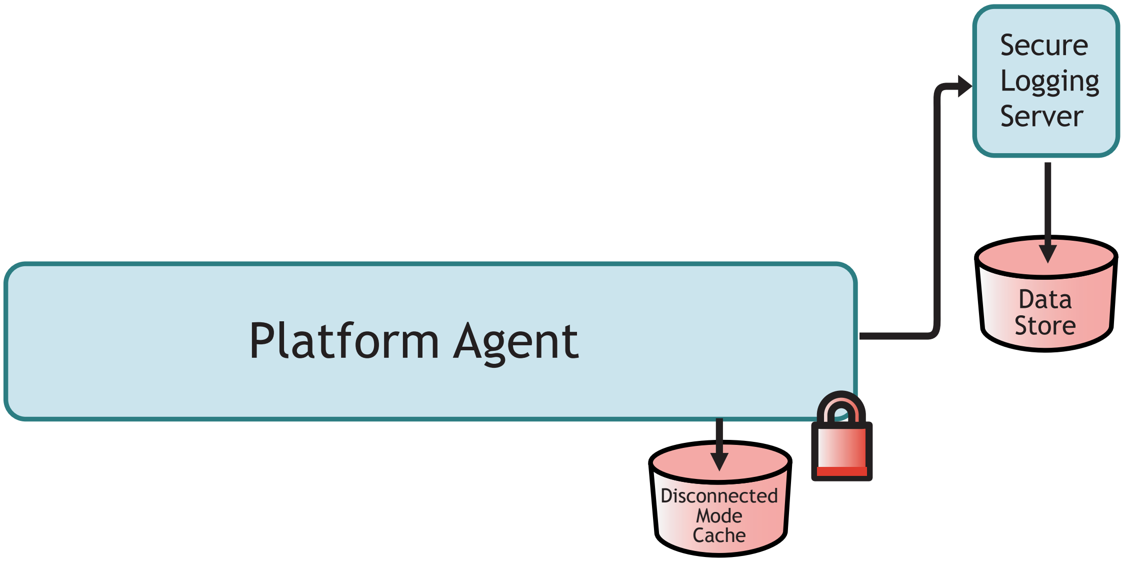
- "Platform Agent" on page 18
- "Secure Logging Server" on page 22
- "Data Store" on page 27
- "Reporting Applications" on page 28

2.1.1 Platform Agent

The Platform Agent (logevent) is the client portion of the Nsure auditing system. The Platform Agent receives logging information and system requests from authenticated applications and transmits the information to the Secure Logging Server.

For more information on program binaries, see Section H.1, "Program Files and Directories," on page 245. For more information on how applications authenticate with Nsure Audit, see Section 10.1, "Authenticating Logging Applications," on page 159.

If the connection between the Platform Agent and the Secure Logging Server fails, applications continue to log events to the local Platform Agent, just as they always do. The Platform Agent simply switches into Disconnected Cache Mode and the Cache Module writes all logged events to the local cache until the connection is restored. The switch into Disconnected Cache Mode is completely transparent to the logging applications. For more information, see "Disconnected Mode Cache" on page 41.

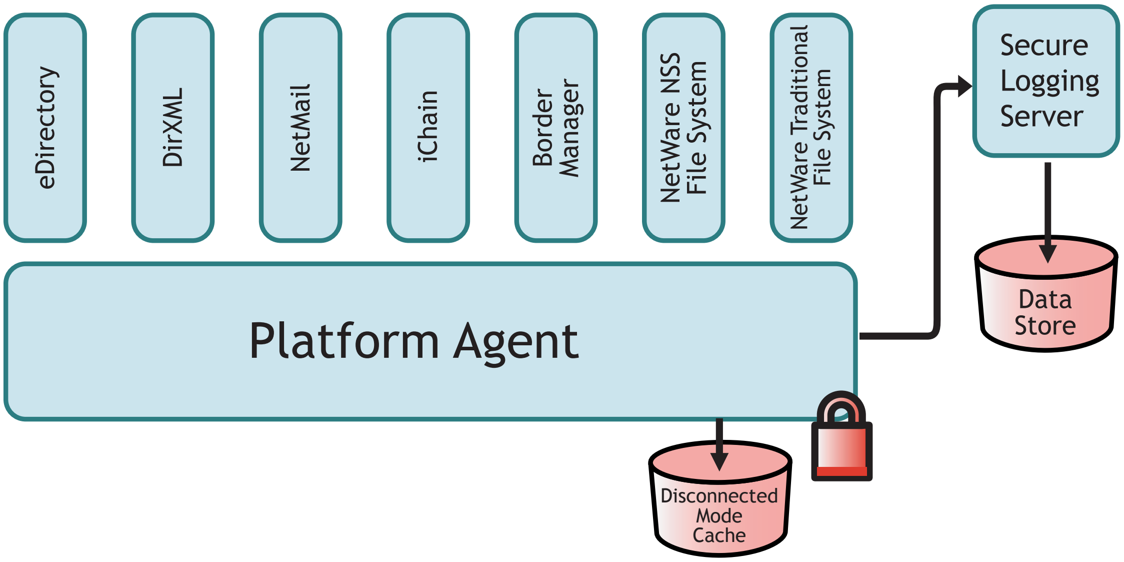


Supported Applications

Currently, the Platform Agent can receive log events from the following:

- Novell eDirectory™ 6.0 and higher
- DirXML® 2.0
- NetMail™ 3.5 and higher
- iChain® 2.2 SP1
- BorderManager® 3.8
- NetWare® NSS File System
- NetWare Traditional File System

NOTE: Before an application can log events to Novell Nsure Audit, it must be able to authenticate with the system and report events in the auditing system's required format. For more information on the authentication process, see Section 10.1, "Authenticating Logging Applications," on page 159. For more information on event structure, see Section A.1, "Event Structure," on page 175.

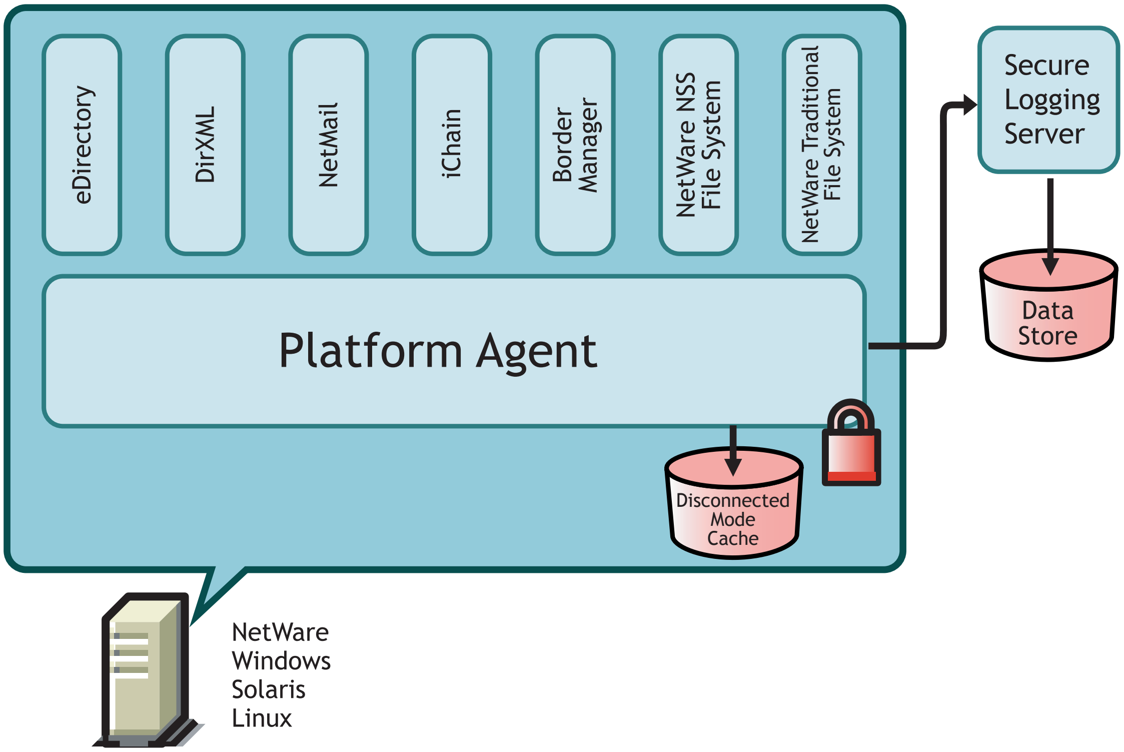


Supported Platforms

The Nsure Audit architecture requires that the Platform Agent be locally installed on every server or workstation running applications that log events to Nsure Audit.

This design ensures secure, uninterrupted logging because the logging applications are insulated from external communication failures. The Platform Agent is supported on the following platforms:

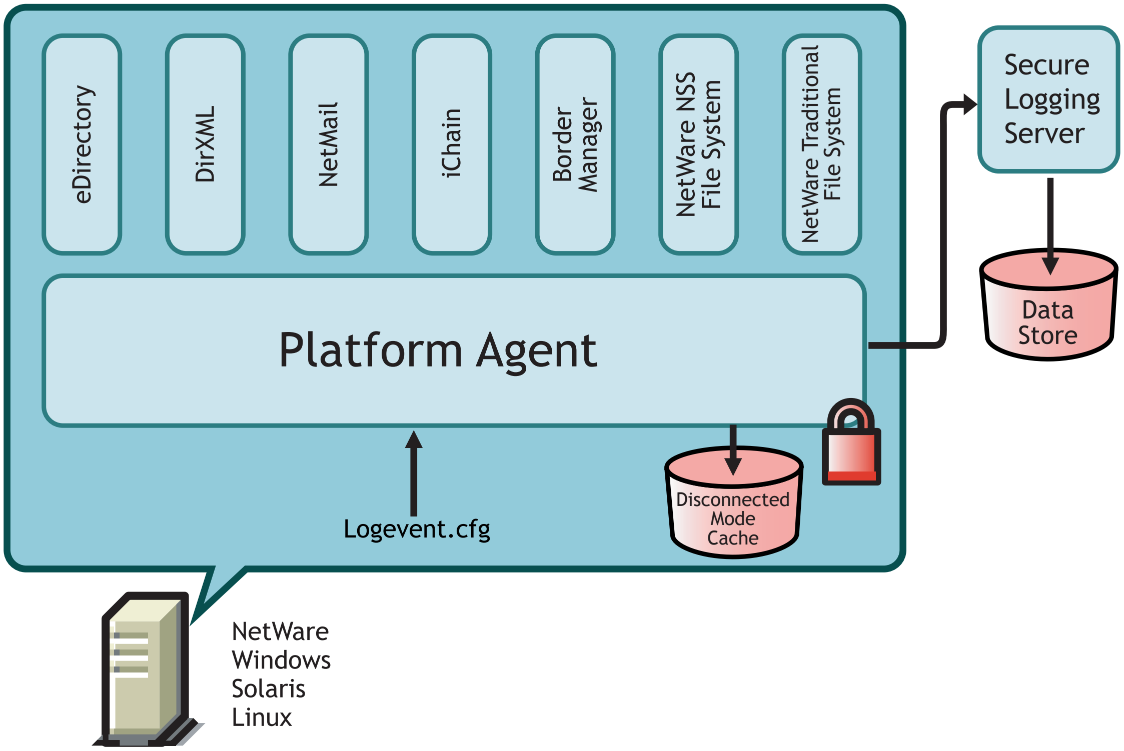
- NetWare 4.2 or later
- Windows* NT, Windows 2000 and Windows 2000 Server, Window XP, Windows 2003 Server.
- SUSE® Linux* Enterprise Server 8
- Solaris 8 and 9
- RedHat* Linux 7.3, 8, AS, and ES 2.1



Platform Agent Configuration

The Platform Agent is not configured through eDirectory. Instead, the Platform Agent's configuration settings are stored in a simple, text-based configuration file (logevent). This makes the Platform Agent small, unobtrusive, and self-contained—that is, it has no external dependencies so it is always available to receive logged events. Storing the Platform Agent's configuration in a text-based file also allows the Platform Agent to eventually run on platforms that do not have eDirectory support.

The logevent file stores the host name or IP address of the logging server, the Disconnected Mode Cache directory, port assignments, and other related information. For more information on Platform Agent configuration settings, including a sample logevent file, see "Logevent" on page 42.



2.1.2 Secure Logging Server

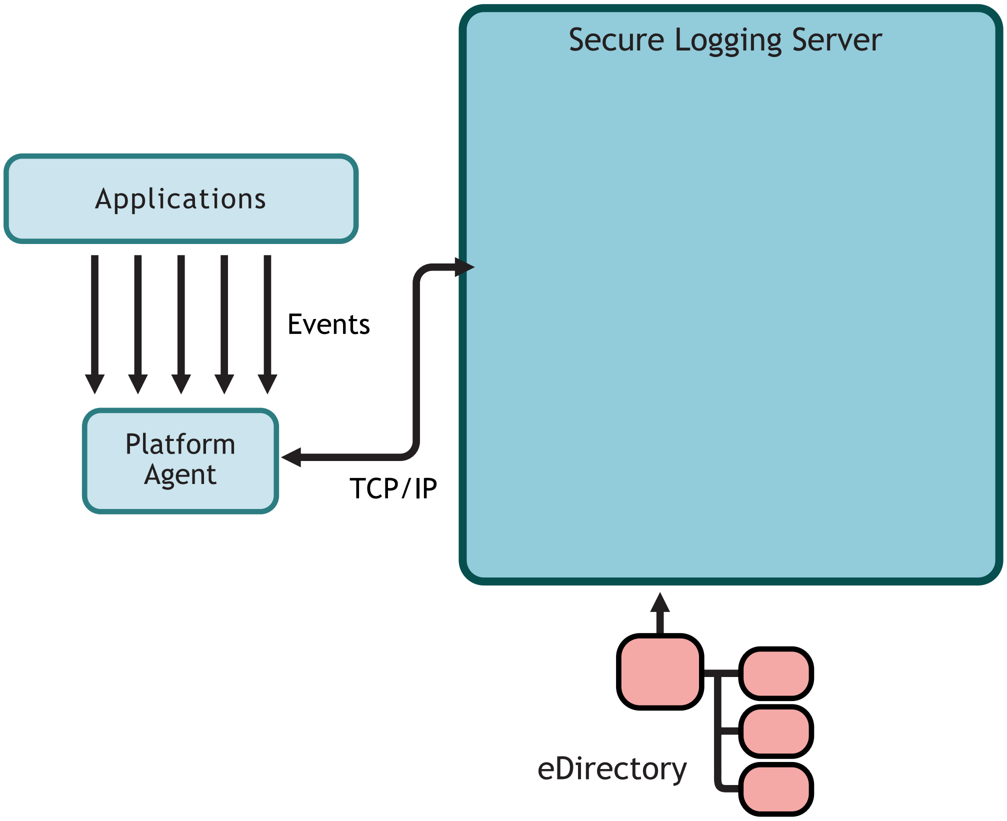
The Secure Logging Server (lengine) is the server component in the Nsure auditing system.The Secure Logging Server manages the flow of information to and from the Nsure auditing system—that is, it receives incoming events and requests from the Platform Agents, logs information to the data store, monitors designated events, and provides filtering and notification services. It can also be configured to automatically reset critical system attributes according to a specified policy. For more information on program binaries, see Section H.1, "Program Files and Directories," on page 245.

The Secure Logging Server supports the following platforms:

- NetWare 6.5
- NetWare 6.0 SP3 or later
- NetWare® 5.1 SP6 or later
- Windows 2003 Server
- Windows 2000 Server SP4 or later
- Solaris 8 and 9
- SUSE Linux Enterprise Server 8
- Red Hat Linux AS and ES 2.1

The Secure Logging Server is configured through eDirectory. The Logging Server object contains all the configuration settings for the Secure Logging Server. Consequently, the logging server must have access to eDirectory and the Logging Server object before it can launch the Secure Logging Server. For more information, see Section 4.2, “Configuring the Secure Logging Server,” on page 44.

NOTE: To minimize server reaction time and ensure high system performance, you should create a local replica of the Logging Server object and its associated objects on the logging server.



The Secure Logging Server provides the following services:

- Event Management
- Logging and Notification Channels
- Logging Service
- Notification Service

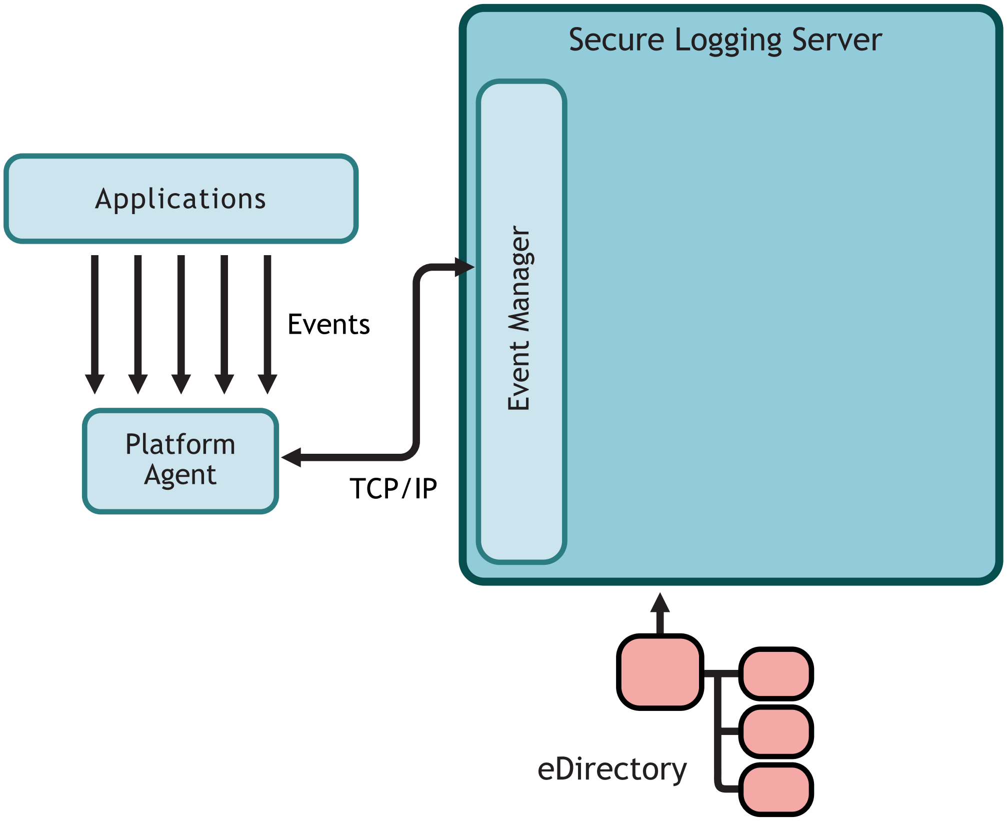
A description of each service follows.

Event Management

The Event Manager receives all incoming data from the Platform Agents and directs the information to the appropriate service. It also routes outgoing information from the logging server to the appropriate Platform Agent.

This mode of operation is very efficient. Indeed, the Event Manager service is designed to maximize system efficiency and performance. Depending on the Secure Logging Server's cache settings, the

Event Manager can handle more than 60,000 events per second. For information on configuring the Secure Logging Server's cache, see "Logging Server Objects" on page 45.



Logging and Notification Channels

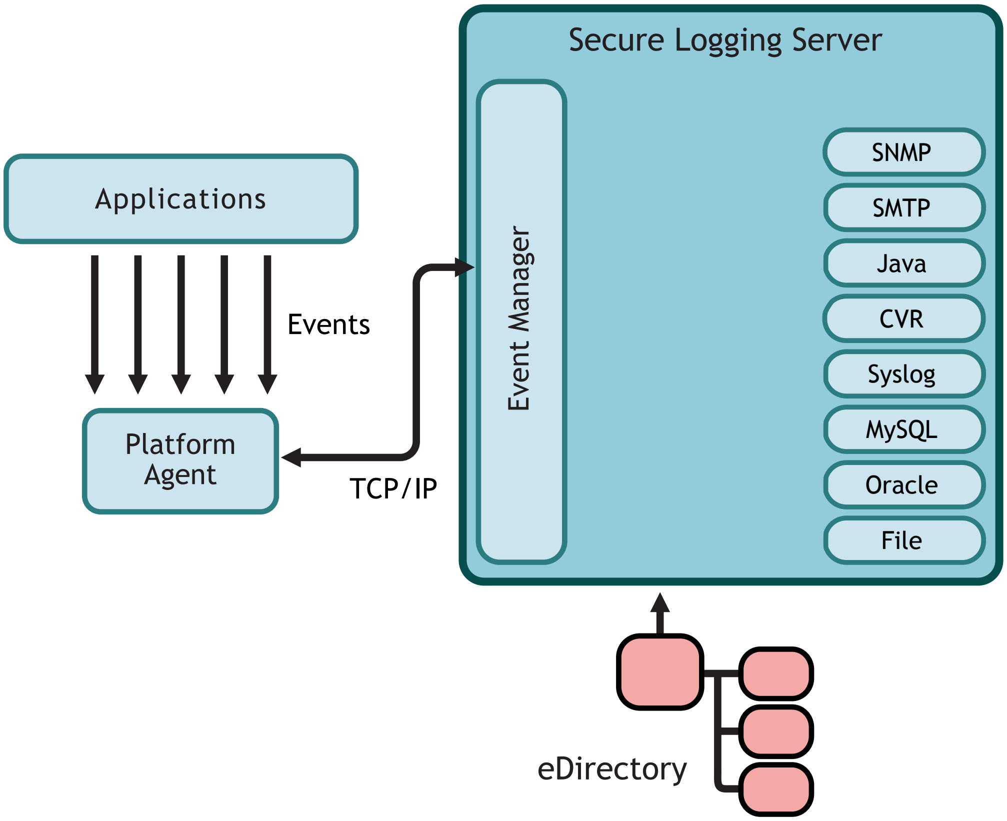
The Secure Logging Server uses channels to log events and provide event notification. For example, to e-mail events, the logging server uses the SMTP channel; to log events to an Oracle* database, the logging server uses the Oracle channel; and so forth.

Nsure Audit currently supports the following channels:

SMTP	Oracle (Available on NetWare using the Java JDBC channel driver.)
SNMP	File
Java	Syslog
MySQL	CVR (Critical Value Reset)
JDBC	Microsoft* SQL Server

Third-party channels can be easily incorporated into this structure. For more information, see the Novell Nsure Audit SDK (http://developer.novell.com/ndk/naudit.htm).

Channels are configured in eDirectory using Channel objects. Each Channel object stores the information the logging server needs to use its associated channel. For further information, see "Channel Objects" on page 31.

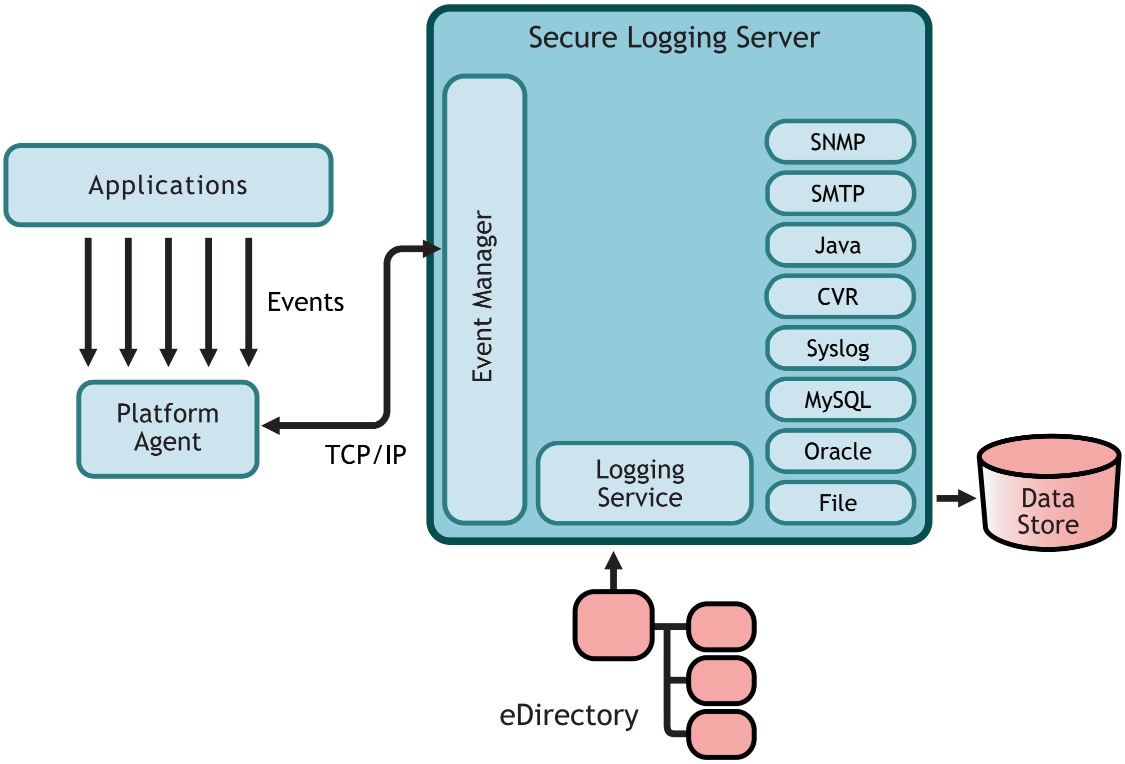


Logging Service

The Logging Service is the only Nsure Audit component that can write to the data store. This design protects log data from unauthorized record modification, insertion, or deletion.

To ensure that the auditing system maintains accurate records, the Event Manager delivers all incoming data directly to the Logging Service. All events and requests must be recorded in the data store before the Event Manager sends them to the Monitoring or Notification services.

Write times vary per storage option. On a P4 Xeon class server, the Logging Service can write approximately 60,000 events per second to a flat file in a file system or 3,000 events per second to a MySQL* database.



Using the logging server's channels, the Logging Service can write events to the following storage devices:

- Flat file in the file system
- MySQL database
- Oracle database
- Syslog database
- Microsoft* SQL Server database

NOTE: Although you can actually use any channel to log events, for performance reasons we recommend that you only log to the Syslog, MySQL, Oracle, Microsoft SQL Server, or File channels.

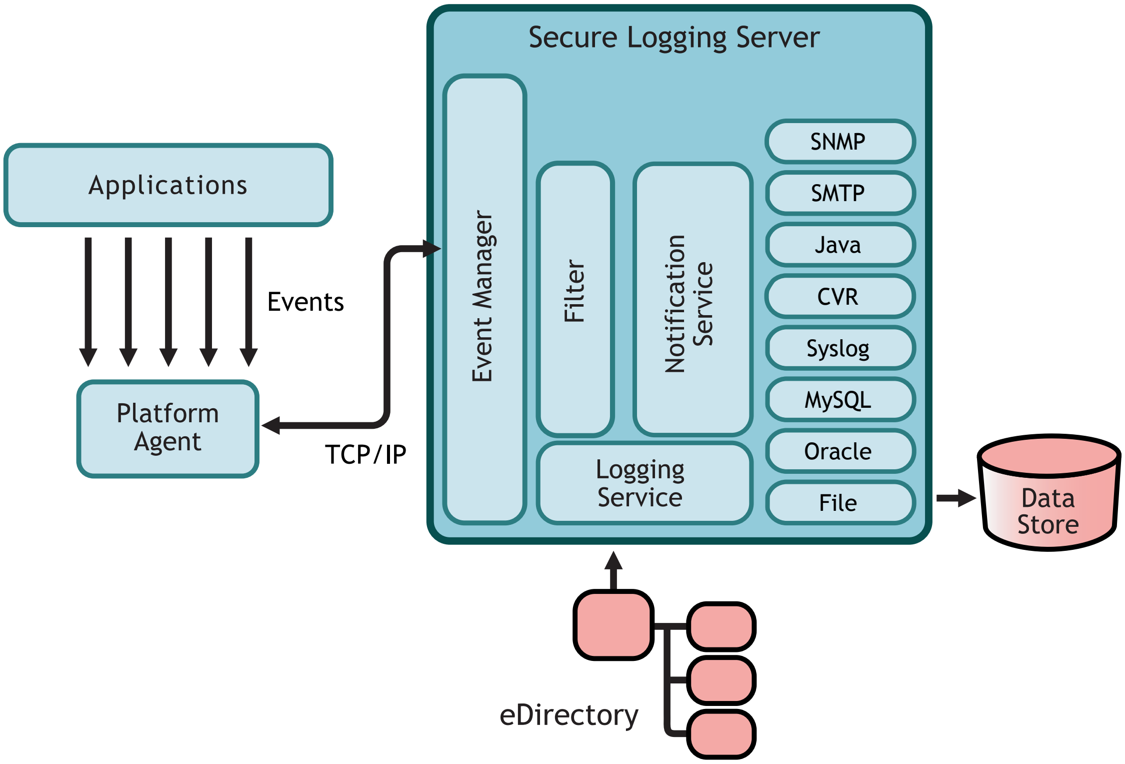
For more information on configuring these channels, see Chapter 7, “Configuring System Channels,” on page 69.

Notification Service

The Notification Service provides two kinds of notification: filtered and heartbeat. Filtered notification tells you when a specific event has occured; heartbeat notification tells you when an event has not occured.

In both cases, the Notification Service identifies the event and routes it to specific channels. For example, the Notification Service can send a notification event to a system administrator's mailbox or cell phone using the SMTP channel, route the event to the network management system as an SNMP trap, or write the event to a flat file in the file system or to a Syslog, MySQL, Oracle, or SQL Server database.

Both filtered and heartbeat notifications are configured in eDirectory using Notification Filter and Heartbeat objects. These objects define event criteria and designate which Channel objects are used to provide event notification. For more information, see "Notification Objects" on page 32.



2.1.3 Data Store

Using its available channel drivers, Nsure Audit can log events to the following storage devices:

- Flat file in the file system
- MySQL database
- Oracle database
- Microsoft SQL Server database
- Syslog database

Nsure Audit protects log data from record modification, insertion, or deletion by allowing only one program component, the Logging Service, to write events to the data store. Nsure Audit also limits read access to log data by controlling which applications can request log information through the auditing system. However, the security of the data store itself is up to you and the security mechanisms provided by the database. Although Nsure Audit maintains an internal security perimeter around the data store, it is possible for individuals or applications to directly access the data store outside the boundaries of the auditing system. Therefore, file system rights, directory rights, and the database's internal security features must be carefully configured to secure your log data.

For further information, see Section 4.3, "Configuring the Data Store," on page 54.

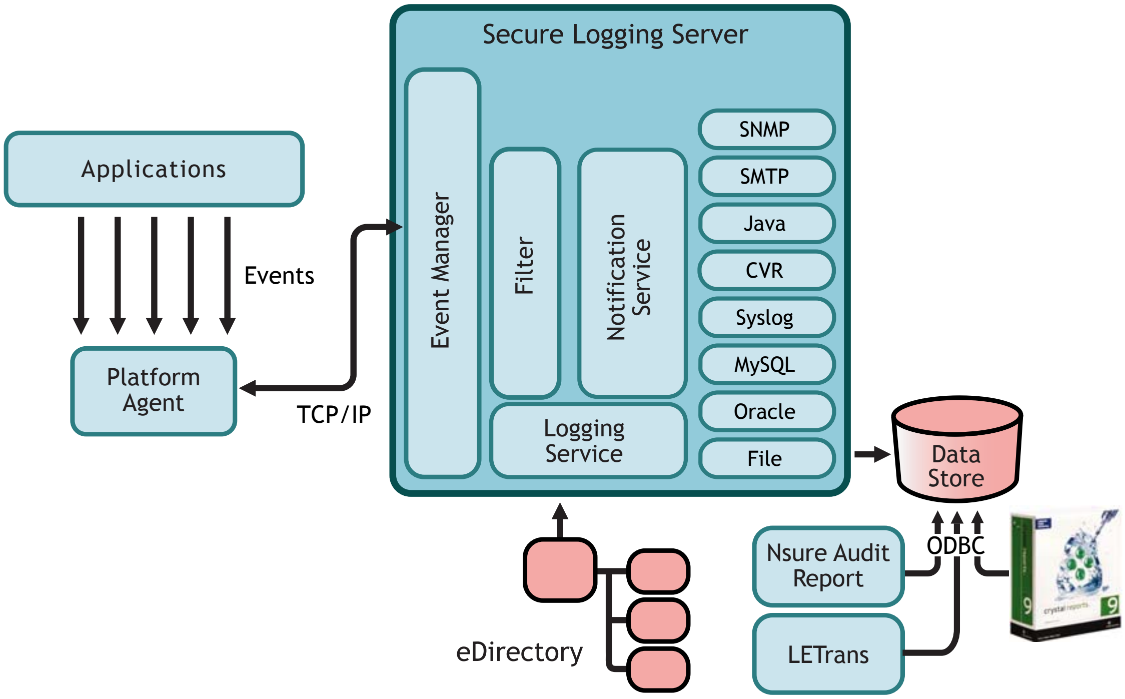
2.1.4 Reporting Applications

Nsure Audit provides two tools that can be used to generate reports from MySQL, Microsoft SQL Server, and Oracle data stores.

NOTE: Any standardized syslog reporting tool can be used to generate reports from syslog data stores.

- Nsure Audit Report is a Windows-based, ODBC-compliant application that can generate reports from Oracle and MySQL data stores. It includes predefined reports and can be integrated with Crystal Reports* to provide full custom reporting capabilities.
- iManager is a browser-based, JDBC*-compliant application that can generate reports from MySQL data stores.

For more information on generating reports with these tools, see Chapter 9, "Generating Queries and Reports," on page 107. Any standardized syslog reporting tool can be used to generate reports from syslog data stores.



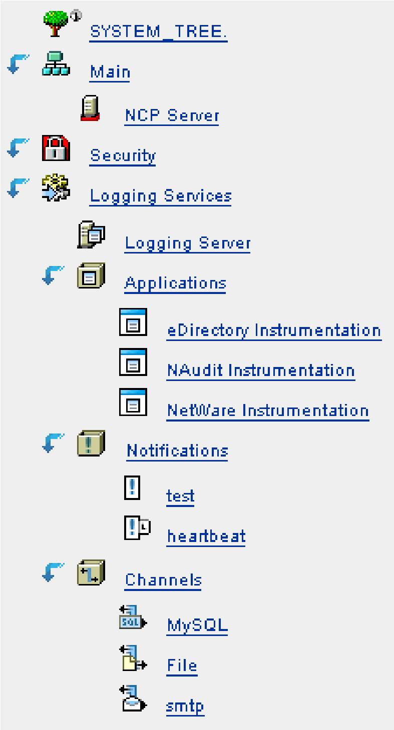
2.2 Configuration Objects

When you install the Secure Logging Server, the installation program extends the eDirectory schema to include the following objects:

- Logging Services Container (page 29)

- Logging Server Object (page 30)
- Nsure Audit Attributes on the NCP Server Object (page 30)
- Application Objects (page 30)
- Channel Objects (page 31)
- Notification Objects (page 32)

Nsure Audit uses these objects to store and look up system configuration parameters.



IMPORTANT: The Platform Agent is not configured through eDirectory. Instead, the Platform Agent's configuration settings are stored in a simple, text-based configuration file (logevent). For more information, see "Logevent" on page 42.

2.2.1 Logging Services Container

During your initial installation, Nsure Audit extends the eDirectory schema and creates the Logging Services container at the root of your directory tree. Because it is part of Nsure Audit, there can only be one Logging Services container per tree and, as the logging system container, it only contains Nsure Audit component objects.

Locating all logging system components in the Logging Services container is ideal for organizations that need a simple, easy-to-manage logging system. It also suits organizations that are implementing Nsure Audit as an auditing solution and, for security reasons, want to centrally manage their system. To facilitate distributed administration, however, Nsure Audit components can also be created and managed outside the Logging Services container.

If the Logging Services container is deleted, it can only be re-created by re-running AuditExt. For more information, see Section G.5, “AuditExt,” on page 240.

2.2.2 Logging Server Object

In eDirectory, the Logging Server object represents the physical server where you installed the Secure Logging Server. However, because the Logging Server object is specific to Nsure Audit, it does not replace the NCP Server object. Instead, each Logging Server object is associated with an NCP Server object.

The Logging Server object is represented as a container with server attributes; it can contain Nsure Audit objects and it stores all the properties and attributes for the Secure Logging Server. For information on creating and configuring the Logging Server object, see Section 4.2, “Configuring the Secure Logging Server,” on page 44.

2.2.3 Nsure Audit Attributes on the NCP Server Object

During installation, Nsure Audit extends the definition of the NCP Server object to include the log settings for eDirectory, NetWare, traditional file system, and NSS events. These settings are found under the NCP Server object's Nsure Audit tab.

The Nsure Audit screen has separate menus for NetWare, Filesystem, and eDirectory events. Each menu lists the events that fall in its respective category. To configure NetWare, Filesystem, or eDirectory instrumentation to log a particular type of event, simply mark the event's check box and click Apply. The instrumentation automatically begins logging the marked events to the Secure Logging Server.

NOTE: You do not need to restart the logging server to effect changes to NSure Audit attributes in the NCP Server object.

For more information on configuring the NCP Server object's Nsure Audit attributes, see Chapter 5, “Logging eDirectory, NetWare, and File System Events,” on page 61.

2.2.4 Application Objects

Application objects are associated with applications that log to or request information from Nsure Audit. These objects store the information required by the logging server to authenticate logging applications. They also identify which users have rights to monitor the applications' events and they store the applications' log schemas.

NOTE: The log schema catalogs the events that can be logged for a given application. For more information, see Section A.4, “Log Schema Files,” on page 187.

Application objects are usually created automatically when either Nsure Audit or the logging application is installed. If necessary, they can also be manually added to the tree using iManager.

During installation, Novell Nsure Audit automatically creates Application objects for itself (the Naudit Instrumentation), the eDirectory Instrumentation, and the NetWare Instrumentation.The Naudit Instrumentation allows Nsure Audit to audit its own events such as creating Channel or Notification objects. The eDirectory Instrumentation manages logging of eDirectory events and the NetWare Instrumentation provides logging for NetWare and file system events.

NOTE: The NetWare Instrumentation is only installed on NetWare versions.

Application objects can be created only within Application containers. Novell Nsure Audit creates the Application objects for the Naudit, eDirectory, and NetWare Instrumentations in the Application container under Logging Services.

For more information on creating and configuring Application objects, see Chapter 6, “Managing Applications that Log to Nsure Audit,” on page 65.

Application Containers

Application containers provide a reference point through which the logging server can locate Application objects. At startup, the logging server scans its list of Application containers and loads the included Application object configurations in memory where it can quickly access the information when authenticating applications. For information on configuring the Application Container property on the logging server, see “Logging Server Objects” on page 45.

IMPORTANT: The logging server scans its list of Application containers only at startup. Therefore, if you create or modify an Application object, you must restart the logging server. For information on restarting the logging server, see Section G.3, “Secure Logging Server Startup Commands,” on page 236.

The Application container under Logging Services is automatically created during installation; however, additional Application containers can be created anywhere in the tree.

2.2.5 Channel Objects

Channel objects store the information the logging server needs to use channel drivers. For example, a MySQL Channel object contains the IP address or host name of the MySQL database server; a username and password for connecting to the server, the name of the database and table, and any other relevant information. An SMTP Channel object, on the other hand, includes the address of the SMTP server; a username and password; and the recipient, sender, subject, and body of the log message.

Nsure Audit is designed so you can create multiple Channel objects for any given channel. This means you can apply different channel configurations to different functions or events. For instance, you can configure the logging server to use one MySQL Channel object to add events to the central data store and configure a Notification Filter to use another MySQL Channel object to create a filtered log.

The available types of Channel objects are:

SMTP	Oracle (Only available on NetWare using the JDBC Java channel.)

SNMP	File
Java	Syslog
MySQL	CVR
JDBC	Microsoft SQL Server

Additional Channel objects can be easily incorporated in this model. For more information, see the Nsure Audit SDK (http://developer.novell.com/ndk/naudit.htm).

Of particular note is the Critical Value Reset (CVR) Channel object. In configuring a CVR Channel object, you can flag an attribute in eDirectory with a reset policy. If the value of that specific attribute is changed, the CVR channel automatically resets the value as per the policy defined in the CVR Channel object.

The logging server looks for Channel objects only in Channel containers; therefore, Channel objects can only be created within Channel containers. For information on creating and configuring Channel objects, see Chapter 7, "Configuring System Channels," on page 69.

Channel Containers

Channel containers provide a reference point through which the logging server can locate Channel objects. At startup, the logging server scans its list of Channel containers and loads the included Channel object configurations and their drivers. The drivers and Channel object configurations are then available to provide event notification and to log events. Note that the logging server only loads those drivers that have Channel objects in supported Channel containers. For information on configuring the Channel Container property on the logging server, see "Logging Server Objects" on page 45.

IMPORTANT: The logging server scans its list of Channel containers only at startup. Therefore, if you create or modify a Channel object, you must restart the logging server. For information on restarting the logging server, see Section G.3, "Secure Logging Server Startup Commands," on page 236.

The Channel container under Logging Services is automatically created during installation; however, Channel containers can be created anywhere in the tree.

2.2.6 Notification Objects

Nsure Audit provides two kinds of event notification:

- Filtered Notification
- Heartbeat Notification

Filtered notification tells you when a specific event has occured; heartbeat notification tells you when an event has not occured. The following sections discuss the objects associated with each notification.

Notification Filter Objects

Notification Filter objects store the criteria the logging server uses to filter system events. They also designate which Channel objects the logging server uses to provide event notification.

When you define a Notification Filter, you specify a value for a given event field. To narrow the results, you can define values for multiple event fields. Using standard “and,” “or,” and “not” operators, you can define up to 15 event conditions. For more information on the event fields, see Section A.1, “Event Structure,” on page 175.

After you define the filter criteria, you must select the object's notification channel. Notification channels are simply the Channel objects the logging server uses to provide event notification. For example, if you want to e-mail filtered events to your mailbox, you must select an SMTP Channel object that is configured to relay events to your e-mail address. Similarly, if you want to log filtered events to a MySQL database, you must select a MySQL Channel object that is configured to write events to the correct database and table. You can define multiple notification channels for any given Notification Filter.

The logging server looks for Notification Filter objects only in Notification containers; therefore, Notification Filter objects can be created only within Notification containers. For information on creating and configuring Notification Filter objects, see Chapter 8, “Configuring Filters and Event Notifications,” on page 101.

Heartbeat Objects

Heartbeat objects define which Event IDs the logging server looks for and the interval at which those events must occur. If an event does not occur within the designated interval, the logging server generates a heartbeat event.

The heartbeat event is automatically logged to the central data store; however, if you want to receive notification that a specific event has not occurred, you must create a Notification Filter for the corresponding heartbeat event.

The logging server looks for Heartbeat objects only in Notification containers; therefore, Heartbeat objects can be created only within Notification containers. For information on creating and configuring Heartbeat objects, see Chapter 8, “Configuring Filters and Event Notifications,” on page 101.

Notification Containers

Notification containers provide a reference point through which the logging server can locate Notification objects. At startup, the logging server scans its list of Notification containers and loads the included Notification object configurations in memory where it can quickly access the information to filter or monitor events. For information on configuring the Notification Container property on the logging server, see “Logging Server Objects” on page 45.

IMPORTANT: The logging server scans its list of Notification containers only at startup. Therefore, if you create or modify a Notification object, you must restart the logging server. For information on restarting the logging server, see Section G.3, “Secure Logging Server Startup Commands,” on page 236.

The Notification container under Logging Services is automatically created during installation; however, Notification containers can be created anywhere in the tree.

Nsure Audit iManager Plug-in

3

Novell® Nsure™ Audit is managed and configured 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. With the Nsure Audit iManager plug-in, iManager can be used to manage Nsure Audit objects in eDirectory.

The Nsure Audit iManager plug-in is included with the Nsure Audit installation.

- Section 3.1, “System Requirements,” on page 35
- Section 3.2, “Opening iManager,” on page 35
- Section 3.3, “iManager Interface,” on page 35
- Section 3.4, “Performing Basic Administrative Functions in iManager,” on page 37
- Section 3.5, “Securing Your iManager Connection,” on page 39

3.1 System Requirements

Installing and using the Nsure Audit iManager Plug-in requires iManager 2.5. See the Novell iManager Product Page (http://www.novell.com/products/consoles/imanager/index.html) for requirements and download instructions.

3.2 Opening iManager

1 Access 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

2 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.

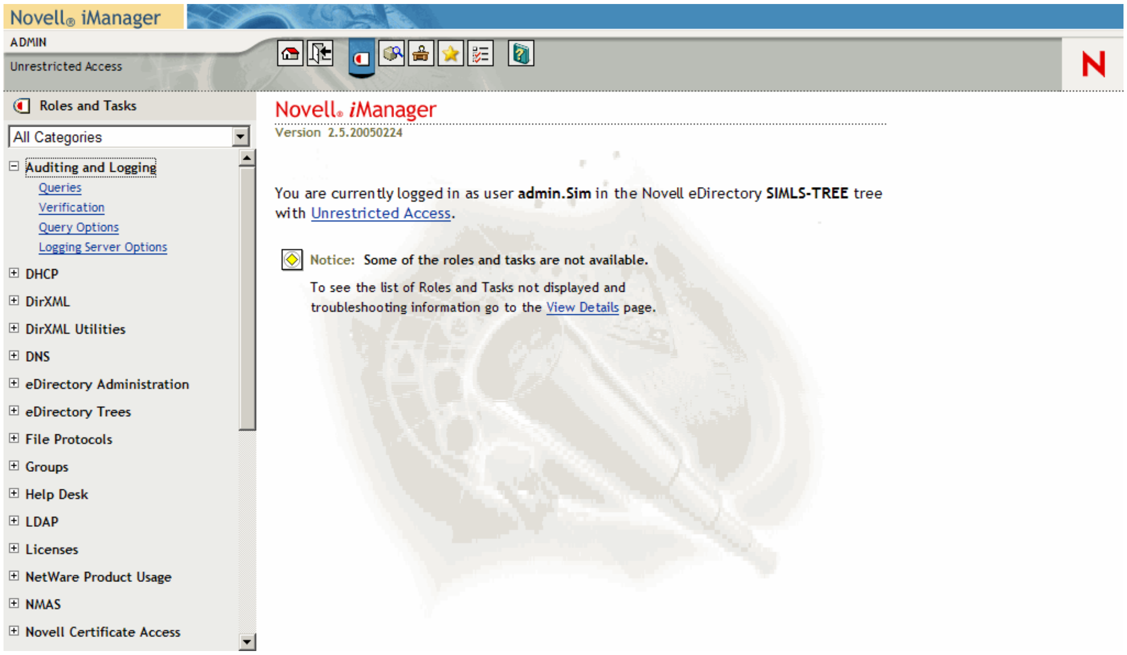
3.3 iManager Interface

The iManager interface has two administrative views: the Roles and Tasks view and the Object view. Both views are task driven.

The Roles and Tasks view provides a list of available roles and their associated tasks. The user does not need to browse the tree to find an object to administer; instead, the plug-in for that task presents the necessary tools and interface to perform the task.

NOTE: You only have access to those tasks for which you have assigned rights.

Figure 3-1 *iManager Role and Task View*



You are currently logged in as user **admin.Sim** in the Novell eDirectory **SIMLS-TREE** tree with Unrestricted Access.

Notice: Some of the roles and tasks are not available.

To see the list of Roles and Tasks not displayed and troubleshooting information go to the View Details page.

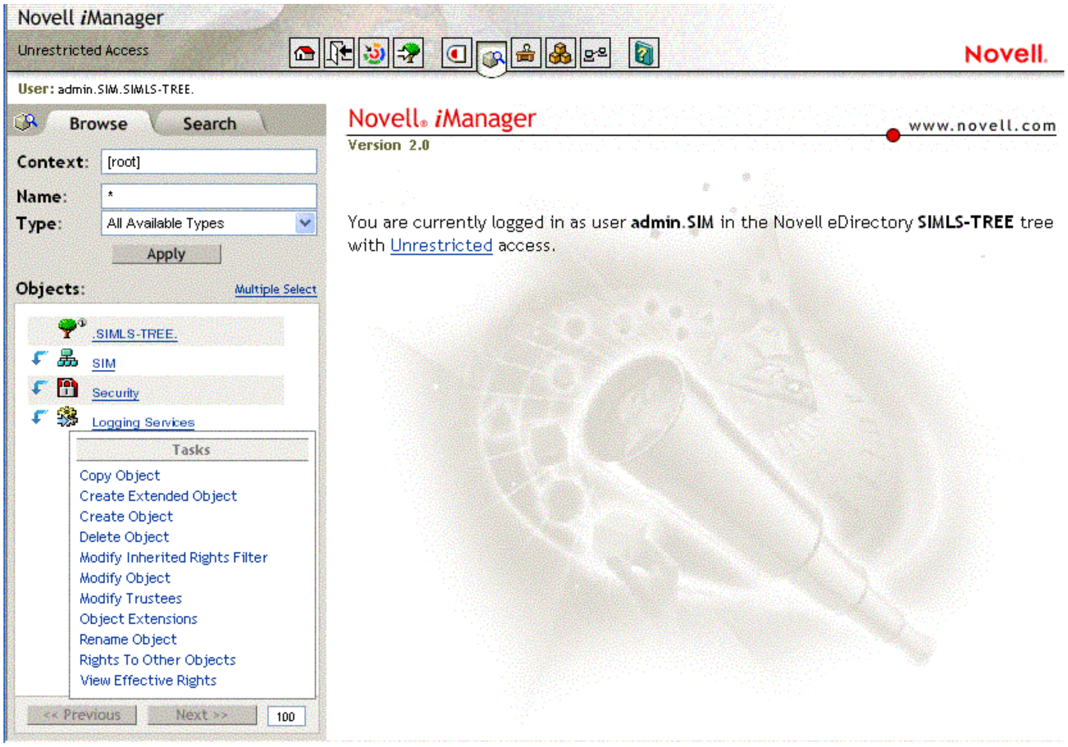
The Auditing and Logging Role includes the following tasks:

Task	Description
Queries	Guides you through the process of defining and running queries. For more information, see Section 9.1, “Using iManager to Generate Queries,” on page 107.
Verification	Verifies that the events logged to the data store for each logging application are authentic; that is, it can validate the event signatures to determine if an application’s events have been tampered with, deleted, or if the sequence of events has been changed. For more information, see Section 9.1.6, “Verifying Event Authenticity in iManager,” on page 127 and Section 9.2.7, “Verifying Event Authenticity in Nsure Audit Report,” on page 140.
Query Options	Allows you to define your available databases, import log schemas, and set general report settings such as query limits and default sort order. For more information, see Section 9.1, “Using iManager to Generate Queries,” on page 107.
Logging Server Options	Allows you to configure a specific logging server. You can also create Notification, Channel, and Application objects in the logging server’s supported containers. For more information, see Section 4.2, “Configuring the Secure Logging Server,” on page 44.

The Object view displays objects by context. To move up or down in the tree, click the navigation arrows. You can also navigate the tree by specifying a specific context in the *Context* field.

When you select an object, iManager displays the tasks that can be performed on that object. Select a task to open the corresponding menu.

Figure 3-2 iManager Object View



User: admin.SIM.SIMLS-TREE.

Context: [root]

Name: *

Type: All Available Types

You are currently logged in as user **admin.SIM** in the Novell eDirectory **SIMLS-TREE** tree with Unrestricted access.

IMPORTANT: Do not use the browser's Back and Forward buttons while using iManager. Because iManager is a Web-based application, it is important to navigate through the interface using the buttons inside the application, not the buttons on the browser's toolbar.

3.4 Performing Basic Administrative Functions in iManager

This section reviews how to perform the following administrative functions:

- Creating Objects in iManager
- Renaming Objects in iManager
- Deleting Objects in iManager
- Modifying Object Attributes in iManager

IMPORTANT: Do not use the browser's Back and Forward buttons while using iManager. Because iManager is a Web-based application, it is important to navigate through the interface using the buttons inside the application, not the buttons on the browser's toolbar.

3.4.1 Creating Objects in iManager

1 Click the *View Objects* button on the iManager toolbar.

2 In the Object view, select the container where you want to create the object.

3 Select *Create Object* from the Task list.

4 In the Create Object page, select the type of object you want to create then click *OK*.

5 Specify the object name.

6 When finished, click *OK*.

3.4.2 Renaming Objects in iManager

1 Click the *View Objects* button on the iManager toolbar.

2 In the Object view, select the object you want to rename.

3 Select *Rename Object* from the Task list.

4 Specify the new object name.

IMPORTANT: Do not include the context with the new object name.

5 If you want, you can select the rename options.

5a The *Save old name* option saves the original name as an attribute on the object.

5b The *Create an alias in place of renamed object* option renames the current object and creates an alias of that object using the original name.

6 When finished, click *OK*.

3.4.3 Deleting Objects in iManager

1 Click the *View Objects* button on the iManager toolbar.

2 In the Object view, select the object you want to delete.

3 Select *Delete Object* from the Task list.

4 Click *OK* to delete the object.

3.4.4 Modifying Object Attributes in iManager

1 Click the *View Objects* button on the iManager toolbar.

2 In the Object view, select the object you want to modify.

3 Select *Modify Object* from the Task list.

The configuration page displays the object's attributes.

4 Modify the object's attributes.

IMPORTANT: If you modify attributes in multiple tabs, you must click *Apply* in each page to save your changes.

5 When finished, click *OK*.

3.5 Securing Your iManager Connection

When you log in to iManager, your connection is automatically forwarded to a secure port. The default HTTPS port for iManager is 443.

For more information on running iManager over an SSL connection, see "Configuring and Using SSL for LDAP Connections" in the *iManager Administration Guide*. (http://www.novell.com/documentation/lg/imanager151/imanager/data/adbcvpt.html)

Configuring the Logging System

4

The base components in the Novell® Nsure™ auditing system are the Platform Agent, the Secure Logging Server, and the data store. This section provides the information you need to configure and manage these components.

4.1 Configuring the Platform Agent

The Platform Agent, logevent, is the client portion of the Nsure auditing system. It receives logging information and system requests from authenticated applications and transmits the information to the Secure Logging Server.

For more information on program binaries, see Section H.1, "Program Files and Directories," on page 245. For information on how applications authenticate with Nsure Audit, see Section 10.1, "Authenticating Logging Applications," on page 159.

There are several advantages to having applications connect to the Platform Agent instead of the Secure Logging server:

- It is easier for applications to plug into the Nsure Audit system because the complexity and overhead of communicating with the logging server is off-loaded to the Platform Agent.
- Centralizing the communication load makes the system more efficient.
- The Platform Agent and logging applications are on the same computer, giving you secure, uninterrupted logging. If the connection between the Platform Agent and the Secure Logging Server fails, the Platform Agent simply switches into Disconnected Cache Mode.

4.1.1 Disconnected Mode Cache

If the connection between the Platform Agent and the Secure Logging Server fails, applications continue to log events to the local Platform Agent, just as they always do. The Platform Agent simply switches into Disconnected Cache mode; that is, it begins sending events to the Logging Cache module. The Logging Cache module then writes the events to the Disconnected Mode Cache until the connection is restored. The switch into Disconnected Cache Mode is completely transparent to the logging applications.

NOTE: The port at which the Platform Agent connects to the Logging Cache Module is configured in the logevent.cfg file. For more information on this parameter, see "Logevent" on page 42.

The Logging Cache Module maintains a separate cache file for each authenticated application. The cache files include the authentication credentials as well as the log events for their respective applications.

When the connection to the Secure Logging Server is restored, the Logging Cache Module transmits the cache files to the Secure Logging Server. To protect the integrity of the data store, the Secure Logging Server validates the authentication credentials in each cache file before logging its events.

4.1.2 Logevent

The Platform Agent is not configured through Novell eDirectory™. Instead, the Platform Agent's configuration settings are stored in a simple, text-based configuration file, `logevent`.

The `logevent` file is stored in the following directories:

Table 4-1 *Platform Agent Configuration File*

Operating System	File
NetWare	`/etc/logevent.cfg`
Linux	`/etc/logevent.conf`
Solaris	`/etc/logevent.conf`
Windows	/*Windows_Directory*/logevent.cfg The *Windows_Directory* is usually *drive*:\windows.

Storing the Platform Agent's configuration in a local text file makes the Platform Agent small, unobtrusive, and self-contained—that is, it has no external dependencies, so it is always available to receive logged events. Storing the Platform Agent's configuration in a text based file also allows the Platform Agent to eventually run on platforms that do not have eDirectory support.

The following is a sample `logevent` file.

```
LogHost=127.0.0.1
LogCacheDir=c:\logcache
LogCachePort=288
LogEnginePort=289
LogCacheUnload=no
LogReconnectInterval=600
LogDebug=never
LogSigned=always
```

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

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

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

Table 4-2 `logevent` *Settings*

Setting	Description
LogHost=*dns_name*	Name or IP address of the Secure Logging Server the Platform Agent should use. If you are configuring multiple Secure Logging Servers, add the IP address of each logging server separated with commas to the LogHost entry. For example, `LogHost=192.168.0.1,192.168.0.3,192.168.0.4` With this modification, the Platform Agents log specifically to the group of logging servers that they are a member of, regardless of the status of the servers. For more information, see Section 4.2.3, "Configuring Multiple Secure Logging Servers," on page 48.
LogCacheDir=*path*	The directory where the Platform Agent should store the cached event information if the Primary or Secondary Secure Logging Server becomes unavailable.
LogEnginePort=*port*	Port used by the Secure Logging Server to accept data from Platform Agents.
LogCachePort=*port*	Port used by the Platform Agent caching mechanism.
LogCacheUnload=Y|N	Set to `N` if lcache should not allow unloading
LogCacheSecure=Y|N	If the local cache file should be encrypted, this option must be set to `Y`.
LogReconnectInterval=*seconds*	The interval, in seconds, at which the Platform Agent and the Platform Agent Cache try to reconnect to the Secure Logging Server if the connection is lost.
LogDebug=Never|Always|Server	The Platform Agent debug setting. • Set to `Never` to never log debug events. • Set to `Always` to always log debug events.
LogSigned=Never|Always|Server	The signature setting for Platform Agent events. • Set to Never to never sign or chain events. • Set to Always to always log events with a digital signature and to sequentially chain events. **NOTE:** Event signing can significantly impact program execution and CPU utilization on some systems. For more information on event signatures, see Chapter 10, "Security and Non-Repudiation," on page 159.
LogMaxBigData=*bytes*	The maximum size of the event data field. The default value is 3072 bytes. Set this value to the maximum number of bytes the client allows. Data that exceeds the maximum is truncated or not sent if the application doesn't allow truncated events to be logged.
LogMaxCacheSize=*bytes*	The maximum size, in bytes, of the Platform Agent cache file.

Setting	Description
LogCacheLimitAction=stop logging|drop cache	The action that you want the cache module to take when it reaches the maximum cache size limit. • Set to `stop logging` if you want to stop collecting new events. • Set to `drop cache` if you want to delete the cache and start over with any new events that are generated.

4.1.3 Platform Agent Configuration Tool

The Platform Agent Configuration Tool is a Java* utility that provides a graphical interface to manage Nsure Audit Platform Agents. This tool operates by making changes to the `logevent.cfg` file, which contains configuration settings for the Platform Agent.

IMPORTANT: You must have Java installed on the server where the Platform Agent Configuration Tool is installed to use the utility.

To make configuration changes, you can either open and edit an existing `logevent.cfg` configuration file, or create a new `logevent.cfg` file. When your changes are complete, the updated file must be saved in the correct location for your changes to be applied.

To run the Platform Agent Configuration Tool:

1. Locate the Platform Agent Configuration tool Java Archive file (`.jar`). By default it is installed in the following location:

Operating System	Path
NetWare	`sys:\system\nauditpaconfig.jar`
Windows	`\program files\novell\nsure audit\nauditpaconfig.jar`
Linux	`/opt/novell/naudit/java/nauditpaconfig.jar`
Solaris	`/opt/NOVLnaudit/java/nauditpaconfig.jar`

2. Launch the Platform Agent Configuration tool by executing the following command at a console from the directory where the Platform Agent Configuration tool Java Archive file is located:

```
java -jar nauditpaconfig.jar
```

4.2 Configuring the Secure Logging Server

The Secure Logging Server, the server component in the Nsure auditing system, is implemented in the shared library, lengine. For more information on program binaries, see Section H.1, "Program Files and Directories," on page 245.

The Secure Logging Server manages the flow of information to and from the Nsure auditing system. It receives incoming events and requests from the Platform Agents, logs information to the data store, monitors designated events, and provides filtering and notification services. It can also be configured to automatically reset critical system attributes according to a specified policy.

The Secure Logging Server is configured through eDirectory. The Logging Server object in eDirectory stores all the properties and attributes for the Secure Logging Server. Consequently, the server must have access to eDirectory and the Logging Server object before it can launch the Secure Logging Server.

To minimize server reaction time and ensure high system performance, you should create a local replica of the Logging Server object and its associated objects on the logging server.

4.2.1 Creating the Logging Server Object

On NetWare, Linux, and Solaris systems, the Logging Server object can be created automatically in the Logging Services container during installation or it can be created manually anywhere in the tree after installation. Which option you choose depends, to a degree, on your system design.

NOTE: The Windows installation program automatically creates the Logging Server object in the Logging Services container.

Locating the Logging Server object in the Logging Services container is ideal for organizations that need a simple, easy-to-manage logging system. It also suits organizations that are implementing Nsure Audit as an auditing solution and, for security reasons, want to centrally manage their systems. Using the Configure Logging Server option, these systems are immediately operational after install.

Conversely, to facilitate distributed system administration, Logging Server objects can be created and managed outside the Logging Services container. In fact, in distributed environments, Logging Servers and their associated objects should be created in a context assigned to their administrator. To create Logging Server objects outside of the Logging Services container, you must manually create the objects using iManager.

For information on creating objects in your respective administrative tool, see Chapter 3, “Nsure Audit iManager Plug-in,” on page 35.

4.2.2 Logging Server Objects

The Logging Server object represents the physical server where you have installed the Secure Logging Server. However, because the Logging Server object is specific to Nsure Audit, it does not replace the NCP Server object. Instead, each Logging Server object is associated with a host NCP Server object.

The Logging Server object is represented as a container with server attributes: it can contain Nsure Audit objects and it stores all the properties and attributes for the Secure Logging Server.

The following table provides an explanation of the Logging Server object's attributes.

Attribute	Description
Configuration	
Host Server	The distinguished name of the NCP Server object associated with the current logging server. Click the Object Selector button 🔍 to select the Host Server in the tree.

Attribute	Description
Driver Directory	The directory in which the channel drivers (lgd*) are located. The default channel driver directories are as follows: • sys:\system\ (NetWare) • \program files\novell\nsure audit\ (Windows) • /opt/novell/naudit/ (Linux) • /opt/NOVLnaudit/ (Solaris)
Log Channel	The Channel object the logging server uses to create the central data store. Click the Object Selector button 🔍 to select the Channel object in the tree. **WARNING:** The JDBC and Java channels do not work on NetWare 5.x. These channels require JVM 1.4.2 which is not compatible with NetWare 5.x. Attempting to run either the JDBC or Java channel on NetWare 5.x abends the server.
Secure Logging Certificate File	The path and filename for the Logging Server Certificate. This attribute enables the logging server to use a custom certificate created with the AudCGen utility. If this field is left blank, the logging server uses the default embedded certificate. **IMPORTANT:** Nsure Audit only recognizes certificates that are generated with the AudCGen utility. For information on generating custom certificates, see Section 10.3, "Creating the Secure Logging Certificate," on page 162. NSure Audit uses certificates to authenticate client connections. The logging server only accepts connections from applications that have a valid Logging Application Certificate. For general information on how certificates are used in Nsure Audit, see Chapter 10, "Security and Non-Repudiation," on page 159.
Secure Logging Privatekey File	The path and filename for the Secure Logging Certificate's private key file. If this field is left blank, the logging server assumes the private key is included with the certificate and uses the path and filename for the Secure Logging Certificate. Again, this is only required if you do not use the Nsure Audit program's embedded certificates. **IMPORTANT:** Nsure Audit only recognizes certificates and private keys that are generated with the AudCGen utility. For more information, see Section 10.3, "Creating the Secure Logging Certificate," on page 162.
Containers	**IMPORTANT:** The logging server scans these containers only at startup. Therefore, if you add a container, you must restart the logging server. For information on restarting the Secure Logging Server, see Section G.3, "Secure Logging Server Startup Commands," on page 236.

Attribute	Description
Application Containers	The Application containers supported by the current Logging Server object. Application containers provide a reference point through which the logging server can locate Application objects. Application containers must be included in this list for the logging server to locate their associated Application objects. For more information on Application containers and objects, see "Application Objects" on page 30. The Application container in Logging Services is added to this list by default.
Notification Containers	The Notification containers supported by the current Logging Server object. Notification containers provide a reference point through which the logging server can locate Notification Filter and Heartbeat objects. Notification containers must be included in this list for the logging server to locate their associated Notification objects. For more information, see "Notification Objects" on page 32. The Notification container in Logging Services is added to this list by default.
Channel Containers	The Channel containers supported by the current Logging Server object. Channel containers provide a reference point through which the logging server can locate Channel objects. Channel containers must be included in this list for the logging server to locate their associated Channel objects. For more information on Channel containers and objects, see "Configuring System Channels" on page 69. The Channel container in Logging Services is added to this list by default.
Memory	The memory configuration settings allow you to optimize your logging server's performance. You should adjust these settings based on logging traffic and the amount of memory available to your system. Reasonable values depend on your network. In organizations that require high-performance logging, these parameters should be set high enough to accommodate peak loads. For organizations that must minimize potential data loss, these settings should be very small. Although this might slow performance, it minimizes the amount of data that might be lost in the event of server failure. If incoming log events exceed the amount of memory you have allocated on your logging server, the Platform Agents temporarily write events to their Disconnected Mode Caches until the logging server clears its cache. This prevents any logged events from being lost.
Minimum	The amount of memory the server automatically allocates at boot time to handle logging processes. Because allocating additional memory on the fly can slow down code execution, this setting should represent the minimum amount of memory needed to handle your system's baseline level of logging traffic. Pre-allocating the minimum amount of memory required by your system reduces additional blocking delays when the system is under high load and facilitates faster processing of incoming events.
Normal	The amount of memory the server can immediately allocate if logging traffic exceeds the Minimum memory setting.

Attribute	Description
Maximum	The maximum amount of memory that can be allocated to logging processes. Setting a maximum prevents Nsure Audit from monopolizing the server's resources. Ideally, this should be set close to the Normal memory setting. If logging traffic exceeds the Normal memory setting, the server incrementally increases the logging cache 4 KB at a time. (4 KB is the amount of memory required to process a single event.) When the Maximum memory allocation is reached, the server begins dropping Platform Agent connections. If the logging server drops its connection, the Platform Agent simply logs events to its Disconnected Mode Cache, thereby ensuring no information is lost. When free cache is available, the logging server once again accepts Platform Agent connections.
`Status`	This option allows you to enable or disable the Secure Logging Server. By default, the logging server is enabled. If you mark the Disabled option, you must either restart the server or manually unload the Secure Logging Server for this setting to become effective. Thereafter, the server cannot launch the Secure Logging Server (lengine) until you mark Enabled. For information on unloading the logging server, see Section G.3, “Secure Logging Server Startup Commands,” on page 236.

4.2.3 Configuring Multiple Secure Logging Servers

By default, the installation program creates the Secure Logging Server in the Logging Services container. The logging server then reads its channel and notification configuration information from the Channels.Logging Services and Notifications.Logging Services containers and loads the channels and notifications located within these containers.

If you want to provide system redundancy or load balancing in your logging system, you can create multiple Secure Logging Server objects for servers on the same platform in the default Logging Services container. In this way, all the logging servers load the same channels and send data to the same database.

However, if you want to log data to different databases (such as in a WAN environment); load different channels and notifications on each logging server; if you have an extremely large eDirectory tree; or if you are running Nsure Audit Secure Logging servers on multiple platforms, we recommend that you create separate eDirectory organizational units as containers for each Secure Logging Server’s configuration object.

If you want a combination of both configurations—for example, you want to provide system redundancy or load balancing in a WAN environment—you can create multiple eDirectory organizational units with multiple Secure Logging Server objects.

The following sections review how to implement multiple Secure Logging Servers in your logging system based on whether you want to locate the Secure Logging Server objects in the same container, different containers, or a combination of both:

- “Creating Multiple Secure Logging Server Objects in a Single eDirectory Container” on page 49
- “Creating Secure Logging Server Objects in Different eDirectory Containers” on page 50

- “Creating Multiple Secure Logging Server Objects in Multiple eDirectory Containers” on page 52

Creating Multiple Secure Logging Server Objects in a Single eDirectory Container

IMPORTANT: Creating multiple Nsure Audit Secure Logging Server objects in a single eDirectory container works best if the objects are use the same platform. If you create multiple Nsure Audit Secure Logging Server objects from different platforms in the same container, it can mix the servers’ configuration information. If you are running Nsure Audit Secure Logging servers on multiple platforms, we recommend that you create separate eDirectory organizational units as containers for each Secure Logging Server’s configuration objects. For more information, see Section , “Creating Secure Logging Server Objects in Different eDirectory Containers,” on page 50.

To provide redundancy in your logging system:

1. Create multiple Secure Logging Server objects in the default Logging Services container so that all the logging servers load the same channels and send data to the same database.
2. Configure the Platform Agents to shut down and start logging to another log server if the Secure Logging Server goes down.

   To configure the Platform Agents to connect to multiple logging servers in a failover manner:

   2a Add the IP address of each failover logging server to the LogHost entry in the `logevent.conf` (Linux and Solaris) or `logevent.cfg` (NetWare and Windows) file.

   2b List the logging servers in the order you want to use them in the event of a failover.

   The Platform Agents connect to the servers in the order specified in the configuration file. Therefore, if the first logging server goes down, the Platform Agent tries to connect to the second logging server, and so on.

   2c Separate each IP address in the LogHost entry with commas. For example, `LogHost=192.168.0.1,192.168.0.3,192.168.0.4`.
3. For a failover configuration, you might also consider running the database server on a server separate from the logging servers.

The `logevent` configuration files can also be modified for load balancing. A Nsure Audit logging server can log approximately about 5,000 events per second on a P4 Xeon class server to a MySQL database; however, the database is capable of processing much more data than that. If you run multiple logging servers with the same configuration from the same eDirectory container, all logging servers log to the same location and process the same notification list. As a group, the logging servers can then log much more data to the data store.

To configure load balancing:

1. Modify the LogHost entry in the `logevent.conf` (Linux and Solaris) or `logevent.cfg` (NetWare and Windows) file for each Platform Agent so that it logs to a different Secure Logging Server.

   For example, one Platform Agent sends data to 192.168.0.1, while another Platform Agent sends data to 192.168.0.3. In this way, you can maximize the number of events that can be logged to your data store.

Creating Secure Logging Server Objects in Different eDirectory Containers

Creating separate eDirectory organizational units as containers for each Secure Logging Server's configuration objects is an effective option under the following circumstances:

- If you want to log data to different databases (such as in a WAN environment)
- If you want to load different channels and notifications on each logging server
- If you have an extremely large eDirectory tree
- If you are running Nsure Audit Secure Logging servers on multiple platforms

The following sections review how to configure multiple Secure Logging Servers in different eDirectory containers.

- "Creating Multiple Containers for Audit Configuration Objects" on page 50
- "Configuring the Secure Logging Server to Use Custom Containers" on page 51

Creating Multiple Containers for Audit Configuration Objects

The ability to create organizational units is not a function of the Nsure Audit install utility. It must be done with iManager or another eDirectory management utility.

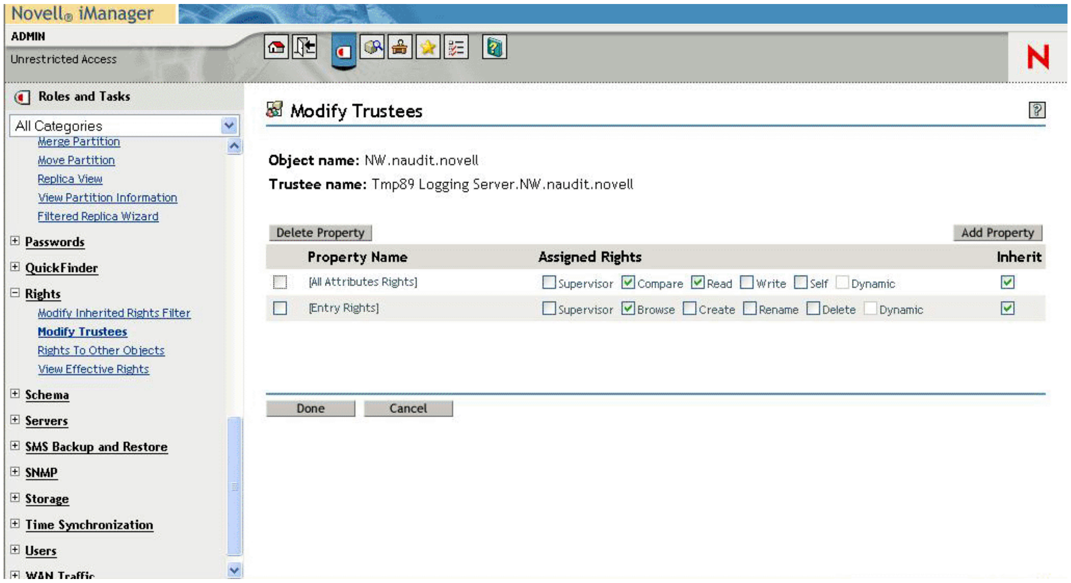
The following steps outline the process required to separate the Nsure Audit containers.

IMPORTANT: This procedure assumes that you installed Nsure Audit with the default containers.

1. Within iManager, create a new organizational unit for Nsure Audit objects (such as NAudit or Logging Services).
2. (Optional) If you are running Secure Logging Servers on multiple platforms, create organizational units for each platform——that is, NetWare, Windows, Linux, and Solaris—under the Naudit or Logging Services container.
3. Verify that eDirectory is healthy by checking replica synchronization.

   For more information, see TID10060600, *NDS / eDirectory Health Check Procedures - Cross Platform* (http://www.novell.com/support/search.do?cmd=displayKC&docType=kc&externalId=10060600&sliceId=&dialogID=3963716&stateId=0%200%203967974).
4. Use iManager's *Partitions and Replica Management* task to create partitions for all subcontainers and objects under the default Logging Services container—that is Channel, Application, Notification, and Logging Server objects—in preparation to move these objects.
5. Use iManager's *Partitions and Replica Management* task to move the Channel, Application, Notification, and Logging Server objects to the new organizational unit created in Step 2.
6. Use iManager's *Partitions and Replica Management* task to merge the partitions for the Channel, Application, Notification, and Logging Server objects back into the parent partition.
7. Use iManager's *Rights* tasks to add the Logging Server object as a Trustee of its parent organizational unit.

If you are running Secure Logging Servers on multiple platforms, the parent organizational unit is the associated platform container (NetWare, Windows, Linux, or Solaris).



Modify Trustees

Object name: NW.naudit.novell
Trustee name: Tmp89 Logging Server.NW.naudit.novell

	Property Name	Assigned Rights	Inherit
□	[All Attributes Rights]	□ Supervisor ☑ Compare ☑ Read □ Write □ Self □ Dynamic	☑
□	[Entry Rights]	□ Supervisor ☑ Browse □ Create □ Rename □ Delete □ Dynamic	☑

Each Secure Logging Server object needs Read and Compare attribute rights and Compare entry rights for its parent container and contents.

8 To verify that Audit is functioning correctly, unload lengine at the server, then reload it.

9 Remove the default Logging Services container at the root to prepare for the next Secure Logging Server installation.

After you remove the default Logging Services container, the tree is ready for you to install the next Secure Logging Server.

10 When the installation programs asks, “Use Default Container? Y” during subsequent Secure Logging Server installations, choose N and select the organizational unit you created in Step 2.

Configuring the Secure Logging Server to Use Custom Containers

To configure a logging server to use a custom channel container:

1 Click the *Roles and Tasks* button on the iManager toolbar.

2 In the Roles and Tasks view, click *Directory Administration* > *Create Object* > *Nsure Audit Channel Container* > *OK*.

3 On the Create Nsure Audit Channel Container page, specify a channel name and context, then click *OK*.

For the context, browse to and select the appropriate Logging Server object in the tree.

4 In Roles and Tasks, click *Auditing and Logging* > *Logging Server Options*, browse to and select the Logging Server object, then click *OK*.

5 Click the *Channels* tab or select *Channels* from the drop-down menu, depending on your browser.

6 Click *Container Actions* > *Add Container*.

7 On the Object Selector page, browse to the new container you created in the logging server container.

8. To remove the original Channels container from the logging server configuration, go to the Channels page, select the *Channels* check box, then click *Container Actions* > *Remove Container*.
9. Repeat this process for each logging server.

To configure a logging server to use a custom notification container:

1. In iManager, click *Roles and Tasks* > *Directory Administration* > *Create Object* > *Nsure Audit Notification Container* > *OK*.
2. On the Create Nsure Audit Notification Container page, specify a notification name and context, then click *OK*.

   For the context, browse to and select the appropriate Logging Server object in the tree.
3. In Roles and Tasks, click *Auditing and Logging* > *Logging Server Options*; browse to and select the Logging Server object, then click *OK*.
4. Click the *Notifications* tab or select *Notifications* from the drop-down menu, depending on your browser.
5. Click *Container Actions* > *Add Container*.
6. On the Object Selector page, browse to and select the new container created in the logging server container.
7. To remove the original Notifications container from the logging server configuration, select the *Notifications* check box in the Notifications page, then click *Container Actions* > *Remove Container*.
8. Repeat this process for each logging server.

Creating Multiple Secure Logging Server Objects in Multiple eDirectory Containers

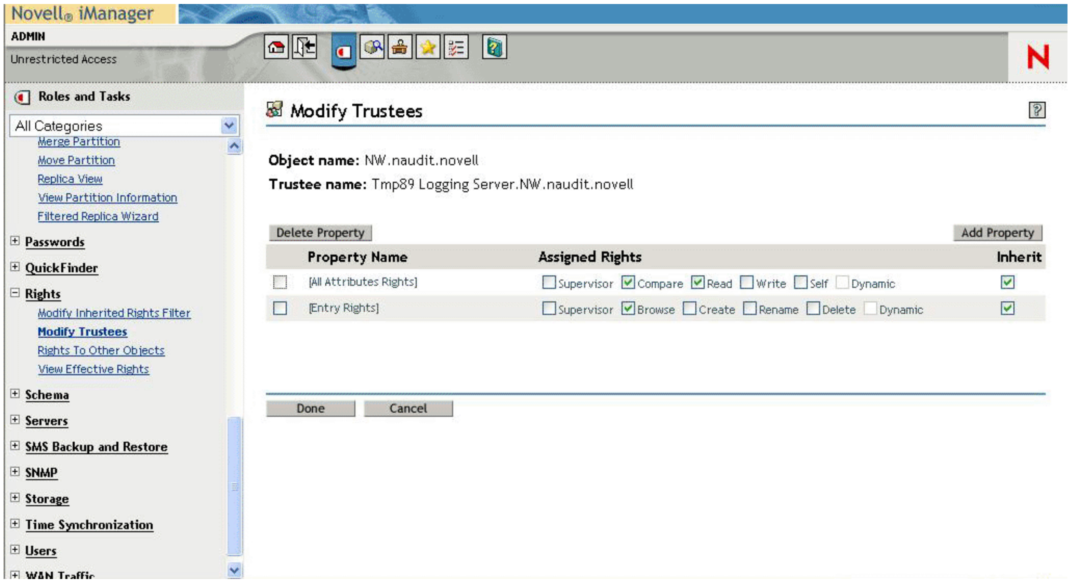
If you want to provide system redundancy or load balancing in a large eDirectory tree or a WAN environment, you must configure multiple Secure Logging Server objects in multiple eDirectory containers.

The following steps outline the general process required to implement this configuration:

1. Identify the Secure Logging Servers that should load the same channels and send data to the same location.
2. Create a custom Channels container in the default Logging Services container for each server group.

   For information on creating a custom Channels containers, see "Creating Multiple Containers for Audit Configuration Objects" on page 50.

3 Add the Secure Logging Server objects in each server group to the Trustees List for their associated Channels container.



Modify Trustees

Object name: NW.naudit.novell

Trustee name: Tmp89 Logging Server.NW.naudit.novell

	Property Name	Assigned Rights	Inherit
□	[All Attributes Rights]	□ Supervisor ☑ Compare ☑ Read □ Write □ Self □ Dynamic	☑
□	[Entry Rights]	□ Supervisor ☑ Browse □ Create □ Rename □ Delete □ Dynamic	☑

Each Secure Logging Server object needs Read and Compare attribute rights and Compare entry rights for the Channels container and its contents.

4 Configure each Secure Logging Server object's channel configuration so it references the server's associated Channel container.

5 Remove the original Channels.Logging Services container object from each Secure Logging Server object's channel configuration.

6 Configure the `logevent.conf` (Linux and Solaris) or `logevent.cfg` (NetWare and Windows) file for each Platform Agent so that it logs events to a specific server group.

To configure the Platform Agents to connect to multiple logging servers in a failover manner:

- Add the IP address of each failover logging server to the LogHost entry in the `logevent.conf` (Linux and Solaris) or `logevent.cfg` (NetWare and Windows) file.
- List the logging servers in the order you want to use them in the event of a failover.

  The Platform Agents connect to the servers in the order specified in the configuration file. Therefore, if the first logging server goes down, the Platform Agent tries to connect to the second logging server, and so on.
- Separate each IP address in the LogHost entry with commas. For example, `LogHost=192.168.0.1,192.168.0.3,192.168.0.4`.

To configure load balancing:

- Modify the LogHost entry in the `logevent.conf` (Linux and Solaris) or `logevent.cfg` (NetWare and Windows) file for each Platform Agent so that it logs to a different Secure Logging Server in the server group.

  For example, one Platform Agent sends data to 192.168.0.1, while another Platform Agent sends data to 192.168.0.3. In this way, you can maximize the number of events that can be logged to each server group's data store.

4.2.4 Configuring a Secure Logging Server with More Than One IP Address

Secure Logging Servers with more than one IP address have problems running Nsure Audit because MDB does not know which IP address to use with eDirectory. You can point Nsure Audit to a specific IP address using an MDB configuration file.

The required filename and path for the MDB configuration file is as follows:

Table 4-3 *MDB Configuration File*

Platform	Directory
NetWare	`sys:\etc\mdb.cfg`
Windows	`\windows\mdb.cfg`
Linux	`/etc/mdb.conf`
Solaris	`/etc/mdb.conf`

To point Nsure Audit to a specific IP address for eDirectory, the MDB configuration file must store the following parameter:

```
driver=mdbds referral=eDirectory_IP_ Address
```

For example,

```
driver=mdbds referral=192.168.123.45.
```

4.3 Configuring the Data Store

Nsure Audit is able to write the data store to the following storage devices:

- File Data Store
- MySQL Data Store
- Oracle Data Store
- Syslog Data Store

Before selecting a storage device for your data store, you need to consider your system's logging traffic. On the high end, the File driver can process over 60,000 events per second on a P4 Xeon class server. Databases, on the other hand, are, much slower. The MySQL driver can handle about 3,000 events per second on a P4 Xeon class server.

Novell Nsure Audit is designed to handle occasional peaks that exceed a given database's limitations; however, if you expect to consistently exceed the database driver's capacity, you must plan your setup accordingly, either by using multiple Secure Logging Servers or by using the file driver.

IMPORTANT: In planning your system setup, you should perform your own throughput test in your environment and not rely solely on the numbers provided in this document.

To configure the Nsure Audit data store, you must first create a Channel object. Each Channel object defines the parameters associated with its corresponding storage device. For example, MySQL Channel objects include the IP address or host name of the MySQL database server, a username and password for connecting with the server, the database and table names, and fields for SQL table create and expiration commands. For more information on creating and configuring Channel objects, see Chapter 7, "Configuring System Channels," on page 69.

After creating the Channel object, you must configure the logging server to log events to that channel. The Log Driver property in the Logging Server object determines which Channel object the server uses to create the data store. For more information on the Log Driver property, see "Logging Server Objects" on page 45.

After the Channel and Logging Server objects are configured, you must restart the logging server to load the Channel object configuration and the channel driver. In most cases, the channel driver automatically creates the necessary file or database table for the data store.

IMPORTANT: Novell Nsure Audit does not secure the data store. Therefore, you must manage data store security at the database, for MySQL and Oracle data stores, or through the file system, in the for file data stores.

The data store structure for each storage device is discussed in the following sections.

4.3.1 File Data Store

Depending on the File Channel object configuration, the File channel driver (lgdfile) can log events in raw format or it can translate the event data into a human-readable log. All file data stores are named "log."

Raw files simply contain the event data; consequently, they are not in a human-readable format. However, because they maintain a consistent field structure across events, they can be imported into spreadsheet programs like Microsoft Excel.

The following is a sample from a raw log file:

```
16777343,1051924636,1051924647,eDirInst\Object,721699,7,0,.OntarioTest
Data.Channels.Logging
Services,,0,0,0,LlNhdHVybiBMb2dnaW5nIFNlcnZlci5Mb2dnaW5nIFNlcnZpY2Vz
16777343,1051924636,1051924647,eDirInst\Object,721690,7,0,.eDirectoryI
nstrumentation.Applications.Logging
Services,,0,0,0,LlNhdHVybiBMb2dnaW5nIFNlcnZlci5Mb2dnaW5nIFNlcnZpY2Vz
16777343,1051926065,1051926065,eDirInst\Object,720897,7,0,.BillBob.SIM
,,0,0,1,LmFkbWluLlNJTQ=
```

Translated log files, on the other hand, can be visually scanned for content; however, it is difficult to generate reports from these files because there is no consistent field structure—they contain only the event descriptions.

The following is a sample from a translated log file:

```
[Sat, 03 May 2003 01:25:10 +0000] eDirInst\Object: A read operation was
performed on object .OntarioTestData.Channels.Logging Services by
.Saturn Logging Server.Logging Services
[Sat, 03 May 2003 01:25:10 +0000] eDirInst\Object: A list Subordinate
Entires operation has been performed on container .eDirectory
```

```
Instrumentation.Applications.Logging Services by .Saturn Logging
Server.Logging Services
[Sat, 03 May 2003 01:39:41 +0000] eDirInst\Object: A new eDirectory
object called .BillBob.SIM (Class:User) was created by .admin.SIM
```

In addition to providing different log formats, the File channel is capable of creating localized logs. If the logging applications have localized Log Schema (LSC) files, the File channel can write translated log files in the language designated in the File Channel object.

Nsure Audit includes a utility, called LETrans, that can translate raw log files into human readable format. See Section G.7, “LETrans,” on page 243.

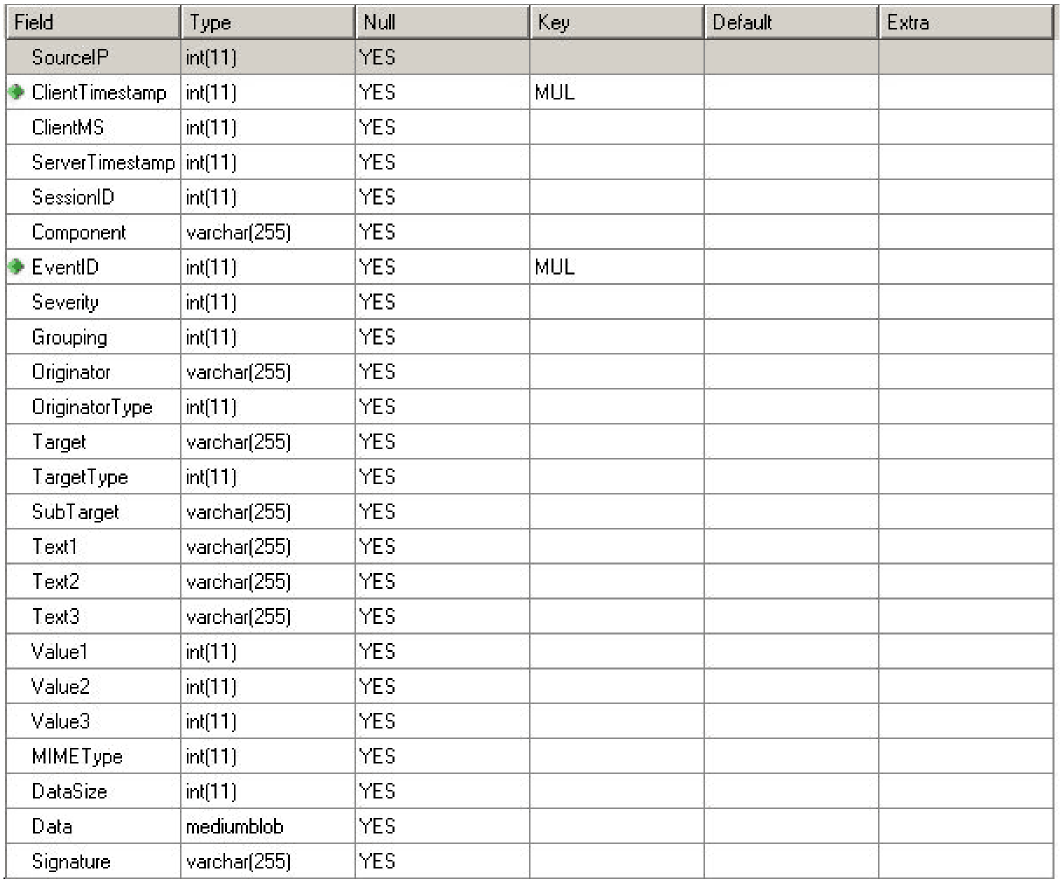
NOTE: LSC files catalog the events that can be logged for a given application. They can also indicate what kind of data is stored in the event fields and provide descriptive information on the event itself. For more information, see Section A.4, “Log Schema Files,” on page 187.

For more information on the File channel, see Section 7.5, “File,” on page 74.

4.3.2 MySQL Data Store

When the logging server loads the MySQL Channel object configuration, the MySQL channel driver, lgdmsql, automatically creates the data store's database and table using the names defined in the MySQL Channel object.

The MySQL channel driver builds the data store using the following table structure:



Field	Type	Null	Key	Default	Extra
SourceIP	int(11)	YES			
◆ ClientTimestamp	int(11)	YES	MUL		
ClientMS	int(11)	YES			
ServerTimestamp	int(11)	YES			
SessionID	int(11)	YES			
Component	varchar(255)	YES			
◆ EventID	int(11)	YES	MUL		
Severity	int(11)	YES			
Grouping	int(11)	YES			
Originator	varchar(255)	YES			
OriginatorType	int(11)	YES			
Target	varchar(255)	YES			
TargetType	int(11)	YES			
SubTarget	varchar(255)	YES			
Text1	varchar(255)	YES			
Text2	varchar(255)	YES			
Text3	varchar(255)	YES			
Value1	int(11)	YES			
Value2	int(11)	YES			
Value3	int(11)	YES			
MIMEType	int(11)	YES			
DataSize	int(11)	YES			
Data	mediumblob	YES			
Signature	varchar(255)	YES			

The default number of rows depends on the operating system. The default maximum size for a MySQL table is 4 GB (or 2 GB if your operating system only supports 2 GB tables). This default size limitation keeps pointer sizes down, making the index smaller and faster.

IMPORTANT: If the SQL server data volume runs out of disk space, any clients logging events will freeze and need to be restarted.

NOTE: If you need larger tables, use the `max_rows` and `avg_row_length` commands in the MySQL Channel object's Create Table Options property

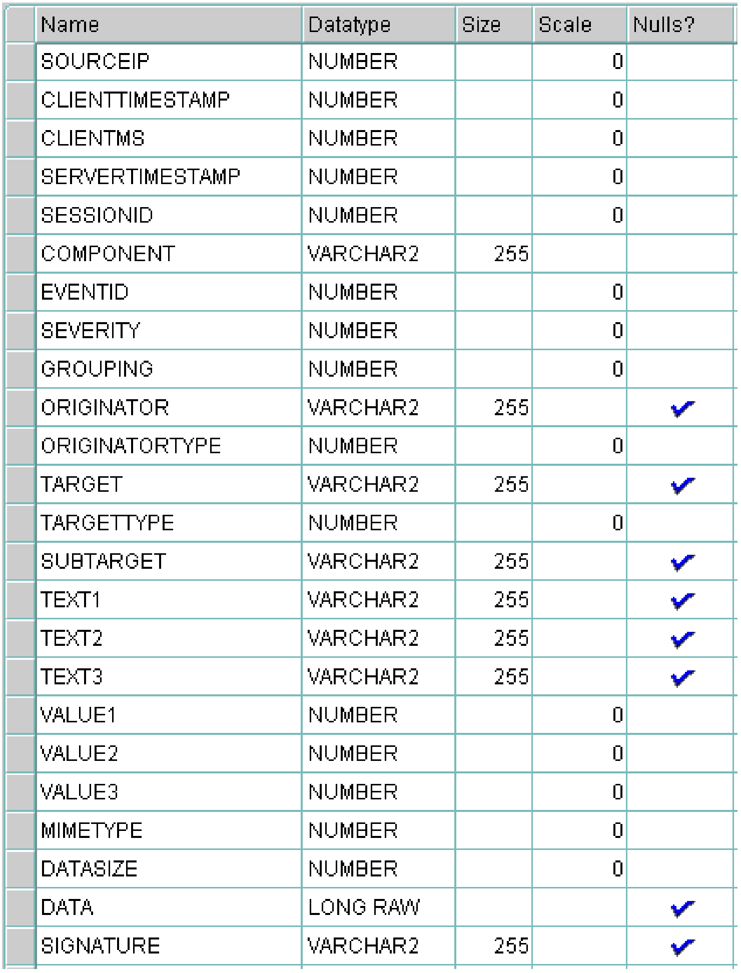
For more detailed information on using MySQL with Nsure Audit, see Appendix C, “Using MySQL with Nsure Audit,” on page 211.

For more information on the MySQL channel, see Section 7.9, “MySQL,” on page 86.

4.3.3 Oracle Data Store

The Oracle channel drive, lgdora, creates the table for the Oracle data store automatically. In most circumstances, you do not need to create the data store table.

If a situation arises requiring you to create this table manually, you can create the data store using the following table structure:



Name	Datatype	Size	Scale	Nulls?
SOURCEIP	NUMBER		0	
CLIENTTIMESTAMP	NUMBER		0	
CLIENTMS	NUMBER		0	
SERVERTIMESTAMP	NUMBER		0	
SESSIONID	NUMBER		0	
COMPONENT	VARCHAR2	255		
EVENTID	NUMBER		0	
SEVERITY	NUMBER		0	
GROUPING	NUMBER		0	
ORIGINATOR	VARCHAR2	255		✓
ORIGINATORTYPE	NUMBER		0	
TARGET	VARCHAR2	255		✓
TARGETTYPE	NUMBER		0	
SUBTARGET	VARCHAR2	255		✓
TEXT1	VARCHAR2	255		✓
TEXT2	VARCHAR2	255		✓
TEXT3	VARCHAR2	255		✓
VALUE1	NUMBER		0	
VALUE2	NUMBER		0	
VALUE3	NUMBER		0	
MIMETYPE	NUMBER		0	
DATASIZE	NUMBER		0	
DATA	LONG RAW			✓
SIGNATURE	VARCHAR2	255		✓

For more information on the Oracle channel, see Section 7.10, “Oracle,” on page 90.

IMPORTANT: Because Oracle no longer supports NetWare, Oracle data stores can be created only on Windows, Solaris, and Linux systems.

4.3.4 Syslog Data Store

The Syslog channel driver, lgdsyslog, allows the logging server to log events to a specific syslog facility on any syslog host.

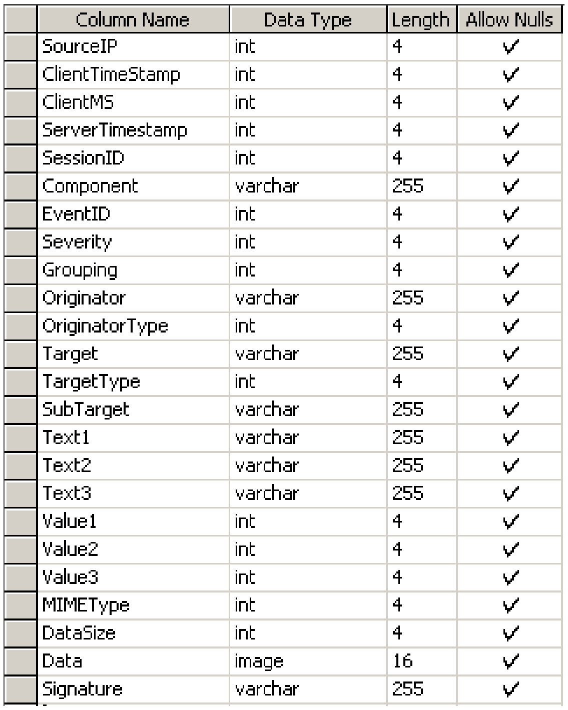
It is also capable of creating localized logs. If the logging applications have localized Log Schema (LSC) files, the Syslog channel can write the log files in the language designated in the Syslog Channel object.

For more information on the Syslog channel, see Section 7.13, “Syslog,” on page 98.

4.3.5 Microsoft SQL Server Data Store

The Microsoft SQL Server channel driver, lgdmssql, creates the table for the SQL Server data store automatically. In most circumstances, you do not need to create the data store table.

If a situation arises requiring you to create this table manually, you can create the data store using the following table structure:



Column Name	Data Type	Length	Allow Nulls
SourceIP	int	4	✓
ClientTimeStamp	int	4	✓
ClientMS	int	4	✓
ServerTimestamp	int	4	✓
SessionID	int	4	✓
Component	varchar	255	✓
EventID	int	4	✓
Severity	int	4	✓
Grouping	int	4	✓
Originator	varchar	255	✓
OriginatorType	int	4	✓
Target	varchar	255	✓
TargetType	int	4	✓
SubTarget	varchar	255	✓
Text1	varchar	255	✓
Text2	varchar	255	✓
Text3	varchar	255	✓
Value1	int	4	✓
Value2	int	4	✓
Value3	int	4	✓
MIMEType	int	4	✓
DataSize	int	4	✓
Data	image	16	✓
Signature	varchar	255	✓

Logging eDirectory, NetWare, and File System Events

5

Novell® Nsure™ Audit provides the instrumentations necessary to log Novell eDirectory™, NetWare®, traditional file system, and NSS events. This section provides the information you need to determine which events you should log to protect your corporate assets and how to log those events.

5.1 Instrumentation Files

The NetWare and eDirectory Instrumentations for Novell Nsure Audit (auditNW and nauditDS, respectively) allow Nsure Audit to log NetWare, eDirectory, and file system events.

To enable NetWare and file system logging, auditNW must be loaded on every NetWare server on which you want to log NetWare and file system events. To avoid receiving duplicate entries for eDirectory events, enable the do not sent replicated events option. To enable this, open the Nsure Audit tab of your NCP Server object and check the "Do not send replicated events" checkbox. To log non-replicated events (such as logins), it must be installed on each individual server for which you want to log non-replicated events.

Additionally, the Platform Agent must be installed on every server on which you want to log NetWare, file system, and eDirectory events. AuditNW and nauditDS automatically load the Platform Agent (logevent) to send events to the Secure Logging Server.

On NetWare, auditNW and nauditDS are automatically loaded each time the server restarts. On Windows, Linux, and Solaris systems, you must manually load nauditDS or add nauditDS to the server startup scripts to begin logging eDirectory events. For information on starting the NetWare and eDirectory Instrumentations, see Section G.4, "NetWare and eDirectory Instrumentation Startup Commands," on page 238

5.1.1 Supported Platforms

The eDirectory Instrumentation can log events from the following versions of the Directory:

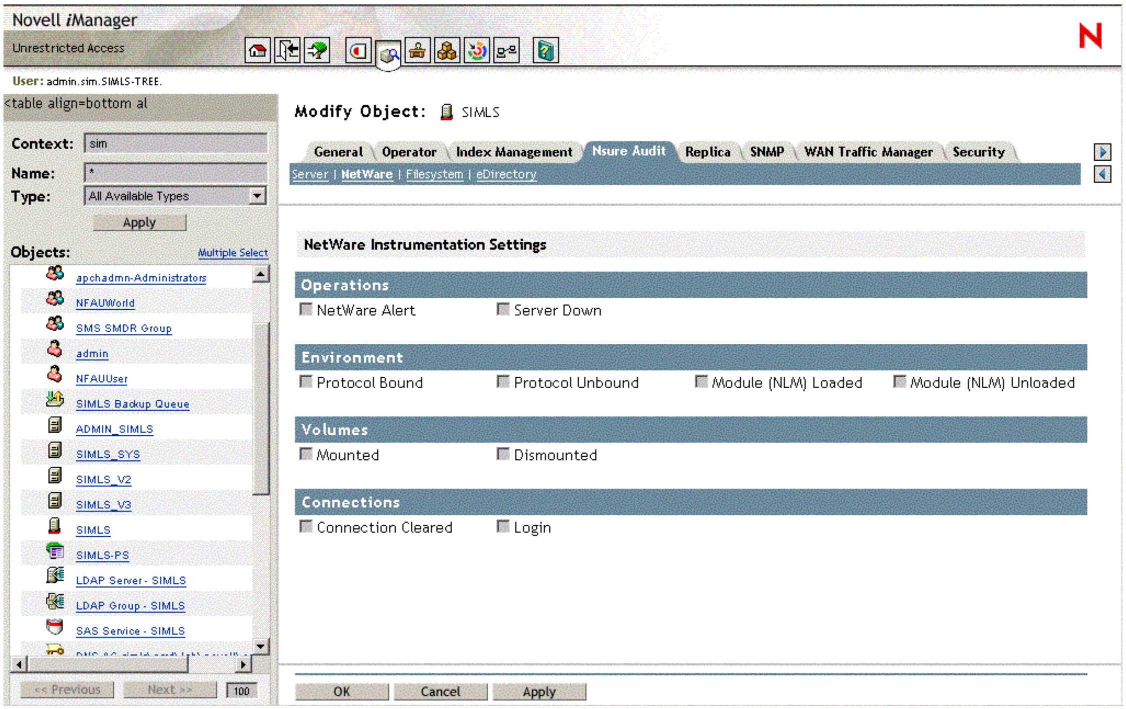
- NDS® 6.x
- NDS 7.x
- NDS 8.x
- eDirectory 8.6 (NetWare, Windows, Linux, and Solaris)
- eDirectory 8.7 (NetWare, Windows, Linux, and Solaris)

The NetWare Instrumentation can log NetWare and file system events from the following platforms:

- NetWare 4.2 SP9
- NetWare 5.1 SP6
- NetWare 6.0 SP3
- NetWare 6.5

5.2 Configuring eDirectory, File System, and NetWare Events

During installation, Nsure Audit extends the definition of the NCP Server object to include log settings for eDirectory, NetWare, and file system events. These settings are found under the Nsure Audit option in the NCP Server object.



NetWare Instrumentation Settings

Operations

□ NetWare Alert □ Server Down

Environment

□ Protocol Bound □ Protocol Unbound □ Module (NLM) Loaded □ Module (NLM) Unloaded

Volumes

□ Mounted □ Dismounted

Connections

□ Connection Cleared □ Login

The Nsure Audit screen has four different menus: Server, NetWare, Filesystem, and eDirectory.

The Server menu identifies the Logging Server object associated with the current NCP Server object. This menu is for informational purposes only and cannot be modified.

The NetWare, Filesystem, and eDirectory menus list the events that fall in their respective categories.

To select which NetWare, File system, or eDirectory events you want to log:

1. Click *Nsure Audit* in the NCP Server object.
2. Select an event menu (NetWare, Filesystem, or eDirectory).
3. Mark the check box next to the events you want to log.
4. Click *Apply*.

IMPORTANT: You must click *Apply* in each screen to save your changes.

5 When finished, click *OK*.

When you click *Apply*, the logging server automatically begins logging the marked events.

NOTE: You do not need to restart the logging server to effect changes to Nsure Audit attributes in the NCP Server object.

5.3 NetWare Events

NetWare events are server-specific settings; that is, they must be enabled on each NCP Server object in the tree. The NetWare Instrumentation can log NetWare and file system events from NetWare 4.2 systems and higher.

For a complete list of the NetWare events that can be logged to Novell Nsure Audit, see Section B.3, "NetWare Events," on page 207.

NOTE: You do not need to restart the logging server to activate your changes in the NetWare menu.

5.4 File System Events

File System events are server-specific settings; that is, they must be enabled on each NCP Server object in the tree. The NetWare Instrumentation logs all traditional and NSS file system activity for the selected events.

NOTE: If you want to filter events on a volume or directory level, you can create Notification filters that select events based on the volume or directory listed in the Text2 field.

For a complete list of the File System events that can be logged to Novell Nsure Audit, see Section B.2, "File System Events," on page 206.

NOTE: You do not need to restart the logging server to activate your changes in the Filesystem menu.

5.5 eDirectory Events

eDirectory events are partition-specific; that is, they only need to be enabled on one NCP Server object per partition. The eDirectory Instrumentation can log events from NDS version 6 or 7 and eDirectory 8.5 or higher.

For a complete list of the eDirectory events that can be logged to Novell Nsure Audit, see Section B.1, "eDirectory Events," on page 191.

NOTE: You do not need to restart the logging server to activate your changes in the eDirectory menu.

eDirectory events describing attribute changes store the new attribute values in the event's data field.

In the case of a few critical events, eDirectory cannot complete the transaction until the corresponding event is sent to the Secure Logging Server. This ensures that the transaction is logged to the data store. (These events are noted in the event table.)

eDirectory events such as login and logout are ubiquitous and can quickly fill your data store. Therefore, you should monitor your system's event traffic and configure your data store's expiration or roll policies accordingly. For information on the MySQL channel's expiration properties, see "MySQL Channel Object" on page 87. For information on configuring the File channel to purge or roll its log files, see "File Channel Object" on page 76.

Managing Applications that Log to Nsure Audit

6

Novell® Nsure™ Audit is a comprehensive auditing system that is capable of logging events from multiple, multi-vendor applications. This section provides the information you need to manage your system's logging applications.

- Section 6.1, "Overview," on page 65
- Section 6.2, "Creating Application Objects," on page 65
- Section 6.3, "Application Objects," on page 67

6.1 Overview

Applications that log to or request information from Nsure Audit are represented in Novell eDirectory™ by an Application object. Application objects store the information the logging server needs to authenticate logging applications.

NOTE: For more information on the authentication process, see Section 10.1, "Authenticating Logging Applications," on page 159.

In addition to facilitating system authentication and monitoring, Application objects store the application's log schema. The log schema catalogs the events that can be logged for a given application. For more information, see Section A.4, "Log Schema Files," on page 187.

6.2 Creating Application Objects

Typically, Application objects are automatically created in the Application container under Logging Services when their associated logging application is installed.

For example, during installation, Novell Nsure Audit automatically creates Application objects for itself (the Naudit Instrumentation), the eDirectory Instrumentation, and the NetWare® Instrumentation. Novell Nsure Audit creates these objects in the Application container under Logging Services.

NOTE: The Naudit Instrumentation allows Nsure Audit to audit its own events, such as creating Channel or Notification objects. The eDirectory Instrumentation manages logging of eDirectory events, and the NetWare Instrumentation (NetWare only) provides logging for NetWare and file system events. For more information on the eDirectory and NetWare instrumentations, see Chapter 5, "Logging eDirectory, NetWare, and File System Events," on page 61.

If necessary, you can manually create Application objects using iManager. For information on using iManager to create objects, see "Nsure Audit iManager Plug-in" on page 35.

To manually create the Application object using iManager, you must have the following information:

Application Object Attribute	Description
Application Identifier	The name the logging application uses to identify itself to the logging server. The Application Identifier should be available in the product's documentation and it is included in the product's Log Schema file. For more information, see Section 6.3, "Application Objects," on page 67.
Application ID	The four-digit hex value assigned to the current application. All Application IDs are assigned through Novell Developer Support and are maintained in the Novell Nsure Audit central registry. The Application ID should be available in the product's documentation and it is included in the product's Log Schema file. For more information, see Section 6.3, "Application Objects," on page 67.
Log Schema File	Log Schema (LSC) files catalog the events that can be logged for a given application. They also provide event descriptions and field titles, although this is optional. Novell Nsure Audit stores each application's LSC files as attributes in its respective Application object. English LSC files are stored under the NAuditAppSchemaEn attribute, French LSC files are stored under the NAuditAppSchemaFr attribute, and so forth. **NOTE:** If you modify or localize an application's LSC file, you must manually add the LSC file to the Application object's log schema attribute by running the AuditExt utility at the server console. For information on manually adding LSC files to Application objects, see "Using AuditExt to Add LSC Files to Application Objects" on page 241.

Application Containers

You must create Application objects in Application containers. The Application container under Logging Services is automatically created during installation; however, additional Application containers can be created anywhere in the tree.

Creating Application objects in the central Application container under Logging Services is ideal for organizations that need a simple, easy-to-manage logging system. It also suits organizations that are implementing Nsure Audit as an auditing solution and, for security reasons, want to centrally manage their system.

If you want to distribute logging system administration, however, Application objects can be created anywhere in the tree. For example, if administration is divided by logging server, you can create an Application container under each Logging Server object. On the other hand, if administration is divided by application (for example, one person manages logging for iChain®, another DirXML® logging, etc.), the Application container can be created in any context assigned to its administrator.

If you create an Application container elsewhere in the tree, you must add that container to the logging server's list of supported containers. At startup, the logging server scans its list of supported Application containers and loads the included Application object configurations in memory so it can authenticate applications. If an Application object is not in one of the logging server's supported Application containers, it cannot be used to authenticate logging applications. For more information on the logging server's Application Container property, see "Logging Server Objects" on page 45.

IMPORTANT: The logging server loads the Application object configurations at startup only. Therefore, if you create a new Application container or Application object, you must first ensure that the Application container is included in the logging server's Application Container list and then restart the logging server. For information on restarting the logging server, see Section G.3, "Secure Logging Server Startup Commands," on page 236.

6.3 Application Objects

Application objects store the information required by the logging server to authenticate logging applications. They also identify which users have rights to monitor the application's events and they store the application's log schema.

The following table provides a description of each Application object attribute.

IMPORTANT: You must restart the logging server to effect any changes in Application object configuration. For more information, see Section G.3, "Secure Logging Server Startup Commands," on page 236.

Attribute	Description
Application Identifier	The name the logging application uses to identify itself to the logging server. The Application Identifier is also stored in the application's certificate. For information on how the Application Identifier is used in the authentication process, see Section 10.1, "Authenticating Logging Applications," on page 159. The Application Identifier is part of the Component string for every event logged from the current application. For more information, see Section A.1, "Event Structure," on page 175. This field is automatically populated when the Application object is created during install. If you manually create the Application object, you can find the Application Identifier in the application's Log Schema file. For more information, see Section A.4, "Log Schema Files," on page 187.

Attribute	Description
Application ID	The four-digit hex value assigned to the current application. All Application IDs are assigned through Novell Developer Support and are maintained in the Novell Nsure Audit central registry. The Application ID is also part of the Event ID for every event logged from the current application. For more information, see Section A.1, “Event Structure,” on page 175. This field is automatically populated when the Application object is created during install. If you manually create the Application object, you can find the Application ID in the application's Log Schema file. For more information, see Section A.4, “Log Schema Files,” on page 187 **NOTE:** The logging server uses the Application ID to manage Access Control. The users designated in the Access Control field can monitor all events containing this Application ID.
Access Control	The users who have rights to monitor the current application's events. This is for future iterations of the product.
Status	This option allows you to enable or disable the Application object. By default, all Application objects are enabled. This means that the logging server loads the Application object's configuration in memory at startup. **IMPORTANT:** The Application object must be located in a supported Application container for the logging server to use it. For more information on the logging server's Application Container property, see “Logging Server Objects” on page 45. If you mark the Disabled option, you must restart the Secure Logging Server for the setting to become effective. Thereafter, the logging server cannot load the object's configuration until you mark Enabled. For information on unloading the logging server, see Section G.3, “Secure Logging Server Startup Commands,” on page 236.

Configuring System Channels 7

Novell® Audit uses channels to provide event notification and log events. This section provides the information you need to configure your system channels.

7.1 Overview

Channel drivers enable Nsure Audit to log system events and provide event notification. Channel drivers, in turn, are configured and managed through Channel objects in Novell eDirectory™.

Channel objects store the information the logging server needs to use a certain channel. For example, MySQL Channel objects include the IP address or host name of the MySQL database server, a username and password to connect to the server, the database and table names, and other relevant information. SMTP Channel objects, on the other hand, include the IP address or host name of the SMTP server, a username and password, and message information (the message recipients, sender, subject, and body).

7.2 Creating Channel Objects

Nsure Audit is designed so you can create multiple Channel objects for any given channel driver. This means you can create different channel configurations for different functions or events. For instance, you can configure the logging server to use one MySQL Channel object to add events to the central data store and configure a Notification Filter to use another MySQL Channel object to create a filtered log.

To create and configure a channel object:

1. Click the *Roles and Tasks* button on the iManager toolbar.
2. In the Roles and Tasks view, expand the *Auditing and Logging* Role and select the *Logging Server Options* task.

3. Select the Secure Logging Server object and click *OK*.
   - Click the *Object History* button to see a list of Logging Server objects that have been selected during this iManager session.

     or
   - Click the *Object Selector* button to locate the object in the directory tree. To move up or down in the tree, click the navigation arrows. You can also search the tree by specifying the object name and context in the Search tab.
4. In the Logging Server Options page, click *Channels*.
5. Select the Channels container and click *Channel Actions* > *New*, then click *OK*.
6. In the New Channel page, select the desired channel.
7. Specify a name for channel object and click *OK*.
8. Configure the channel attributes.

   For more information about each channel, see Section 7.3, “Supported Channels,” on page 71.
9. When finished, click *OK*.

Channel Containers

You must create Channel objects in Channel containers. The Channel container under Logging Services is automatically created during installation; however, additional Channel containers can be created anywhere in the tree.

Creating Channel objects in the central Channel container under Logging Services is ideal for organizations that need a simple, easy-to-manage logging system. It also suits organizations that are implementing Nsure Audit as an auditing solution and, for security reasons, want to centrally manage their system.

If you want to distribute logging system administration, however, Channel objects can be created anywhere in the tree. For example, if administration is divided by logging server, you can create a Channel container under each Logging Server object. On the other hand, if administration is divided by application (for example, one person manages logging for iChain®, another Identity Manager logging, etc.), the Channel container can be created in any context assigned to its administrator.

If you create a Channel container elsewhere in the tree, you must add that container to the logging server’s list of supported containers. At startup, the logging server scans its list of supported Channel containers and loads the included Channel object configurations and their associated drivers in memory so it can provide event notification and log events. If a Channel object is not in one of the logging server’s supported Channel containers, it cannot be used to provide event notification or log events. For more information on the logging server’s Channel Container property, see “Logging Server Objects” on page 45. For more information on creating channel containers, see Section 3.4.1, “Creating Objects in iManager,” on page 38.

IMPORTANT: The logging server loads the Channel object configurations only at startup. Therefore, if you create a new Channel container or Channel object, you must first ensure that the Channel container is included in the logging server’s Channel Container list and then restart the logging server. For information on restarting the logging server, see Section G.3, “Secure Logging Server Startup Commands,” on page 236.

7.3 Supported Channels

IMPORTANT: The NetWare® 6.5 product license authorizes you to use the Nsure Audit SMTP, File, and MySQL channels. You must acquire a license for every other channel. If the additional channels are configured without a license, the Secure Logging Server does not load.

By default, Nsure Audit supports the following channels:

CVR	MySQL
File	Oracle
Java	SMTP
JDBC	SNMP
Microsoft SQL Server	Syslog

Additional channels can be easily incorporated in this model. For more information, see the Nsure Audit SDK (http://developer.novell.com).

The directory in which the channel drivers (lgd*) are located is defined on the Logging Server object; however, the default channel driver directories are as follows:

***Table 7-1** Channel Driver Directories*

Operating System	Directory
NetWare	`sys:\system\`
Windows	`\program files\novell\nsure audit\`
Linux	`/opt/novell/naudit/`
Solaris	`/opt/NOVLnaudit/`

The following sections provide detailed information on each channel.

7.4 CVR

The Critical Value Reset (CVR) channel allows you to flag an attribute in eDirectory with a reset policy. If the value of that specific attribute is changed, the CVR channel resets the value as per the policy defined in the CVR Channel object.

The CVR channel can be used to maintain critical system settings or enforce organizational policies. For example, if your organization has a policy prohibiting security equivalence, you can create a CVR Channel object that automatically resets the Security Equals attribute to a null value.

It can also be used to provide an added measure of system security. In the event of a security breach, the CVR channel can be configured to maintain your critical system settings.

The CVR channel must be used in conjunction with a notification filter. To optimize event processing, the notification filter should be configured to filter only those events that the CVR channel can act on. For information on configuring Notification Filters, see Chapter 8, “Configuring Filters and Event Notifications,” on page 101.

7.4.1 CVR Channel Driver

The CVR driver is `lgdcvr`.

When the CVR channel driver receives an event, it looks at the event’s Text2 field to determine if the logged attribute matches the attribute defined in the CVR object. If the CVR driver does not find a matching attribute in the event’s Text2 field, it then looks in the Text1 field.

If the contents of the *Text2* or *Text1* field match the attribute defined in the CVR object, the driver looks in the remaining Text field to find the object to which the attribute belongs. It then locates the object in eDirectory and applies the Reset Value defined in the CVR object.

NOTE: All eDirectory events store the event’s attribute in the *Sub Target* field and the object in the *Target* field.

The reset process is very fast. Typically, an attribute is reset the instant it is saved. In fact, in iManager, it simply appears that the change cannot be saved.

7.4.2 CVR Channel Object

The CVR Channel object stores the policy and attribute information the CVR driver needs to reset a given value.

The following table provides a description of each Channel object attribute.

IMPORTANT: You must restart the logging server to effect any changes in Channel object configuration. For more information, see Section G.3, “Secure Logging Server Startup Commands,” on page 236.

Table 7-2 *CVR Channel Object Attributes*

Attribute	Description
Configuration	Configuration information for the CVR Channel object.
User	The User object with rights to the CVR Channel object. **IMPORTANT:** The User object must have directory rights to the attribute that the CVR Channel object is configured to reset.
Password	The user account password.

Attribute	Description
Type	The type of data the CVR channel can expect as a Reset Value for the attribute designated in the Attribute field. The data type must match the Attribute Syntax defined in the directory schema. Currently, the only supported types are Distinguished Name and String. The Distinguished Name type supports only Distinguished Name syntax (.cn.ou.ou.o). For example, .admin.sim.mycorp. The String type, on the other hand, supports multiple Attribute Syntax options. They include: • String • Class Name • Case Exact String • Case Ignore String • Numeric String • Postal Address • Printable String • Telephone Number **NOTE:** The Attribute Syntax for a given attribute can be viewed in the eDirectory schema using a directory editing tool such as NDS® Snoop, ConsoleOne®, or iManager.
Attribute	The name of the attribute the CVR driver resets. You must specify the attribute name exactly as it appears in the eDirectory schema. **NOTE:** You can view the eDirectory schema using a directory editing tool such as NDS Snoop, ConsoleOne, or iManager. The CVR driver scans events' *Target* and *Sub Target* fields for matching attributes. When it finds a match, the CVR driver applies the reset policy. For more information on this process, see "CVR Channel Driver" on page 72.
Reset Value	The value the CVR Channel driver maintains for a given attribute. **IMPORTANT:** The CVR Channel driver does not validate the reset value syntax. Therefore, you must ensure the reset value follows the required attribute syntax. For example, if the Attribute Syntax is Telephone Number, the reset value must be a telephone number.
Operators +-	Click the plus sign (+) to add a new line. Click the minus sign (-) to remove a line. Each line defines a separate reset policy. The policies are not accumulative; the CVR driver applies each policy independently. There is no programmed limit to the number of policies that can be added to a CVR object.

Attribute	Description
Status	Allows you to enable or disable the Channel object. By default, all Channel objects are enabled. This means that the logging server loads the Channel object's configuration in memory at startup. **IMPORTANT:** The Channel object must be located in a supported Channel container for the logging server to use it. For more information on the logging server's Channel Container property, see "Logging Server Objects" on page 45. If you select the *Disabled* option, you must restart the Secure Logging Server for the setting to become effective. Thereafter, the logging server cannot load the object's configuration until you select *Enabled*. For information on unloading the logging server, see Section G.3, "Secure Logging Server Startup Commands," on page 236.

7.5 File

The File channel allows the logging server to log events directly to file in raw format or to translate those events to a human-readable log file.

The default option is to translate the log files. Raw log files can be translated at a later time using the letrans utility. For more information, see Section G.7, "LETrans," on page 243.

Raw files simply contain the event data; consequently, they are not in a human-readable format. However, because these comma-delimited files maintain a consistent field structure across events, you can import these files into spreadsheet programs like Microsoft Excel*.

The following is a sample from a raw log file:

```
2302777206,1118368566,1118955519,eDirInst\Agent,720917,7,346,,0,pki.dl
m,0,,,,,0,0,0,0,
2302777206,1118368567,1118955519,eDirInst\Agent,721735,7,346,,1,\BOOP-
TREE\novell\BOOP-NDS,0,,198.53.162.118,,,33,262176,0,0,
2302777206,1118956181,1118956181,eDirInst\Meta,721748,7,28,.Admin.nove
ll,1,.Rufus.novell,1,,Entry ID: .Rufus.novell, Attribute ID: [All
Attributes Rights], Privileges: Attribute Read,,,1,0,0,0,
2302777206,1118956186,1118956186,eDirInst\Attribute,720902,7,31,.BOOP-
NDS.novell,1,.[Root].,1,Partition
Status,,,,9,0,0,0,AAAAAAIAAACa6rFCAAAAAAAAAAAAAAAADgAAAFs
AUgBvAG8AdABdAAAAAAAQAAAAAAAAAIoiAAD/////AAAAAA=
2302777206,1118956186,1118956186,eDirInst\Attribute,720902,7,32,.BOOP-
NDS.novell,1,.[Root].,
1, Purge Vector,Seconds: 1118956186, Replica Number: 1, Event:
1,,,19,0,0,0,
2302777206,1118955953,1118955953,eDirInst\Attribute,720902,7,22,.Admin
.novell,1,.Birds.novell,1,Owner,.Admin.novell,,,1,0,0,0,
2302777206,1118955953,1118955953,eDirInst\Meta,721748,7,23,.Admin.nove
ll,1,.Birds.novell,1,,Entry ID: .[Root]., Attribute ID: Member,
Privileges: Attribute Read,,,1,0,0,0,
2302777206,1118956181,1118956181,eDirInst\Attribute,720902,7,28,.Admin
.novell,1,
.Rufus.novell,1,Password Allow Change,True,,,7,0,0,0,
```

Translated log files can be visually scanned for content; however, it is difficult to generate reports from these files because there is no consistent field structure—they contain only the event descriptions defined in the application's Log Schema (LSC) file.

The following is a sample from a translated log file:

```
[Thu, 09 Jun 2005 14:28:03 -0700] [eDirInst\Object]: A list
Subordinate Entries operation has been performed on container
.eDirectory Instrumentation.Applications.Logging Services by .Boop-nds
Logging Server.Logging Services [Thu, 09 Jun 2005 14:28:40 -0700]
[eDirInst\Partition]: Synchronization has ended on partition .[Root]..
All Processed: Yes [Thu, 09 Jun 2005 14:28:51 -0700]
[eDirInst\Object]: User .Admin.novell (using null password: No) logged
in (NDS Login: Yes) to server \BOOP-TREE\novell\BOOP-NDS. [Thu, 09 Jun
2005 14:29:04 -0700] [eDirInst\Object]: A read operation was performed
on object .BOOP-TREE CA.Security by .Admin.novell [Thu, 09 Jun 2005
14:34:04 -0700] [eDirInst\Object]: A read operation was performed on
object .File.Channels.Logging Services by .Boop-nds Logging
Server.Logging Services Thu, 09 Jun 2005 14:48:41 -0700]
[eDirInst\Agent]: The connection state has been changed by .BOOP-
NDS.novell [Thu, 09 Jun 2005 14:48:41 -0700] [eDirInst\Replica]: A
purge operation has started on partition .[Root]. [Thu, 09 Jun 2005
14:48:41 -0700] [eDirInst\Replica]: A purge operation has ended on
partition .[Root].
```

In addition to providing different log formats, the File channel is capable of creating localized logs. If the logging applications have localized log schema files and if those files are added to their respective Application objects, the File channel can write translated log files in the language designated in the File Channel object.

The logging server can use the File channel to write the central data store or create filtered log files.

7.5.1 File Channel Driver

At startup, the File channel driver, `lgdfile`, loads each application's log schema. If a logging application has multiple language versions of its log schema, the File channel loads the schema for the language designated in the File Channel object.

NOTE: The Log Schema catalogs the events that can be logged for a given application. It can also provide event descriptions and labels for the event fields. For more information, see "Log Schema Files" on page 187.

If the File and Syslog Channel objects reference the same language, the drivers independently load the log schema in their own memory. The only time the log schema is shared is between multiple instances of the same driver. For example, if you have two File channels configured to write translated log files in English, the English log schema for each application is loaded only once.

When the File channel driver creates a raw log file, it writes the event data "as is" to the data store. If the data is in raw format and the DataSize = 0, then each line in the file is written as a comma-separated list of 19 fields in the following order:

```
SourceIP,ClientTimestamp,ServerTimestamp,Component,ID,Severity,
GroupID,Originator, OriginatorType,Target,TargetType,SubTarget,Text1,
Text2,Text3,Value1,Value2,Value3,0(Just a trailing zero)
```

If DataSize is not 0, then each line in the raw file is written as a comma-separated list of 20 fields. MIMEHint replaces the trailing 0 and the last field is the Data string:

```
SourceIP,ClientTimestamp,ServerTimestamp,Component,ID,Severity,
GroupID,Originator, OriginatorType,Target,TargetType,SubTarget,Text1,
Text2,Text3,Value1,Value2,Value3,MIMEHint,DataString
```

When it creates a translated log file, the File driver uses the Event ID to look up each event in the corresponding application's log schema and then it writes the event description to the data store. If the log schema isn't available, or if there isn't a descriptive entry for the current event, the File channel defaults to the following format:

```
$DC $TC,$SO,$NI,$NL,$NG,$SB,$NH,$SU,$NV,$SY,$N1,$N2,$N3,$SS,$ST,$SF\n
```

(Client date and time Stamp, Component, EventID, Log Level, Group ID, Originator, OriginatorType, Target, TargetType, Subtarget, Value1, Value2, Value3, Text1, Text2, Text3.) See Section A.3, "Managing Event Data," on page 181 for an explanation of each field and format variable.

Because it uses the log schema to write translated logs, the File driver is also capable of creating localized logs. If a logging application has localized log schema files and if those files are added to the Application object, the File driver uses the log schema for the language designated in the File Channel object to write the event descriptions. For more information on the File channel's language attribute, see "File Channel Object" on page 76. For information on localized log schema files, see "Localized Log Schema Files" on page 189.

7.5.2 File Channel Object

The File Channel object stores the information the File driver needs to write events to the file system.

The following table provides a description of each Channel object attribute.

IMPORTANT: You must restart the logging server to effect any changes in Channel object configuration. For more information, see Section G.3, "Secure Logging Server Startup Commands," on page 236

Table 7-3 *File Channel Object Attributes*

Attribute	Description
Configuration	File Channel object configuration information.

Attribute	Description
Log File Location	The path to the log file. The default Log File directories are as follows: • `sys:\etc\logdir\` (NetWare) • `\program files\novell\nsure audit\logs\` (Windows) • `/var/opt/novell/naudit/logs/` (Linux) • `/opt/NOVLnaudit/logs/` (Solaris) **IMPORTANT:** By default, file data stores are named `auditlog`. Therefore, if you have multiple File Channel objects, you must either give each log file a different filename or point them to different paths.
Log File Name	The name of the file to which the logging server writes events. The default filename is `auditlog`.
Purge log files after _____ seconds	The life span of the log files. The logging server deletes all log files older than the designated time period.
Flush log files after _____ seconds	The interval at which the file channel driver flushes the events in memory to the log file on disk. **NOTE:** On NetWare, the file channel driver writes events to memory and intermittently flushes the events to disk. To manually flush the file channel buffers, enter `naudit file flush` at the server console.
Roll when log file reaches _____ bytes	The log file's maximum file size. When a log file reaches the designated file size, lgdfile renames the file and creates a new log file. The archive filename is a combination of the current date and a hexadecimal sequence number (l/yy/mm/dd.###). For example, the first log file archived on July 10, 2003 would be named l030710.001. Subsequent log files archived on the same day would be named l030710.002, l030710.003, etc.
Log format	The File channel driver can log events in either translated or raw format. Select either Translated or Raw to set the logging mode for the current Channel object.

Attribute	Description
Translated	This is the default option. In Translated mode, the File channel driver uses the Event ID to look up each event in the application’s log schema and it writes the event description to the data store. If the log schema isn’t available, or if there isn’t a descriptive entry for a particular event, the File channel defaults to the following format: `$DC $TC,$SO,$NI,$NL,$NG,$N1,$N2,$SS,$ST\n` (Client Date and Time Stamp, Component, EventID, Log Level, Group ID, Value1, Value2, Text1, Text2) **NOTE:** Log Schema files (`*.lsc`) catalog the events that can be logged for a given application. They can also provide event descriptions and labels for the event fields. For more information, see Section A.4, “Log Schema Files,” on page 187. Although a translated log file can be visually scanned for content, no reports can be generated from this file because there is no consistent field structure; it contains only the event descriptions.
Raw	In Raw mode, the File channel driver writes the event data in comma-separated format (`csv`) to the data store. The raw log file is not in a human-readable format; however, it can be imported into spreadsheet programs like Microsoft Excel.
Translated Language	The language in which events are written to file. **IMPORTANT:** This option is valid only for Translated log files. If logging applications have localized Log Schema files and if those files are added to their respective Application object, the File channel can write Translated log files in the selected language. If there isn’t a log schema for the selected language, the channel defaults to English. You can create parallel logs in multiple languages by defining multiple File Channel objects with different languages and having a single notification filter pass events to all those channels.
Status	Allows you to enable or disable the Channel object. By default, all Channel objects are enabled. This means that the logging server loads the Channel object’s configuration in memory at startup. The Channel object must be located in a supported Channel container for the logging server to use it. For more information on the logging server’s Channel Container property, see “Logging Server Objects” on page 45. If you select the *Disabled* option, you must restart the Secure Logging Server for the setting to become effective. Thereafter, the logging server cannot load the object’s configuration until you select *Enabled*. For information on unloading the logging server, see Section G.3, “Secure Logging Server Startup Commands,” on page 236.

7.6 Java

The Java channel allows the logging server to output filtered events to a Java application. Typically, the Java Class is a custom application that provides a response to specific types of events. For example, if a user login is disabled, the Java channel driver can launch a Java Class that automatically resets the user account.

WARNING: The Java channel does not work on NetWare 5.x. The Java channel requires JVM* 1.4.2 which is not compatible with NetWare 5.x. Attempting to run the Java channel on NetWare 5.x abends the server.

7.6.1 Java Channel Driver

Nsure Audit installs its Java drivers to the following Nsure Audit classpath directories:

Table 7-4 *Nsure Audit Java Classpath*

Platform	Java Classpath
NetWare	`sys:\system\naudit\`
Windows	`\program files\novell\nsure audit\java\logdriver\`
Linux	`/opt/novell/naudit/java/logdriver/`
Solaris	`/opt/NOVLnaudit/java/logdriver/`

At startup, the Java driver, `lgdjava`, looks in the Nsure Audit Java classpath for the Java Class designated in the Java Channel object configuration. It then attempts to launch the Java Class. If it is successful, that instance of the Class remains active until the Java Channel object is disabled or the Secure Logging Server is shut down.

If it cannot launch the Java Class, the Java driver refuses to load. This safeguard ensures that no events are lost because of misconfiguration.

NOTE: The Java driver does not buffer events that are undeliverable because of misconfiguration or a server failure.

Configuration Requirements

To configure the Java channel, you must perform the following tasks:

- Copy the `.jar` files required for additional Java channels you are using with Nsure Audit to the Nsure Audit Java classpath or a subdirectory thereof.
- If you are using the Java Channel on a Windows machine, you must add the `jvm.dll` directory path to the Path system variable. For example, `c:\j2sdk1.4.2_09\jre\bin\server\`. You must reboot the machine for the changes to take effect
- On Linux/Solaris, the LD_LIBRARY_PATH variable needs to point to the paths for `libverify.so`, and `libjvm.so`. You must reboot the machine for the changes to take effect.

For information on how to hook your Java Class into the Java channel driver, refer to the Java channel API in the Nsure Audit SDK (http://developer.novell.com/ndk/naudit.htm).

7.6.2 Java Channel Object

The Java Channel object stores the information the Java driver needs to launch a Java Class.

The following table provides a description of each Channel object attribute.

IMPORTANT: You must restart the logging server to effect any changes in Channel object configuration. For more information, see Section G.3, "Secure Logging Server Startup Commands," on page 236

Table 7-5 Java Channel Object Attributes

Attribute	Description
Configuration	Contains configuration information for the Java Channel object.
Java Driver Class	The name of the Java Class the Java driver launches.
Max Data Size	The maximum size (in bytes) of information that can be written at one time to the Java application.
Status	Allows you to enable or disable the Channel object. By default, all Channel objects are enabled. This means that the logging server loads the Channel object's configuration in memory at startup. The Channel object must be located in a supported Channel container for the logging server to use it. For more information on the logging server's Channel Container property, see "Logging Server Objects" on page 45. If you select the *Disabled* option, you must restart the Secure Logging Server for the setting to become effective. Thereafter, the logging server cannot load the object's configuration until you select *Enabled*. For information on unloading the logging server, see Section G.3, "Secure Logging Server Startup Commands," on page 236.

7.7 JDBC

The JDBC channel allows the logging server to output filtered events to any JDBC-enabled data store.

WARNING: The JDBC channel does not work on NetWare 5.*x*. The JDBC channel requires JVM 1.4.2, which is not compatible with NetWare 5.*x*. Attempting to run the JDBC channel on NetWare 5.*x* abends the server.

7.7.1 JDBC Channel Driver

Nsure Audit installs its Java drivers to the following Nsure Audit classpath directories:

Table 7-6 *Nsure Audit Java Classpath*

Platform	Java Classpath
NetWare	`sys:\system\naudit\`
Windows	`\program files\novell\nsure audit\java\logdriver\`
Linux	`/opt/novell/naudit/java/logdriver/`
Solaris	`/opt/NOVLnaudit/java/logdriver/`

At startup, the JDBC driver, `lgdjava`, looks in the Nsure Audit Java classpath for the JDBC Class designated in the JDBC Channel object configuration. It then attempts to launch the JDBC Class. If it is successful, that instance of the Class remains active until the JDBC Channel object is disabled or the Secure Logging Server is shut down.

If it cannot launch the JDBC Class, the JDBC driver refuses to load. This safeguard ensures that no events are lost because of misconfiguration.

NOTE: The JDBC driver does not buffer events that are undeliverable because of misconfiguration or a server failure.

Configuration Requirements

The configuration requirements to use the JDBC channel with a JDBC-enabled data store are as follows:

- For performance reasons, we recommend using only the channels discussed in "Data Store" on page 27 as the primary log channel, and use JDBC data stores for notifications.
- Install and configure any JDBC-enabled data store according to the instructions provided by the vendor.
- In the JDBC-enabled data store, create a Nsure Audit database and a database user.
- The server hosting your JDBC data store must have JVM* 1.4.1 or later.
- Obtain the JDBC drivers for your data store.

  The JDBC drivers are available at the following sites:

Table 7-7 *JDBC Driver Sites*

Data Store	Driver	Site
MySQL	MySQL Connector/J	http://dev.mysql.com/downloads/
Oracle	Oracle Instant Client	http://www.oracle.com/technology/tech/oci/instantclient/instantclient.html
Microsoft SQL Server	Microsoft SQL Server Driver for JDBC	http://www.microsoft.com/downloads/
QL Server and Sybase JDBC driver	jTDS (S)	http://www.sourceforge.net/

- Copy the JDBC drivers for your data store to the Nsure Audit Java classpath or a subdirectory thereof. See Table 7-6 on page 81 for the Nsure Audit Java classpath directories.
- If you are going to query a JDBC data store in iManager, copy all required JDBC drivers (`*.jar`) to the following iManager classpaths on your iManager server:
  - **NetWare:** `sys:\tomcat\4\common\lib`
  - **Linux and Solaris:** /var/opt/novell/tomcat4/common/lib
  - **Windows:** `\program files\novell\tomcat\common\lib`
- If you are using the JDBC Channel on a Windows machine, add the `jvm.dll` directory path to the Path system variable. For example, `c:\j2sdk1.4.2_09\jre\bin\server\`. You must reboot the machine for the changes to take effect.
- On Linux/Solaris, the LD_LIBRARY_PATH variable must point to the paths for `libverify.so`, and `libjvm.so`. You must reboot the machine for the changes to take effect.
- On Linux and Solaris platforms, export LD_LIBRARY_PATH to the path of the server JVM. To do this, create `/etc/profile.local` (if it does not exist), then add an export line similar to the following:

  ```
  export LD_LIBRARY_PATH=/usr/lib/java/jre/lib/i386/server:/usr/lib/
  java/jre/lib/i386/
  ```

  Replace `/usr/lib/java` with the full path to the Java runtime environment, for example, `/usr/lib/SunJava2-1.4.1`.
- When creating the JDBC channel object in iManager, Java classpath entries must be separated by a colon if your JDBC data store is hosted on Linux or Solaris. If your JDBC data store is hosted on NetWare or Windows, Java classpath entries must be separated by a semicolon.

For additional information on configuring the JDBC channel, see Appendix F, “Using JDBC Data Stores with Nsure Audit,” on page 231.

7.7.2 JDBC Channel Object

The JDBC Channel object stores the information the JDBC driver needs to write events to a JDBC-enabled data store.

The following table provides a description of each Channel object attribute.

IMPORTANT: You must restart the logging server to effect any changes in Channel object configuration. For more information, see Section G.3, “Secure Logging Server Startup Commands,” on page 236

Table 7-8 *JDBC Channel Object Attributes*

Attribute	Description
Configuration	

Attribute	Description
JDBC Class	Package and name of the Java Class providing JDBC connectivity. The following are Java class examples for the most common JDBC drivers: • **MySQL:** `com.mysql.jdbc.Driver` • **Oracle:** `oracle.jdbc.driver.OracleDriver` • **SQL Server:** `com.microsoft.jdbc.sqlserver.SQLServerDriver`
JDBC URL	Valid JDBC URL for the target data store, including the table name. The following are JDBC URLs for the most common JDBC drivers: • **MySQL:** `jdbc:mysql://192.168.0.5/naudit` • **Oracle:** `jdbc:oracle:thin:@`*`ip_address`*`:`*`port`*`:`*`sid`* • **SQL Server:** `jdbc:microsoft:sqlserver://`*`ip_address`*`:`*`port`*`;DatabaseName=`*`database_name`*
Username	The username the JDBC driver requires to log in to the data store.
Password	The password the JDBC driver requires to log in to the data store.
JDBC Table	Name of the table used to log Nsure Audit events.
JDBC Table Create SQL	If the table specified in the JDBC Table parameter does not exist, use SQL commands to create the table.
Max Data Size	The maximum size (in bytes) of information that can be written at one time to the data store.
Status	Allows you to enable or disable the Channel object. By default, all Channel objects are enabled. This means that the logging server loads the Channel object's configuration in memory at startup. The Channel object must be located in a supported Channel container for the logging server to use it. For more information on the logging server's Channel Container property, see "Logging Server Objects" on page 45. If you select the *Disabled* option, you must restart the Secure Logging Server for the setting to become effective. Thereafter, the logging server cannot load the object's configuration until you select *Enabled*. For information on unloading the logging server, see Section G.3, "Secure Logging Server Startup Commands," on page 236.

7.8 Microsoft SQL Server

The Microsoft SQL Server channel allows the logging server to log events to a Microsoft SQL Server database. The logging server can use the Microsoft SQL Server channel to create the central data store or a filtered database.

IMPORTANT: Native connections to a Microsoft SQL server database are available only when running the Secure Logging Server on a Windows platform. JDBC must be used to connect to a Microsoft SQL Server database from other platforms.

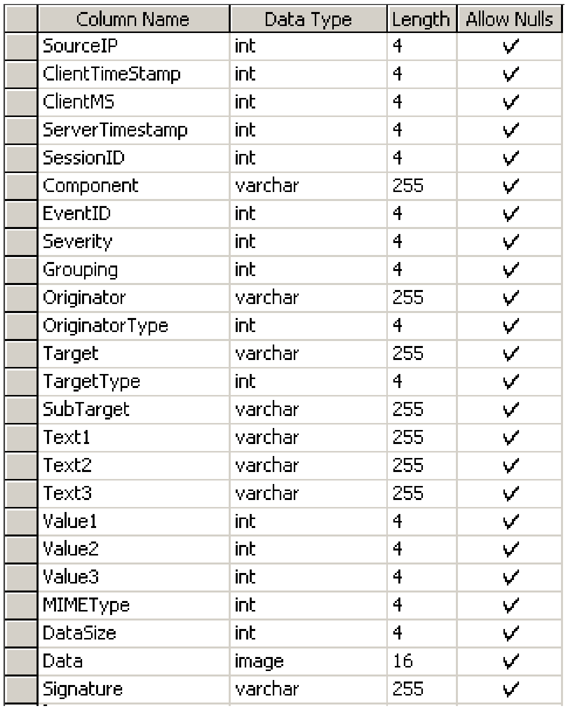
The space you need for your database depends on a number of factors. These include, but are not limited to, how many events per second you are storing and how long you want to keep the data. For the data store, a system that generates around 80 events per second with an average event size of 80 bytes consumes approximately 500 MB of disk space for the database table and 150 MB for the index in a 24-hour period.

IMPORTANT: Microsoft SQL Server limits the size of the data field to 10,484 bytes. If the size of the data field logged by the Platform Agent exceeds 10,484 bytes, the Microsoft SQL Server channel driver truncates the data in the Data field.

7.8.1 Microsoft SQL Server Channel Driver

When the Microsoft SQL Server Channel object configuration is loaded in the logging server's memory, the Microsoft SQL Server channel driver, `lgdmssql`, automatically creates the following table structure for the SQL Server data store:

Figure 7-1 *Microsoft SQL Server Table Structure*



Column Name	Data Type	Length	Allow Nulls
SourceIP	int	4	✓
ClientTimeStamp	int	4	✓
ClientMS	int	4	✓
ServerTimestamp	int	4	✓
SessionID	int	4	✓
Component	varchar	255	✓
EventID	int	4	✓
Severity	int	4	✓
Grouping	int	4	✓
Originator	varchar	255	✓
OriginatorType	int	4	✓
Target	varchar	255	✓
TargetType	int	4	✓
SubTarget	varchar	255	✓
Text1	varchar	255	✓
Text2	varchar	255	✓
Text3	varchar	255	✓
Value1	int	4	✓
Value2	int	4	✓
Value3	int	4	✓
MIMEType	int	4	✓
DataSize	int	4	✓
Data	image	16	✓
Signature	varchar	255	✓

The table name is defined in the Microsoft SQL Channel object configuration page. The default table name is `NAUDITLOG`.

To create this table manually, run the following, replacing *table_name* with the name you want to use for the table:

```
CREATE TABLE IF NOT EXISTS table_name
(SourceIP INT,
ClientTimestamp INT,
ClientMS INT,
ServerTimestamp INT,
SessionID INT,
Component VARCHAR(255),
EventID INT,
Severity INT,
Grouping INT,
Originator VARCHAR(255),
OriginatorType INT,
Target VARCHAR(255),
TargetType INT,
SubTarget VARCHAR(255),
Text1 VARCHAR(255),
Text2 VARCHAR(255),
Text3 VARCHAR(255),
Value1 INT,
Value2 INT,
Value3 INT,
MIMEType INT,
DataSize INT,
Data MEDIUMBLOB,
Signature VARCHAR(255),
INDEX(ClientTimestamp),
INDEX(EventID))
TYPE=MYISAM
```

7.8.2 Microsoft SQL Server Channel Object

The Microsoft SQL Server Channel object stores the information the Microsoft SQL Server driver needs to write events to a Microsoft SQL Server database.

The following table provides a description of each Channel object attribute.

IMPORTANT: You must restart the logging server to effect any changes in Channel object configuration. For more information, see Section G.3, "Secure Logging Server Startup Commands," on page 236

Table 7-9 Microsoft SQL Server Channel Object Attributes

Attribute	Description
Configuration	
Server Name	The IP Address or host name of the database server.
Database	The name of the database to which the logging server writes events. This field is case sensitive. **IMPORTANT:** This database must already exist. The SQL Server channel driver does not create the database. If no database name is specified, the logging server looks for `NAudit`. for information on creating the database, see Section E.2, "Preparing the Microsoft SQL Server Database," on page 228.

Attribute	Description
Table	The name of the database table to which the logging server writes events. The SQL Server channel driver, `lgdmssql`, automatically creates this table when the logging server first loads the current Channel object configuration in memory. For information on the table structure, see Section 7.8.1, "Microsoft SQL Server Channel Driver," on page 84. Do not use hyphens, spaces, or other special characters in the table name. The default table name is `NAUDITLOG`.
User Name	The user name for the account the logging server uses to authenticate with the database. This account adds records to the Microsoft SQL database.
Password	The password for the account the logging server uses to authenticate with the database.
Use SSL	(Optional) Select whether SSL should be used to encrypt data transferred between the Secure Logging Server and the Microsoft SQL server.
Status	Allows you to enable or disable the Channel object. By default, all Channel objects are enabled. This means that the logging server loads the Channel object's configuration in memory at startup. The Channel object must be located in a supported Channel container for the logging server to use it. For more information on the logging server's Channel Container property, see "Logging Server Objects" on page 45. If you select the *Disabled* option, you must restart the Secure Logging Server for the setting to become effective. Thereafter, the logging server cannot load the object's configuration until you select *Enabled*. For information on unloading the logging server, see Section G.3, "Secure Logging Server Startup Commands," on page 236.

7.9 MySQL

The MySQL channel allows the logging server to log events to a MySQL database. The logging server can use the MySQL channel to create the central data store or a filtered database.

The space you need for your database depends on a number of factors. These include, but are not limited to, how many events per second you are storing and how long you want to keep the data. The MySQL install, itself, is about 20 MB. (Keep in mind that the MySQL database does not need to be on the same volume as the MySQL binaries.) For the data store, a system that generates around 80 events per second with an average event size of 80 bytes consumes approximately 500 MB of disk space for the database table and 150 MB for the index in a 24-hour period.

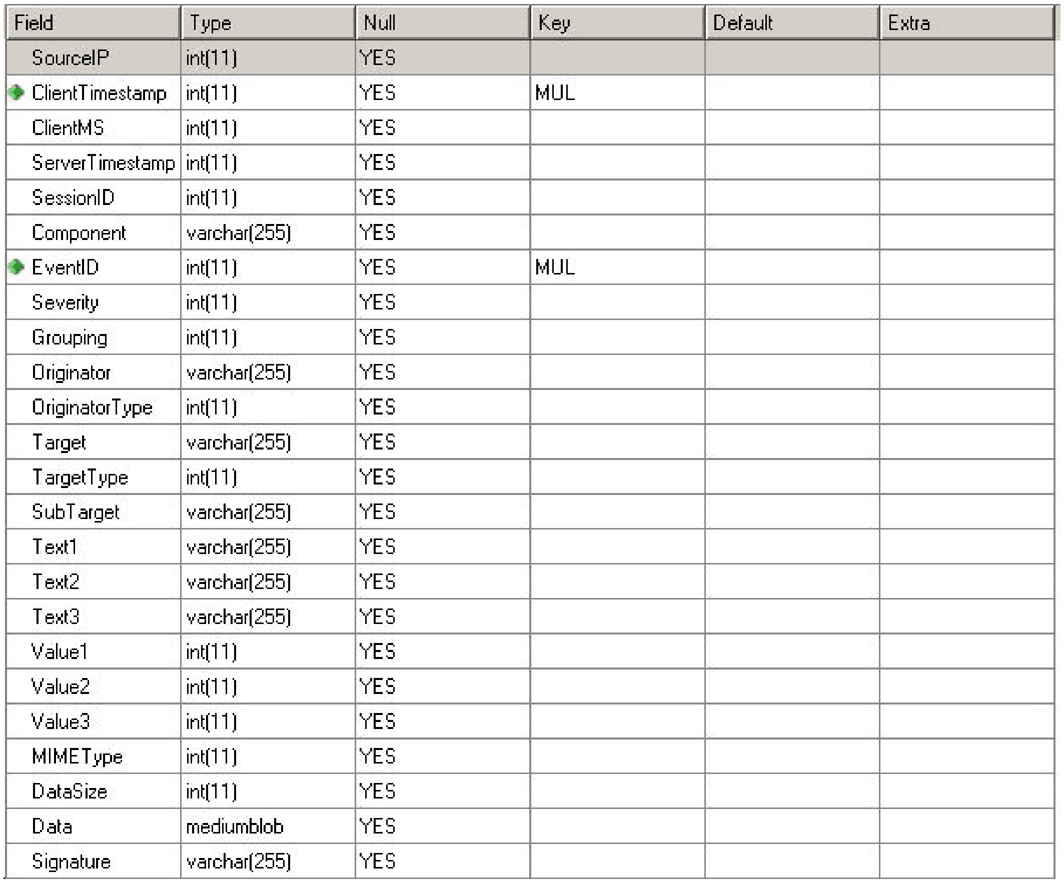
NOTE: To enable the MySQL channel, the MySQL client library, `libmysql`, is installed with the Secure Logging Server.

For further information, see Appendix C, "Using MySQL with Nsure Audit," on page 211.

7.9.1 MySQL Channel Driver

When the MySQL Channel object configuration is loaded in the logging server's memory, the MySQL channel driver, `lgdmsql`, automatically creates the following table structure for the MySQL data store:

Figure 7-2 *MySQL Table Structure*



Field	Type	Null	Key	Default	Extra
SourceIP	int(11)	YES			
◆ ClientTimestamp	int(11)	YES	MUL		
ClientMS	int(11)	YES			
ServerTimestamp	int(11)	YES			
SessionID	int(11)	YES			
Component	varchar(255)	YES			
◆ EventID	int(11)	YES	MUL		
Severity	int(11)	YES			
Grouping	int(11)	YES			
Originator	varchar(255)	YES			
OriginatorType	int(11)	YES			
Target	varchar(255)	YES			
TargetType	int(11)	YES			
SubTarget	varchar(255)	YES			
Text1	varchar(255)	YES			
Text2	varchar(255)	YES			
Text3	varchar(255)	YES			
Value1	int(11)	YES			
Value2	int(11)	YES			
Value3	int(11)	YES			
MIMEType	int(11)	YES			
DataSize	int(11)	YES			
Data	mediumblob	YES			
Signature	varchar(255)	YES			

The table name is defined in the MySQL Channel object configuration page. The default table name is `NAUDITLOG`.

The MySQL Channel uses MyIsam as its database engine; therefore, the default maximum table size using MySQL 4.1 is 4 GB. MySQL 5.0 limits table sizes to 65,536 TB. Table size can be further constrained by the maximum file size your operating system can manage.

NOTE: If you need larger tables, use the `max_rows` and `avg_row_length` commands in the MySQL Channel object's Create Table Options property.

7.9.2 MySQL Channel Object

The MySQL Channel object stores the information the MySQL driver needs to write events to a MySQL database.

The following table provides a description of each Channel object attribute.

IMPORTANT: You must restart the logging server to effect any changes in Channel object configuration. For more information, see Section G.3, "Secure Logging Server Startup Commands," on page 236

Table 7-10 *MySQL Channel Object Attributes*

Attribute	Description
Configuration	
Database	
Host	The IP Address or host name of the database server. If a host name is specified, only the first address associated with that name is used. If the MySQL channel driver loses its connection with the database server, it tries to reconnect every second for 30 seconds. If it cannot reconnect, the driver stores its current events in memory, but it does not accept any new events until the connection is restored. Incoming events are either stored in the Platform Agents' Disconnected Mode Cache (in the case of the central data store) or dropped (in the case of a Notification Filter database).
Name	The name of the database to which the logging server writes events. **IMPORTANT:** This database must already exist. The SQL Server channel driver does not create the database. If no database name is specified, the logging server looks for `NAudit`. for information on creating the database, see Section C.3, "Preparing the MySQL Database," on page 212.
Table	The name of the database table to which the logging server writes events. The MySQL driver, `lgdmsql`, automatically creates this table when the logging server first loads the current Channel object configuration in memory. For information on the table structure, see "MySQL Channel Driver" on page 87. Do not use hyphens, spaces, or other special characters in the table name. The default table name is `NAUDITLOG`.
User	The user account the logging server uses to log in to the database. On NetWare 6.5, MySQL installs in Secure Mode. The default username for the NetWare 6.5 data store is `auditusr`. (This default can be changed during the installation of Nsure Audit.) This account has all privileges to the default database (naudit) and can log in from any IP address. In Secure Mode, the default MySQL administrative account, Root, only has rights to log in at the database server. Therefore, if MySQL is running in Secure Mode and you want the logging server to use the Root account to log in to the database, MySQL and the Secure Logging Server must be located on the same server and you must specify a loopback address ("127.0.0.1" or "localhost") in the Address field.
Password	The password the logging server uses to authenticate with the database. The default password for the NetWare 6.5 data store is `auditpwd`. (This default can be changed during the installation of Nsure Audit.)

Attribute	Description
Test Credentials	This option tests the MySQL channel configuration to verify the MySQL driver (`lgdmsql`) can connect to the database. When you click the Test Credentials link, you are prompted for the JDBC Class. The JDBC Class is the package and name of the Java Class providing JDBC connectivity. Provide the required information, then click OK. The MySQL driver tests the MySQL Channel object configuration by attempting to make a connection to the MySQL database.
Advanced	
CREATE TABLE Options	This property allows you to customize the default table structure using standard SQL Create Table commands. For example, the `max_rows` and `avg_row_length` commands can be used to increase the maximum size of your table as follows: `max_rows=200000000 avg_row_length=76` **NOTE:** When all the rows available to MySQL are used, the logging channel starts logging error messages to the screen. If the database is full during startup, the logging channel does not load at all. If MySQL is the default logging channel, the logging server fails to load entirely. Use an expiration command to avoid this issue. For more information, see Section C.9, “SQL Expiration Command Variables,” on page 215.

SQL Expiration Commands	This property enables you to use SQL Expiration commands to automate database maintenance.
	For example, you can automate data archiving by configuring the MySQL channel to automatically save out the current table and create a new table at designated intervals.
	For a listing of command variables and sample scripts, see Section C.9, “SQL Expiration Command Variables,” on page 215.
	Use a semicolon (;) to separate multiple commands that must be executed in sequence. If the commands can be executed in any order, no semicolon is needed.
	WARNING: If you choose the Autoconfigure for MySQL option in the NetWare 6.5 install, the installation program automatically creates the MySQL Channel object with a default Expiration script that runs every night at midnight and automatically deletes every record older than 12 hours. This is done because the default events logged by the NetWare and eDirectory Instrumentations quickly fill the database. To remove this setting, simply delete the script from the SQL Expiration Commands property and restart the Secure Logging Server. The script reads as follows:
	`DELETE FROM $1 WHERE`
	`clienttimestamp<(unix_timestamp()-43200);`

Attribute	Description
Expire at specified time or interval	The frequency at which the expiration command script is executed. For daily regimens, select a time of day. (00 is midnight.) For weekly regimens, select a day of the week. The expiration commands are executed at midnight on that day. For monthly regimens, the expiration commands are executed at midnight on the first day of the month.
Status	Allows you to enable or disable the Channel object. By default, all Channel objects are enabled. This means that the logging server loads the Channel object's configuration in memory at startup. The Channel object must be located in a supported Channel container for the logging server to use it. For more information on the logging server's Channel Container property, see "Logging Server Objects" on page 45. If you select the *Disabled* option, you must restart the Secure Logging Server for the setting to become effective. Thereafter, the logging server cannot load the object's configuration until you select *Enabled*. For information on unloading the logging server, see Section G.3, "Secure Logging Server Startup Commands," on page 236.

7.10 Oracle

The Oracle channel allows the logging server to log events to an Oracle database. The logging server can use the Oracle channel to create the central data store or a filtered database.

The Oracle channel driver is used only on platforms where Oracle can run natively, such as Windows, Linux, and Solaris. If you are running the Secure Logging Server on NetWare, create a JDBC channel to connect to the Oracle server. For more information, see Section 7.7, "JDBC," on page 80.

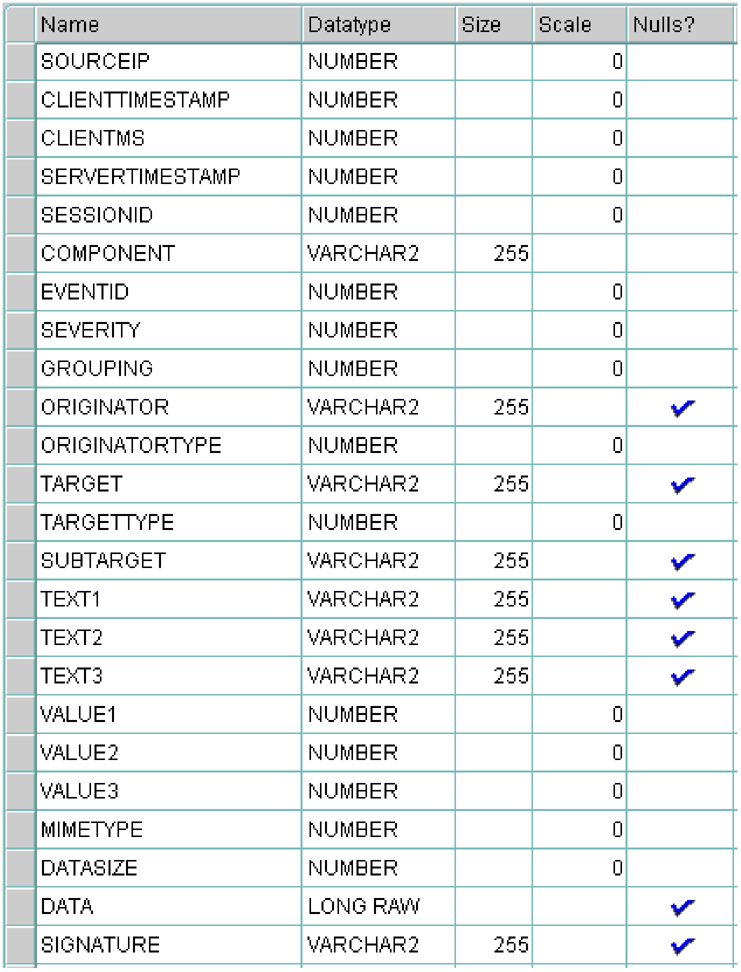
NOTE: On Linux and Solaris systems, the 32-bit version of the Oracle client is required. The 64-bit version does not work because Nsure Audit is compiled as a 32-bit application.

For the Oracle channel to function properly, you must install the Oracle client libraries on the same server as the Secure Logging Server.

7.10.1 Oracle Channel Driver

When the Oracle Channel object configuration is loaded in the logging server's memory, the Oracle channel driver, `lgdora`, automatically creates the following table structure for the Oracle data store:

Figure 7-3 *Oracle Table Structure*



Name	Datatype	Size	Scale	Nulls?
SOURCEIP	NUMBER		0	
CLIENTTIMESTAMP	NUMBER		0	
CLIENTMS	NUMBER		0	
SERVERTIMESTAMP	NUMBER		0	
SESSIONID	NUMBER		0	
COMPONENT	VARCHAR2	255		
EVENTID	NUMBER		0	
SEVERITY	NUMBER		0	
GROUPING	NUMBER		0	
ORIGINATOR	VARCHAR2	255		✓
ORIGINATORTYPE	NUMBER		0	
TARGET	VARCHAR2	255		✓
TARGETTYPE	NUMBER		0	
SUBTARGET	VARCHAR2	255		✓
TEXT1	VARCHAR2	255		✓
TEXT2	VARCHAR2	255		✓
TEXT3	VARCHAR2	255		✓
VALUE1	NUMBER		0	
VALUE2	NUMBER		0	
VALUE3	NUMBER		0	
MIMETYPE	NUMBER		0	
DATASIZE	NUMBER		0	
DATA	LONG RAW			✓
SIGNATURE	VARCHAR2	255		✓

The table name is defined in the Oracle Channel object configuration page. The default table name is `NAUDITLOG`.

7.10.2 Oracle Channel Object

The Oracle Channel object stores the information the Oracle driver needs to write events to an Oracle database.

The following table provides a description of each Channel object attribute.

IMPORTANT: You must restart the logging server to effect any changes in Channel object configuration. For more information, see Section G.3, "Secure Logging Server Startup Commands," on page 236.

Table 7-11 *Oracle Channel Object Attributes*

Attribute	Description
Configuration	
Database	
Name	The transport name in the `TNSNAMES.ORA` file used by the Oracle client to connect to the Oracle database.
Table	The name of the database table to which the logging server writes events. The Oracle Server channel driver, `lgdora`, automatically creates this table when the logging server first loads the current Channel object configuration in memory. For information on the table structure, see Section 7.10.1, "Oracle Channel Driver," on page 91. Do not use hyphens, spaces, or other special characters in the table name. The default table name is `NAUDITLOG`. **NOTE:** To use the Nsure Audit Reporting Application's reports with an Oracle database, you must create a table view. For information on this procedure, see Section D.9, "Creating a View in Oracle," on page 224.
User	The user name for the account that has access to the naudit tablespace. The default username is `auditusr`.
Password	The password for the `audituser` account, which the logging server uses to authenticate with the database.
Test Credentials	This option tests the Oracle channel configuration to verify the Oracle driver (`lgdora`) can connect to the database. When you click the *Test Credentials* link, you are prompted for the following: • **JDBC Class:** The package and name of the Java Class providing JDBC connectivity. • **Host:** The IP Address or host name of the database server. If a host name is specified, only the first address associated with that name is used. Provide the required information, then click *OK*. The Oracle driver tests the Oracle Channel object configuration by attempting to make a connection to the Oracle database.

Attribute	Description
Status	Allows you to enable or disable the Channel object. By default, all Channel objects are enabled. This means that the logging server loads the Channel object’s configuration in memory at startup. **IMPORTANT:** The Channel object must be located in a supported Channel container for the logging server to use it. For more information on the logging server’s Channel Container property, see “Logging Server Objects” on page 45. If you select the *Disabled* option, you must restart the Secure Logging Server for the setting to become effective. Thereafter, the logging server cannot load the object’s configuration until you select *Enabled*. For information on unloading the logging server, see Section G.3, “Secure Logging Server Startup Commands,” on page 236.

7.11 SMTP

The SMTP channel allows the logging server to e-mail logged events. Typically, the SMTP channel is used to e-mail system critical events, such as a server abend, to a system administrator’s mailbox, cell phone, or other e-mail enabled device. To enable e-mail notification, administrators must configure a notification filter and select the SMTP Channel object as one of its Notification Channels. For more information, see Section 8.3, “Notification Filters,” on page 102.

7.11.1 SMTP Channel Driver

At startup, the SMTP driver, lgdsmtp, performs a server check; that is, the driver attempts to connect to the designated host at port 25. The server check verifies the driver can communicate with the server before it attempts to relay events. If the server check fails, the SMTP driver refuses to load. This is done to ensure that no events are lost because of misconfiguration. For troubleshooting purposes, this event is written to the log file. For example, in the case of NetWare, the etc/logdir/log would have the following entry:

```
lgdsmtp.nlm failed to load Error Code -6
```

The SMTP driver does not buffer events that are undeliverable because of a misconfiguration or a server failure.

The SMTP channel driver can send events through servers that require SMTP authentication as long as a valid username and password are defined in the Channel object configuration. If the username and password are configured, the SMTP driver always attempts SMTP authentication when connecting with the relay host. It does not, however, distinguish whether or not this authentication is actually successful and it tries to send the message in either case.

The SMTP driver cannot authenticate with an SMTP server on which SMTP-after-POP is enabled.

7.11.2 SMTP Channel Object

The SMTP Channel object stores the information the SMTP driver needs to relay events through an SMTP server.

The following table provides a description of each Channel object attribute.

IMPORTANT: You must restart the logging server to effect any changes in Channel object configuration. For more information, see Section G.3, “Secure Logging Server Startup Commands,” on page 236.

Table 7-12 *SMTP Channel Object Attributes*

Attribute	Description
Configuration	
SMTP Relay Settings	
Host	The host name or IP address of the SMTP server. If a host name is specified, only the first address associated with that name is used.
User	The user name for the e-mail account the SMTP channel uses to connect to the SMTP server. The user name is required only if SMTP Authentication is enabled on the SMTP server.
Password	The password for the e-mail account the SMTP channel uses to connect to the SMTP server. The password is required only if SMTP Authentication is enabled on the SMTP server.
Message Settings	
Sender	The name that appears in the From: line for all messages sent from this SMTP Channel object. Some SMTP servers require this field to be a valid e-mail address. If this is required by your SMTP server, make sure you provide a valid e-mail address, such as *username@yourcompany.com*.
Recipient	The e-mail addresses to which all events directed through this SMTP Channel object are sent. Multiple recipients are delineated with a comma (,), a space, or a semicolon (;). You can also use the $S or $T event variables in this field instead of a real address. When relaying an event, the SMTP driver replaces these variables with the value of the event’s *Text1* or *Text2* fields, respectively. (See Section A.3, “Managing Event Data,” on page 181 for more information.) As long as the value of the *Text1* or *Text2* field is an e-mail address, the $S and $T variables can be leveraged to automatically send notification messages for user-related events such as a password change. **IMPORTANT:** To use the $S and $T variables, you must also configure a Notification Filter that directs only those events with an e-mail address in the Text1 or Text2 fields to this SMTP Channel object. For information on configuring Notification Filters, see Chapter 8, “Configuring Filters and Event Notifications,” on page 101.

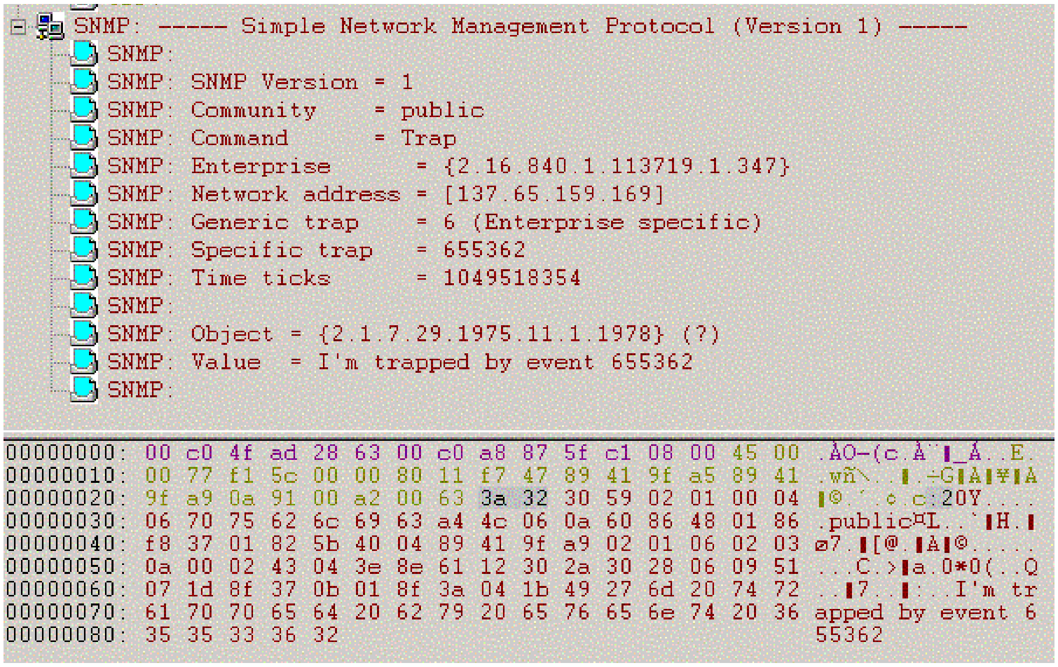
Attribute	Description
Subject	The text that appears in the Subject line for all messages sent from this SMTP Channel object. The subject line can contain up to 255 characters. The subject line can also contain event variables. The SMTP driver replaces these variables with a value from the event's designated field. For a listing of event variables, see Section A.3, "Managing Event Data," on page 181. This field is optional.
Message	The text that appears in the message body for all messages sent from this SMTP Channel object. The message body can be up to 64 KB; however, for performance reasons, this is not recommended. The message body can contain event variables. The SMTP driver replaces these variables with a value from the event's designated field. For a listing of event variables, see Section A.3, "Managing Event Data," on page 181. This field is optional.
Status	Allows you to enable or disable the Channel object. By default, all Channel objects are enabled. This means that the logging server loads the Channel object's configuration in memory at startup. The Channel object must be located in a supported Channel container for the logging server to use it. For more information on the logging server's Channel Container property, see "Logging Server Objects" on page 45. If you select the *Disabled* option, you must restart the Secure Logging Server for the setting to become effective. Thereafter, the logging server cannot load the object's configuration until you select *Enabled*. For information on unloading the logging server, see Section G.3, "Secure Logging Server Startup Commands," on page 236.

7.12 SNMP

The SNMP channel allows the logging server to send filtered events to an SNMP management system.

A decoded SNMP trap appears as follows:

Figure 7-4 *Decoded SNMP Trap*



```
SNMP: ----- Simple Network Management Protocol (Version 1) -----
  SNMP:
  SNMP: SNMP Version = 1
  SNMP: Community    = public
  SNMP: Command      = Trap
  SNMP: Enterprise      = {2.16.840.1.113719.1.347}
  SNMP: Network address = [137.65.159.169]
  SNMP: Generic trap    = 6 (Enterprise specific)
  SNMP: Specific trap   = 655362
  SNMP: Time ticks      = 1049518354
  SNMP:
  SNMP: Object = {2.1.7.29.1975.11.1.1978} (?)
  SNMP: Value  = I'm trapped by event 655362
  SNMP:

00000000: 00 c0 4f ad 28 63 00 c0 a8 87 5f c1 08 00 45 00 .ÀO-(c.À¨▮_Á..E.
00000010: 00 77 f1 5c 00 00 80 11 f7 47 89 41 9f a5 89 41 .wñ\..▮.÷G▮A▮¥▮A
00000020: 9f a9 0a 91 00 a2 00 63 3a 32 30 59 02 01 00 04 ▮©.'.¢.c:20Y....
00000030: 06 70 75 62 6c 69 63 a4 4c 06 0a 60 86 48 01 86 .public¤L..`▮H.▮
00000040: f8 37 01 82 5b 40 04 89 41 9f a9 02 01 06 02 03 ø7.▮[@.▮A▮©.....
00000050: 0a 00 02 43 04 3e 8e 61 12 30 2a 30 28 06 09 51 ...C.>▮a.0*0(..Q
00000060: 07 1d 8f 37 0b 01 8f 3a 04 1b 49 27 6d 20 74 72 ..▮7..▮:..I'm tr
00000070: 61 70 70 65 64 20 62 79 20 65 76 65 6e 74 20 36 apped by event 6
00000080: 35 35 33 36 32                                  55362
```

The trap values are explained in following table.

Table 7-13 *SNMP Trap Values*

SNMP Value	Description
SNMP Version	The trap’s SNMP version. The Nsure Audit SNMP driver sends SNMPv1 traps.
Community	The string, or password, needed to access the SNMP management system.
Command	The SNMP command. This is always Trap.
Enterprise	The Enterprise that sent the event is always 2.16.840.1.113719.1.347.3.1.
Network address	The IP address of the logging server that sent the trap.
Generic trap	The Generic Trap field is always 6 (Enterprise specific).
Specific trap	The Specific Trap field always contains the Event ID of the event that triggered the trap.
Time Ticks	The time the event was sent in seconds since 1970.
Object	The Object ID specified in the SNMP Channel object. If no Object ID is specified, the Nsure Audit internal OID is used (2.16.840.1.113719.1.347.3.1).
Value	The Value associated with the Object is the message configured in the SNMP Channel object.

7.12.1 SNMP Channel Driver

The SNMP driver, `lgdsnmp`, sends SNMPv1 traps.

The SNMP driver does not buffer traps that are undeliverable because of a misconfiguration or a server failure.

7.12.2 SNMP Channel Object

The SNMP Channel object stores the information the SNMP driver needs to send traps to an SNMP management system.

The following table provides a description of each Channel object attribute.

IMPORTANT: You must restart the logging server to effect any changes in Channel object configuration. For more information, see Section G.3, "Secure Logging Server Startup Commands," on page 236

Table 7-14 *SNMP Channel Object Attributes*

Attribute	Description
Configuration	
Send trap to host	The host name or IP address of the SNMP management system. If a host name is specified, only the first address associated with that name is used.
Community string for trap	The community string (password) needed to access the SNMP management system. If no community string is specified, the driver defaults to `public`.
Object ID	The object you wish to associate with this message. You should provide your own asn1 object id. If no Object ID is specified, the Nsure Audit internal OID is used (2.16.840.1.113719.1.347.3.1). The Nsure Audit OID is under the `CCITT/US/novell` tree.
Message	The text that appears in the message body for all traps sent from this SNMP Channel object. Because SNMP specifications require that an SNMP packet can be no larger than 500 bytes, the message body is limited to 300 bytes. The SNMP driver simply truncates anything over 300 bytes. The message body can contain event variables. The SNMP driver replaces these variables with a value from the event's designated field. For a listing of event variables, see Section A.3, "Managing Event Data," on page 181. This field is optional.

Attribute	Description
Status	Allows you to enable or disable the Channel object. By default, all Channel objects are enabled. This means that the logging server loads the Channel object's configuration in memory at startup. The Channel object must be located in a supported Channel container for the logging server to use it. For more information on the logging server's Channel Container property, see "Logging Server Objects" on page 45. If you select the *Disabled* option, you must restart the Secure Logging Server for the setting to become effective. Thereafter, the logging server cannot load the object's configuration until you select *Enabled*. For information on unloading the logging server, see Section G.3, "Secure Logging Server Startup Commands," on page 236.

7.13 Syslog

The Syslog channel allows the logging server to log events to a specific syslog facility on any syslog host or to a remote syslog daemon.

It is also capable of creating localized logs. If the logging applications have localized Log Schema files and if those files are added to their respective Application objects, the Syslog channel can write the log files in the language designated in the Syslog Channel object.

The Log Schema catalogs the events that can be logged for a given application. It can also provide event descriptions and labels for the event fields. For more information, see Section A.4, "Log Schema Files," on page 187.

The logging server can use the Syslog channel to write to the central data store or to create filtered log files.

IMPORTANT: On the SUSE® SLES platform, the syslog daemon must be restarted with the `-r` option before it can accept log events, even when logging locally.

7.13.1 Syslog Channel Driver

At startup, the Syslog driver, `lgdsyslg`, loads each application's log schema. If a logging application has multiple language versions of its log schema, the Syslog channel loads the schema for the language designated in the Syslog Channel object.

Nsure Audit stores log schema files as attributes in their respective Application objects. For further information, see Section A.4, "Log Schema Files," on page 187.

NOTE: If the File and Syslog Channel objects reference the same language, the drivers independently load the log schema in their own memory. The only time the log schema is shared is between multiple instances of the same driver. For example, if you have two Syslog channels configured to write log files in English, the English log schema for each application is loaded only once.

When it writes events to the syslog facility, the Syslog driver uses the Event ID to look up each event in the corresponding application's log schema and then it writes the event description to the data store. If the log schema isn't available, or if there isn't a descriptive entry for the current event, the Syslog channel defaults to the following format:

```
$DC $TC,$SO,$NI,$NL,$NG,$N1,$N2,$SS,$ST\n
```

(Client Date and Time Stamp, Component, EventID, Log Level, Group ID, Value1, Value2, Value3, Text1, Text2, Text3.) See Section A.3, "Managing Event Data," on page 181 for an explanation of each field and format variable.

Because it uses the log schema to log events, the Syslog driver is also capable of creating localized logs. If a logging application has localized log schema files and if those files are added to the Application object, the Syslog driver uses the log schema for the language designated in the Syslog Channel object to write the event descriptions.

For more information on the Syslog channel's language attribute, see "Syslog Channel Object" on page 99. For information on localized log schema files, see "Localized Log Schema Files" on page 189.

7.13.2 Syslog Channel Object

The Syslog Channel object stores the information the Syslog driver needs to write events to syslog.

The following table provides a description of each Channel object attribute.

IMPORTANT: You must restart the logging server to effect any changes in Channel object configuration. For more information, see Section G.3, "Secure Logging Server Startup Commands," on page 236

Table 7-15 *Syslog Channel Object Attributes*

Attribute	Description
Configuration	
Syslog host	The host name or IP address of the syslog server. If a host name is specified, only the first address associated with that name is used. The syslog server must be running a syslog daemon that allows remote connections for log drop-off. UNIX syslog daemons accept remote connections by default; however, Linux system daemons do not. Therefore, the startup script for Linux syslog daemons must be altered to explicitly allow remote connections. This is done using the `-r` switch on `syslogd`.
Facility	The syslog facility to which the logging server writes events.

Attribute	Description
Translated language	The language in which events are written to syslog. If a logging application has localized Log Schema files and if those files are added to the Application object, the Syslog channel can write log files in the selected language. If there isn't a log schema for the selected language, the channel defaults to English. Log Schema files (`*.lsc)` catalog the events that can be logged for a given application. They can also provide event descriptions and labels for the event fields. For more information, see Section A.4, "Log Schema Files," on page 187. If the log schema isn't available, or if there isn't a descriptive entry for the current event, the Syslog channel defaults to the following format: `$DC $TC,$SO,$NI,$NL,$NG,$N1,$N2,$SS,$ST\n` (EventID, Log Level, Group ID, Value1, Value2, Value3, Text1, Text2, Text3) Technically, only English is allowed because syslog is a 7-bit protocol. However, most syslog implementations support 8-bit, so all 8-bit languages can be selected. However, some 8-bit languages, such as Russian, are not very usable in syslog. No 16-bit languages are allowed. You can create parallel logs in multiple languages by defining multiple Syslog Channel objects with different languages and having a single notification filter pass all events to those channels.
Status	Allows you to enable or disable the Channel object. By default, all Channel objects are enabled. This means that the logging server loads the Channel object's configuration in memory at startup. The Channel object must be located in a supported Channel container for the logging server to use it. For more information on the logging server's Channel Container property, see "Logging Server Objects" on page 45. If you select the *Disabled* option, you must restart the Secure Logging Server for the setting to become effective. Thereafter, the logging server cannot load the object's configuration until you select *Enabled*. For information on unloading the logging server, see Section G.3, "Secure Logging Server Startup Commands," on page 236.

Configuring Filters and Event Notifications

8

Using filters and event notifications, Novell® Nsure™ Audit 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.

By default, your main logging channel receives all events. Notifications are required only if you want to send specific events to a channel other than your main logging channel. Do not send notifications to your main logging channel, as this results in duplicate logged events.

- Section 6.1, “Overview,” on page 65
- Section 8.2, “Creating Notification Objects,” on page 101
- Section 8.3, “Notification Filters,” on page 102
- Section 8.4, “Heartbeat Objects,” on page 104

8.1 Overview

Nsure Audit provides two kinds of event notification:

- Filtered Notification
- Heartbeat Notification

Filtered notification tells you when a specific event has occurred; heartbeat notification tells you when an event has not occurred.

As the name implies, Notification Filter objects filter specific events from the stream of incoming events. The filtered events are then routed to one or more channel drivers where they can be logged to a database, routed to a Java application or SNMP management system, or broadcast to an administrator via SMTP. In some cases, filtered events may be directed to the CVR channel to trigger a reset policy.

Heartbeat objects monitor the stream of incoming events for the occurrence of a specific Event ID. If the event does not occur within the designated interval, the logging server generates a heartbeat event (EventID 0001001). This event is automatically logged to the central data store; however, if you want to receive notification that a specific event has not occurred, you must create a Notification Filter for the corresponding heartbeat event.

8.2 Creating Notification Objects

Both filtered and heartbeat notifications are configured in Novell eDirectory™ using Notification Filter and Heartbeat objects.

Notification Filter objects store the criteria the logging server uses to filter system events. They also designate which Channel objects the logging server uses to provide event notification.

Heartbeat objects define which Event IDs the logging server is looking for and the interval at which those events must occur. You can also define the information that is returned in the heartbeat event's Text1, Text2, Value1, and Value2 fields.

You must create a separate Notification Filter object for every event you want to filter. A single Heartbeat object, on the other hand, can monitor multiple events. In fact, you really only need to create one Heartbeat object in your logging system.

You can create Notification Filter and Heartbeat objects using iManager. For information on using iManager to create objects, see Chapter 3, "Nsure Audit iManager Plug-in," on page 35.

You must create Notification Filter and Heartbeat objects in Notification containers. The Notification container under Logging Services is automatically created during installation; however, additional Notification containers can be created anywhere in the tree.

Creating Notification objects in the central Notification container under Logging Services is ideal for organizations that need a simple, easy-to-manage logging system. It also suits organizations that are implementing Nsure Audit as an auditing solution and, for security reasons, want to centrally manage their system.

If you want to distribute logging system administration, however, Notification objects can be created anywhere in the tree. For example, if administration is divided by logging server, you can create a Notification container under each Logging Server object. On the other hand, if administration is divided by application (for example, one person manages logging for iChain®, another DirXML® logging, etc.), the Notification container can be created in any context assigned to its administrator.

If you create a Notification container elsewhere in the tree, you must add that container to the logging server's list of supported containers. At startup, the logging server scans its list of supported Notification containers and loads the included Notification object configurations in memory so it can filter events, monitor Heartbeat events, and route notifications to the appropriate channels. If a Notification object is not in one of the logging server's supported Notification containers, it cannot use it. For more information on the logging server's Notification Container property, see "Logging Server Objects" on page 45.

IMPORTANT: The logging server loads only the Notification object configurations at startup. Therefore, if you create a new Notification container or Notification object, you must first ensure the Notification container is included in the logging server's Notification Container list and then restart the logging server. For information on restarting the logging server, see Section G.3, "Secure Logging Server Startup Commands," on page 236.

8.3 Notification Filters

Notification Filter objects define event criteria and designate which Channel objects should be used to provide event notification.

To define Notification Filters, you must be familiar with event structure. For more information on each event field, see Section A.1, "Event Structure," on page 175.

When you define a Notification Filter, you specify a value for a given event field. To narrow the results, you can define values for multiple event fields. Using standard And, Or, and Not operators, you can define up to 15 event conditions.

After you define the event criteria, you must select a notification channel. Notification channels are simply the Channel objects the logging server uses to provide event notification. For example, if you want to e-mail events to your mailbox, you must select an SMTP Channel object that is configured to relay events to your e-mail address. Similarly, if you want to log events to a MySQL database, you must select a MySQL Channel object that is configured to write events to the correct database and table. You can define multiple notification channels for any given Notification object.

The following table provides a description of each Notification Filter attribute.

IMPORTANT: You must restart the logging server to effect any changes in Filter object configuration. For more information, see Section G.3, "Secure Logging Server Startup Commands," on page 236.

Attribute	Description
Description	This field allows you to enter a description and any necessary explanation for the Notification Filter. The field limit is 255 characters. The information from this field is returned if one uses the SE event variable. For more information, see Section A.3, "Managing Event Data," on page 181.
`Rule`	The Rule defines the filter criteria.
Event Field	The event field on which the logging server filters events. For more information on the event fields, see Section A.1, "Event Structure," on page 175.
Condition	The condition under which the logging server applies the Value to the Event Field. Depending on the Event Field, you can select one of the following conditions from the drop-down list box: • Matches • Is less • Is more • Is between • Contains
Value	The value for the designated Event Field. The logging server applies the Value to the designated Event Field under the defined conditions. If an event matches the criteria, it is sent to the designated notification channel.
Operator	To narrow the filter results, you can define values for multiple event fields. Using standard And, or, and Not operators, you can define up to 15 event conditions. The conditions are accumulative; that is, the logging server applies the first condition, then the second, then the third, etc., to progressively narrow the results.
`Notification Channels`	The Channel objects the logging server uses to provide event notification. You can select multiple notification channels for any given Filter object. Click the Object Selector button 🔍 to select Channel objects in the tree.

Attribute	Description
`Status`	This option allows you to enable or disable the Notification Filter. By default, all Notification Filters are enabled. This means that the logging server loads the filter's configuration in memory at startup. **IMPORTANT:** The Notification Filter object must be located in a supported Notification container for the logging server to use it. For more information on the logging server's Notification Container property, see "Logging Server Objects" on page 45. If you mark the Disabled option, you must restart the Secure Logging Server for the setting to become effective. Thereafter, the logging server cannot load the object's configuration until you mark Enabled. For information on unloading the logging server, see Section G.3, "Secure Logging Server Startup Commands," on page 236.

8.4 Heartbeat Objects

Heartbeat objects define which Event IDs the logging server is looking for and the interval at which those events must occur. You can also define the information that is returned in the heartbeat event's Text1, Text2, Value1, and Value2 fields.

If an event does not occur within the designated interval, the logging server generates a heartbeat event (EventID 0001001). The information in the Heartbeat object's Text1, Text2, Text3, Value1, Value2, and Value3 fields is used to populate the corresponding fields in the heartbeat event. The Notification Filter can differentiate heartbeat events based on the values you define in the Text1, Text2, Text3, Value1, Value2, and Value3 fields.

The heartbeat event is automatically logged to the central data store; however, if you want to receive notification that a specific event has not occurred, you must create a Notification Filter for the corresponding heartbeat event.

The following table provides a description of each Heartbeat object attribute.

IMPORTANT: You must restart the logging server to effect any changes in Heartbeat object configuration. For more information, see Section G.3, "Secure Logging Server Startup Commands," on page 236.

Attribute	Description
Description	This field allows you to enter a description and any necessary explanation for the Heartbeat object. The field limit is 255 characters.

Attribute	Description
Event ID	The Event ID you want the logging server to monitor. The Event ID uniquely identifies each type of logged event. For more information, see Section A.1, “Event Structure,” on page 175. If a logging application does not log the Event ID within the designated interval, the logging server generates a heartbeat event. **IMPORTANT:** The Event ID is not included in the heartbeat event. Therefore, you should enter information in the Text1, Text2, Text3, Value1, Value2, and Value3 fields that allows you to determine which event triggered the heartbeat event.
Interval	The maximum number of seconds between each event occurrence. If the event does not occur within the designated interval, the logging server generates a heartbeat event.
Text1	The information that appears in the heartbeat event's Text1 field. It can contain any text string up to 255 characters. To facilitate filtering of heartbeat events, the Text1, Text2, Value1, and Value2 fields should include information that allows you to identify which event triggered the heartbeat event.
Text2	The information that appears in the heartbeat event's Text2 field. It can contain any text string up to 255 characters.
Text3	The information that appears in the heartbeat event's Text3 field. It can contain any text string up to 255 characters.
Value1	The information that appears in the heartbeat event's Value1 field. It can contain any numeric value up to 32 bits.
Value2	The information that appears in the heartbeat event's Value2 field. It can contain any numeric value up to 32 bits.
Value3	The information that appears in the heartbeat event's Value3 field. It can contain any numeric value up to 32 bits.
Operators +-	Click the plus sign (+) to add a new line. Click the minus sign (-) to remove a line. Each line defines a separate event for the logging server to monitor. If a given event does not occur, the logging server generates a unique heartbeat event using the information from the Text1, Text2, Text3, Value1, Value2, and Value3 fields. There is no programmed limit to the number of events that can be added to Heartbeat objects.

Attribute	Description
`Status`	This option allows you to enable or disable a Heartbeat object. By default, all Heartbeat objects are enabled. This means that the logging server loads the object's configuration in memory at startup. **IMPORTANT:** The Heartbeat object must be located in a supported Notification container for the logging server to use it. For more information on the logging server's Notification Container property, see "Logging Server Objects" on page 45. If you mark the Disabled option, you must restart the Secure Logging Server for the setting to become effective. Thereafter, the logging server cannot load the object's configuration until you mark Enabled. For information on unloading the logging server, see Section G.3, "Secure Logging Server Startup Commands," on page 236.

Generating Queries and Reports

9

Logging information is only half the battle. Obviously, you have to be able to access and understand your log data for the information to be useful. Queries and reports allow you to view and interpret the information in your data store.

iManager and Nsure™ Audit Report are the primary tools used to run queries or reports on Novell® Nsure Audit data stores; however, depending on your system's data store, there are other ways to access your log data. The following sections provide the information you need to generate queries and reports from your Nsure Audit log data.

- Section 9.1, "Using iManager to Generate Queries," on page 107
- Section 9.2, "Using Nsure Audit Report," on page 131
- Section 9.3, "Using Other Utilities to Access Log Data," on page 156

9.1 Using iManager to Generate Queries

Novell iManager is a Web-based application that is used to manage, maintain, and monitor Novell eDirectory™ through wired and wireless devices. With the Nsure Audit plug-in module, iManager can be used to manage Nsure Audit objects in eDirectory.

In addition to managing Nsure Audit objects, the Nsure Audit plug-in module allows you to create and run queries in iManager. Using the Query Options and Queries tasks under the Auditing and Logging Role, you can perform the following tasks:

- "Defining Your Query Databases in iManager" on page 107
- "Managing Product Events in iManager" on page 110
- "Setting Your Global Options in iManager" on page 119
- "Defining Queries in iManager" on page 120
- "Running Queries in iManager" on page 126
- "Verifying Event Authenticity in iManager" on page 127
- "Exporting Query Results in iManager" on page 131
- "Printing Query Results in iManager" on page 131

9.1.1 Defining Your Query Databases in iManager

Before you can query a database, iManager needs to know where to find the data store and how to communicate with the database. This information is stored in the database definition. Every database you want to query must have its own database definition in iManager.

IMPORTANT: The database definitions you create in iManager are stored in the User object you use to log in to iManager. Consequently, they are not available to other users on the system.

To create a database definition in iManager:

1 Open the *Query Options* task.

1a Click the *Roles and Tasks* button on the iManager toolbar.

1b In the Roles and Tasks view, expand the *Auditing and Logging* Role.

1c Click the *Query Options* task.

2 In the *Query Options* page, click *Databases* > *New*.

3 In the *New Database Definition* menu, specify the database information.

The following table provides a description of each field in the Database menu.

4 When finished, click *OK*.

The new database definition now appears in the *Database Name* list.

The following table provides a description of each field in the database definition.

Table 9-1 *Database Definition Menu Fields*

Field	**Description**
Name	The name you want to use to refer to this database. This name appears in the Database Name list.
JDBC Class	Package and name of the Java Class providing JDBC connectivity. The JDBC drivers for the Nsure Audit supported data stores are available at the following sites: • MySQL: `MySQL Connector/J` The MySQL JDBC `.jar` file can be downloaded from the MySQL Development site (http://dev.mysql.com/downloads/). • Oracle: Oracle Instant Client The Oracle JDBC `.jar` file can be downloaded from the Oracle Web site (http://www.oracle.com/technology/tech/oci/instantclient/instantclient.html). • Microsoft SQL Server: Microsoft SQL Server Driver for JDBC The SQL Server JDBC `.jar` file can be downloaded from the Microsoft Download Center (http://www.microsoft.com/downloads/). • QL Server and Sybase JDBC Driver: jTDS (S) The QL Server and Sybase JDBC `.jar` file can be downloaded from SourceForge.net (http://www.sourceforge.net/) site.
JDBC Class continued	You must copy the JDBC drivers for your data store to the following Nsure Audit Java classpath or a subdirectory thereof: • **Windows, Linux, and Solaris:** `*Novell_Audit_install_directory*\java\logdriver\` • **NetWare:** `*Novell_Audit_install_directory*\` Additionally, if you are going to query a JDBC data store in iManager, copy all required JDBC drivers (`*.jar`) to the following iManager classpaths on your iManager server: • **NetWare:** `sys:\tomcat\4\common\lib` • **Linux and Solaris:** /var/opt/novell/tomcat4/common/lib • **Windows:** `\program files\novell\tomcat\common\lib`

Field	Description
JDBC URL	A valid JDBC URL, including the database name, that iManager uses to communicate with the database. Any JDBC-compliant driver can be used.The driver name is case sensitive. Consult the documentation provided by your database vendor for specifics on constructing JDBC URLs. The following are JDBC URL examples for the most common databases using the default port. This database name must be replaced with the name of your Nsure Audit database, the default Nsure Audit database name is naudit. • **MySQL:** jdbc:mysql://*ip_address*:*port* / *database_name* • **Oracle:** jdbc:oracle:thin:@*ip_address*:*port*:*sid* • **Microsoft SQL Server:** jdbc:microsoft:sqlserver://*ip_address*:*port*; DatabaseName=*database_name*
Table	The name of the table iManager queries. In Nsure Audit, table names are defined in MySQL and Oracle Channel objects. The default table name is log. If you have multiple MySQL or Oracle Channel objects, you must create a separate database definition for each data store.
Username	The user name iManager uses to authenticate with the database. The default username for the NetWare 6.5 data store is auditusr. (This default can be changed during the installation of Nsure Audit.) This account has all privileges to the default database (naudit) and can log in from any IP address. By default, MySQL installs in Secure Mode on NetWare 6.5. In Secure Mode, the default MySQL administrative account, Root, has rights to log in only at the database server. Therefore, if MySQL is running in Secure Mode and you want iManager to use the Root account to log in to the database, MySQL and the iManager Web server must be located on the same server and you must specify a loopback address (127.0.0.1 or localhost) in the Host field.
Password	The password the logging server uses to authenticate with the database. The default password for the NetWare 6.5 data store is auditpwd. (This default can be changed during the installation of Nsure Audit.) **IMPORTANT:** If you do not specify a different default password during the installation, new databases can be created and accessed using the default username and password. To prevent this, specify a different default password during the install.
Store Password	Stores the password the logging server uses to authenticate with the database. This enables iManager to automatically log in to the database. If you do not select the *Store Password* option, you must specify the password each time you run a query on the current database.

Editing and Deleting Database Definitions

After you have created a database definition, you can edit or delete the definition by selecting the database, then clicking *Edit* or *Delete*.

NOTE: Deleting a database definition does not affect the actual database. It only removes the database from the Query Options task's *Database* list, which means that iManager can no longer query the database.

9.1.2 Managing Product Events in iManager

The Product Events page displays the events associated with each logging application's log schema (LSC) file.

NOTE: The log schema (LSC) file catalogs the events that can be logged for a given application. It also provides the event descriptions and field labels that iManager uses in its query results. For more information, see Section A.4, "Log Schema Files," on page 187.

From the Product Events page, you can import new or custom LSC files. You can also view, add, modify, or delete events within existing LSC files. The following sections review these processes in more detail:

- "Importing Log Schemas" on page 138
- "Viewing Product Events" on page 111
- "Adding Product Events" on page 112
- "Modifying Product Events" on page 114
- "Using the Argument Builder to Define Event Schema" on page 115
- "Deleting Product Events" on page 118

Importing Log Schemas

The LSC files for the logging application instrumentations installed with Nsure Audit (such as NetWare, eDirectory, and Identity Manager) automatically display in the Product Events page. However, if you add a new logging application to Nsure Audit or localize an existing application's LSC file, you can import those log schemas into iManager. When you import an LSC file into iManager, you can then view or modify the events defined in the LSC file, add new events, or use the events to define queries, Notification filters, or Heartbeat Notifications.

NOTE: For more information on defining queries in iManager, see Section 9.1.4, "Defining Queries in iManager," on page 120. For information on defining Notification Filters, see Section 8.3, "Notification Filters," on page 102. For information on defining Heartbeat Notifications, see Section 8.4, "Heartbeat Objects," on page 104.

To import a logging application's log schema in iManager:

1 Open the *Query Options* task.

1a Click the *Roles and Tasks* button on the iManager toolbar.

1b In the Roles and Tasks view, expand the *Auditing and Logging* Role.

1c Click the *Query Options* task.

2 In the Query Options page, click *Product Events*.

3 Provide the distinguished name (DN) of the Secure Logging Server.

If you have multiple logging servers, specify the distinguished name of the logging server that loads the Application object configuration at startup. For an explanation, see Chapter 6, “Managing Applications that Log to Nsure Audit,” on page 65.

3a Click the *Object Selector* button to locate the object in the directory tree.

To move up or down in the tree, click the navigation arrows. You can also search the tree by specifying the object name and context in the Search tab.

NOTE: iManager only links valid entries.

3b Click the *Object History* button to see a list of Logging Server objects that have been selected during this iManager session.

4 From the *Select Language* drop-down list, select which language version of the log schema you want to import.

If an application does not have a log schema for the selected language, iManager imports nothing.

5 Click *Update*.

When you click *Update*, iManager locates the Logging Server object in eDirectory, scans the logging server’s supported Application containers, and imports the log schemas from the Application objects in those containers. For more information, see Section 6.3, “Application Objects,” on page 67.

The logging applications and their associated events now appear in the *Product Name* list.

Viewing Product Events

From the Product Events page, you can view the events defined in each logging application’s log schema.

To view a logging application’s associated events:

1 Open the *Query Options* task.

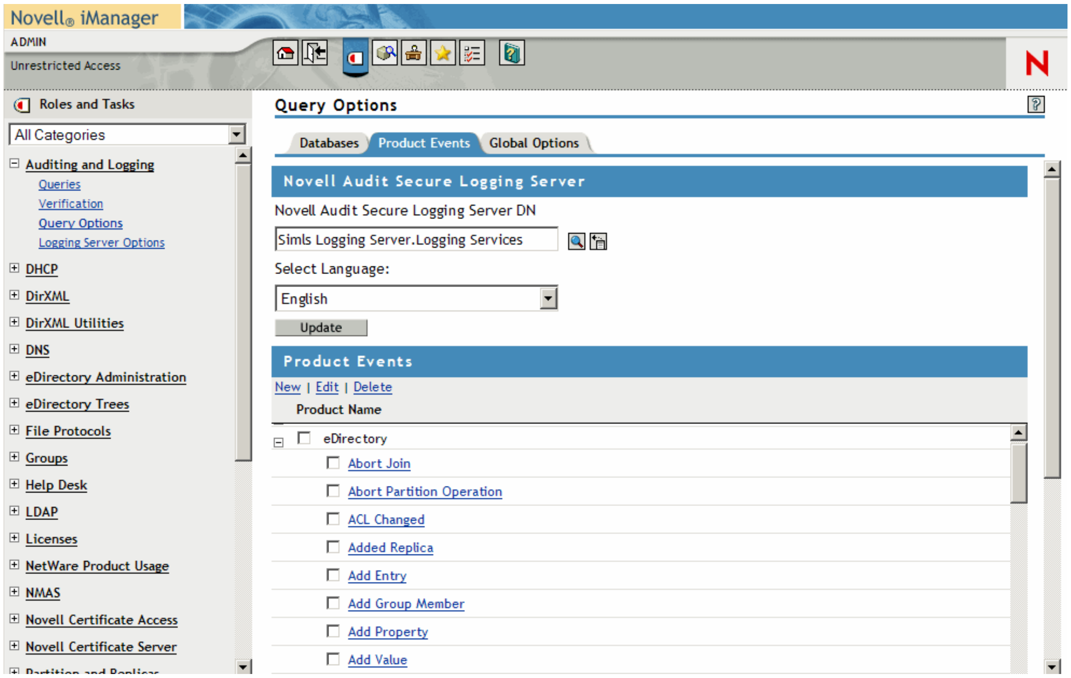
1a Click the *Roles and Tasks* button on the iManager toolbar.

1b In the Roles and Tasks view, expand the *Auditing and Logging* Role.

1c Click the *Query Options* task.

2 In the Query Options page, click *Product Events*.

3 In the Product Events page, click the plus icon ⊞ next to the product name to display the application’s log events.



Novell Audit Secure Logging Server

Novell Audit Secure Logging Server DN: Simls Logging Server.Logging Services

Select Language: English

Product Events

Product Name

⊟ □ eDirectory
- □ Abort Join
- □ Abort Partition Operation
- □ ACL Changed
- □ Added Replica
- □ Add Entry
- □ Add Group Member
- □ Add Property
- □ Add Value

NOTE: Only those events defined in the application’s LSC file appear in the Product Events page.

---

4 Select an event to view the Event ID, description, and field definitions.

For more information on event fields, see Section A.1, “Event Structure,” on page 175.

Adding Product Events

To add an event to an application’s log schema:

1 Open the *Query Options* task.

1a Click the *Roles and Tasks* button on the iManager toolbar.

1b In the Roles and Tasks view, expand the *Auditing and Logging* Role.

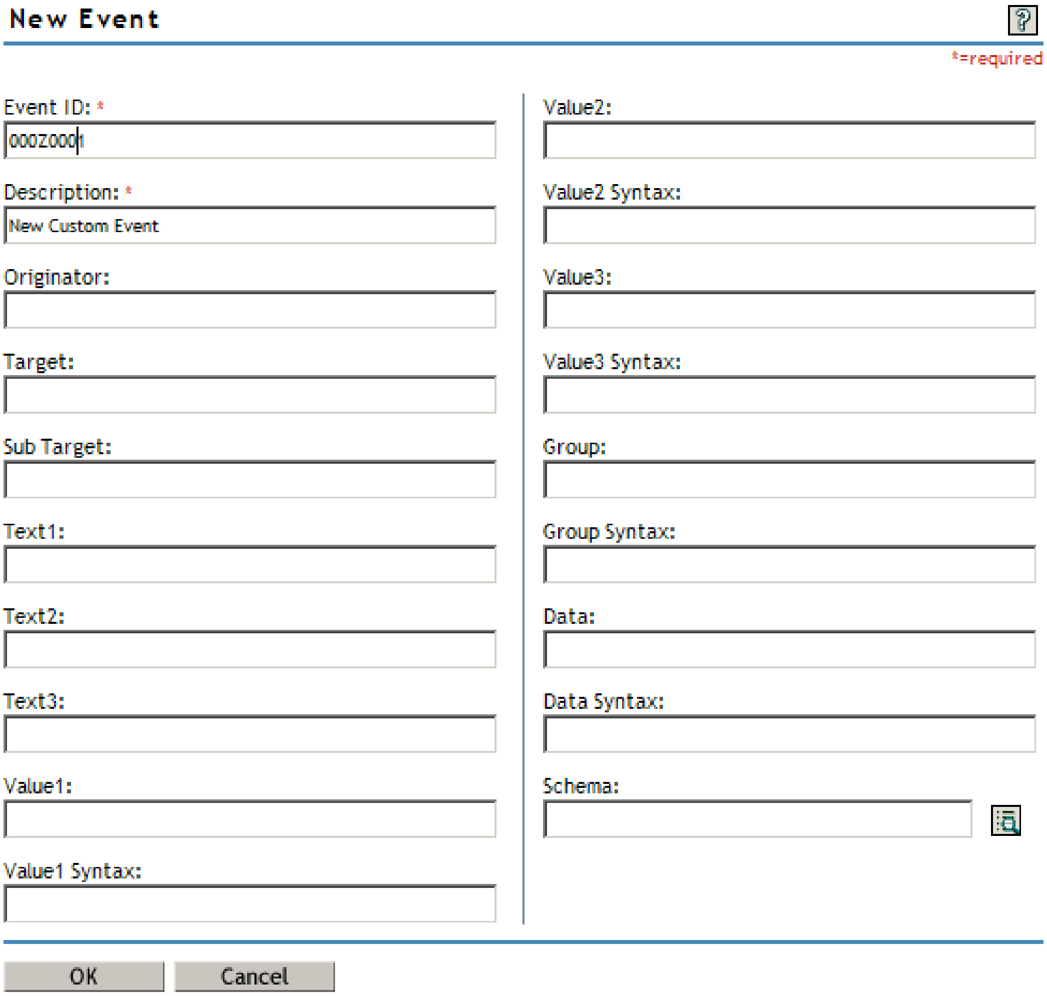
1c Click the *Query Options* task.

2 In the Query Options page, click *Product Events*.

3 In the *Product Events* page, select the logging application to which you want to add an event, then click *New*.

4 Click *OK* to confirm you want to create a new event.

5 In the *New Event* menu, specify the information for each event field.



Event ID: * 000Z0001

Description: * New Custom Event

IMPORTANT: The *Event ID* and *Description* fields are required.

The *Description* field can contain any text string up to 255 characters. The event description is stored in eDirectory in the NAuditSchema*language* attribute is the logging application's associated Application object.

The EventID is comprised of two elements: the HiWord and the LoWord.

•The HiWord is the four-digit hex value assigned to the current application. All Application IDs are assigned through Novell Developer Support and are maintained in the Nsure Audit central registry. Before instrumenting a new application, developers should obtain an AppID through Novell Developer Support (http://developer.novell.com/devres/ss/resource.htm).

•The LoWord is the AppEventID assigned by the person instrumenting the application. Typically, these values are assigned in ascending order.

For more information, see the Nsure Audit SDK (http://developer.novell.com/ndk/naudit.htm).

For an explanation of all the event fields, see Section A.1, "Event Structure," on page 175.

6. To define the event schema, click the Argument Builder button.

   The event schema determines what event fields are reported and how the event field data is displayed when logging to the File or Syslog channel in Translated Mode.

   For information on using the Argument Builder to define the event schema, see "Using the Argument Builder to Define Event Schema" on page 115.

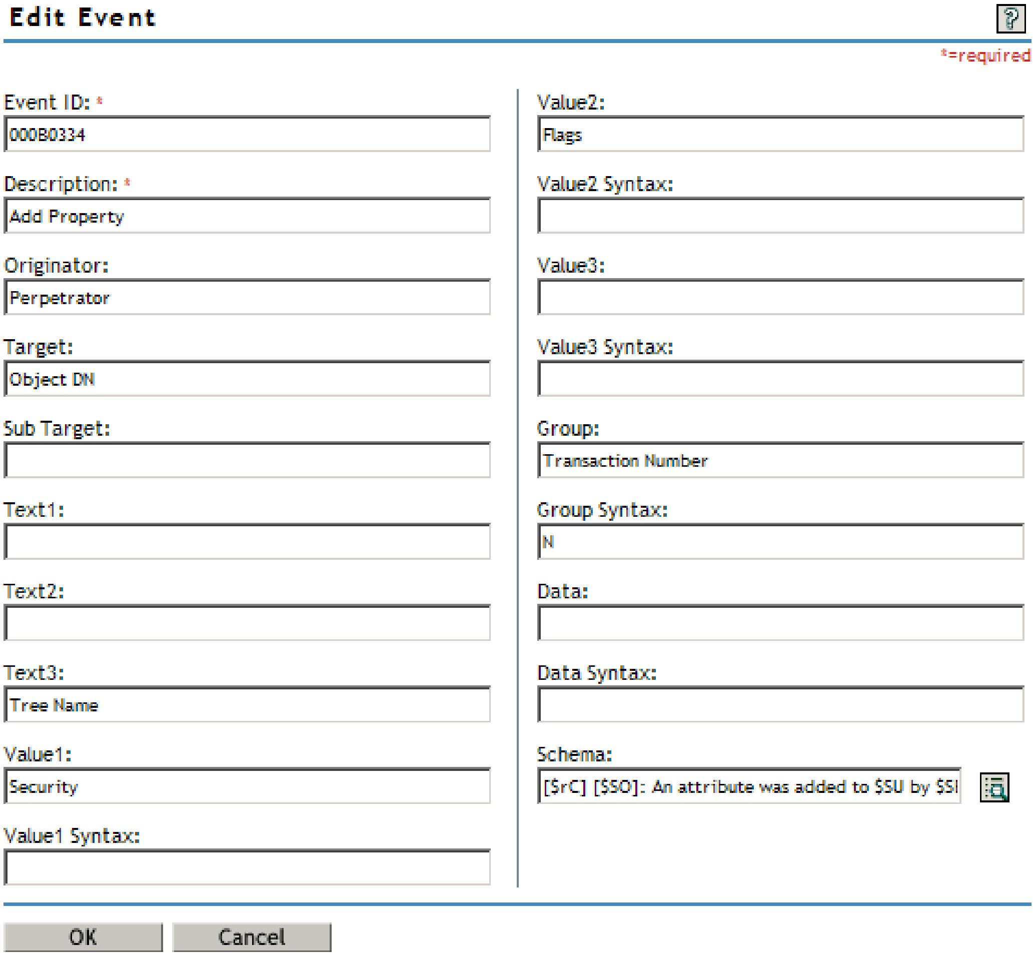
Modifying Product Events

To modify an existing event in an application's log schema:

1. Open the *Query Options* task.
   - **1a** Click the *Roles and Tasks* button on the iManager toolbar.
   - **1b** In the Roles and Tasks view, expand the *Auditing and Logging* Role.
   - **1c** Click the *Query Options* task.
2. In the Query Options page, click *Product Events*.
3. In the Product Events page, click the plus icon ⊞ next to the product name to display the application's log events.

   NOTE: Only those events defined in the application's LSC file appear in the *Product Events* page.
4. Select the event you want to modify, then click *Edit*.

5 In the *Edit Event* menu, modify the event fields.



Edit Event

*=required

Event ID: * 000B0334

Description: * Add Property

Originator: Perpetrator

Target: Object DN

Sub Target:

Text1:

Text2:

Text3: Tree Name

Value1: Security

Value1 Syntax:

Value2: Flags

Value2 Syntax:

Value3:

Value3 Syntax:

Group: Transaction Number

Group Syntax: N

Data:

Data Syntax:

Schema: [$rC] [$SO]: An attribute was added to $SU by $SI

For an explanation of event fields, see "Event Structure" on page 175.

6 To modify the event schema, click the *Argument Builder* button ▤.

The event schema determines what event fields are reported and how the event field data is displayed when logging to the File or Syslog channel in Translated Mode.

For information on using the Argument Builder to modify the event schema, see "Using the Argument Builder to Define Event Schema" on page 115.

Using the Argument Builder to Define Event Schema

The Argument Builder is a tool that simplifies the process of defining the event schema. The event schema determines what event fields are reported and how the event field data is displayed when logging to the File or Syslog channel in Translated Mode.

The Argument Builder provides a graphical interface from which you can select which event fields you want to display in the translated log file and how you want the field data to display. Based on your selections, the Argument Builder defines the event schema using a series of event field and format variables. For information on the event schema syntax, see Section A.3, "Managing Event Data," on page 181.

To define an event’s schema:

1 Open the *Query Options* task.

1a Click the *Roles and Tasks* button on the iManager toolbar.

1b In the Roles and Tasks view, expand the *Auditing and Logging* Role.

1c Click the *Query Options* task.

2 In the Query Options page, click *Product Events*.

3 Open the event menu:

- In the Product Events page, select the logging application to which you want to add an event, click *New*, then click *OK* to confirm you want to create a new event.
- Click the plus icon ⊞ next to the product name to display the application’s log events, select the event you want to modify, then click *Edit*.

4 In the event menu, click the *Argument Builder* button to open the Argument Builder.

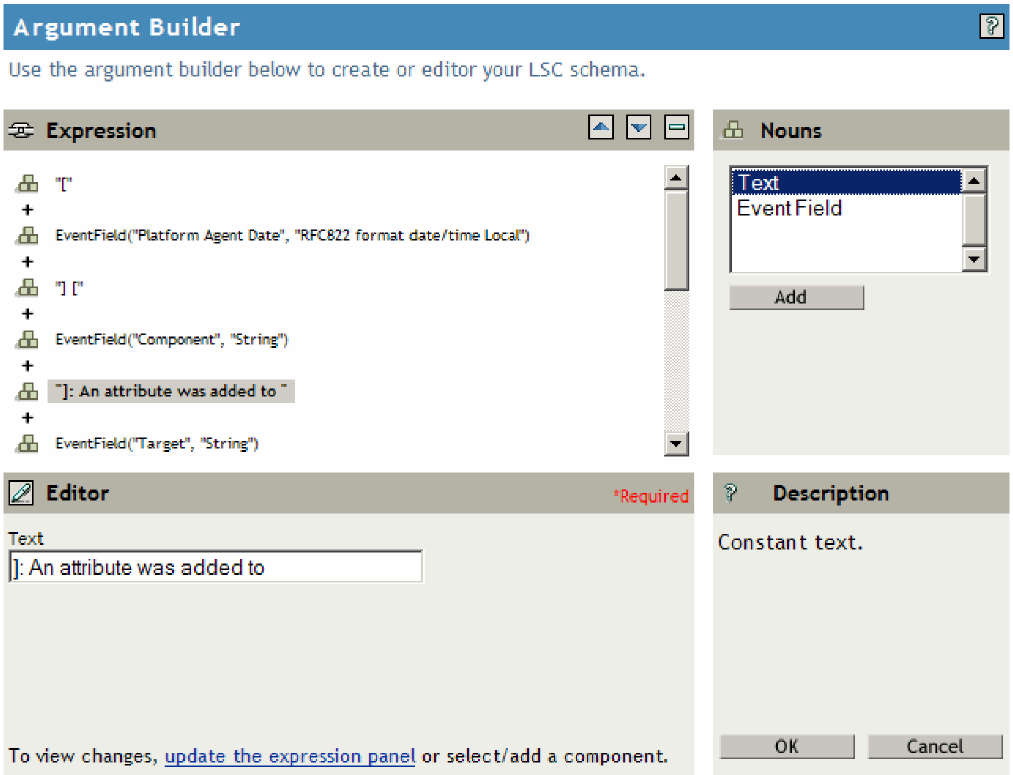
5 To add a text field to the event schema:

5a In the *Noun* frame, select *Text*, then click *Add*.

5b In the *Editor* frame, specify the text string in the *Text* field.

5c In the *Noun* frame, click *Add*.

The new text field appears in the *Expression* frame.



Argument Builder

Use the argument builder below to create or editor your LSC schema.

Expression

"["
+
EventField("Platform Agent Date", "RFC822 format date/time Local")
+
"] ["
+
EventField("Component", "String")
+
"]: An attribute was added to "
+
EventField("Target", "String")

Editor *Required

Text:]: An attribute was added to

To view changes, update the expression panel or select/add a component.

Description

Constant text.

6 To add an event field to the event schema:

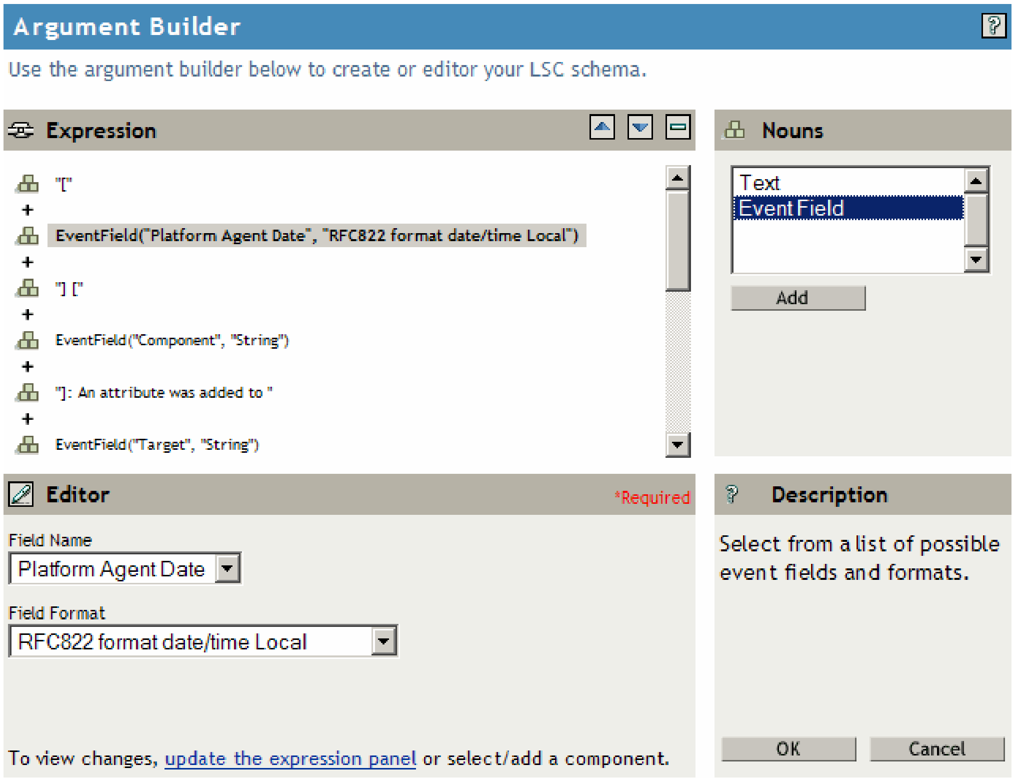
6a In the *Noun* frame, select *Event Field*, then click *Add*.

6b In the *Editor* frame, select an event field from the *Field Name* drop-down list.

6c Select the event field’s associated format from the *Field Format* drop-down list.

6d In the *Noun* frame, click *Add*.

The new event field appears in the *Expression* frame.



Use the argument builder below to create or editor your LSC schema.

```
"["
+
EventField("Platform Agent Date", "RFC822 format date/time Local")
+
"] ["
+
EventField("Component", "String")
+
"]: An attribute was added to "
+
EventField("Target", "String")
```

Field Name: Platform Agent Date

Field Format: RFC822 format date/time Local

To view changes, update the expression panel or select/add a component.

Select from a list of possible event fields and formats.

7 To remove an item from the event schema:

7a In the *Expression* frame, select the text or event field you want to remove.

7b Click the *Remove Token* button ⊟ in the *Expression* frame.

The text or event field is removed from the *Expression* frame.

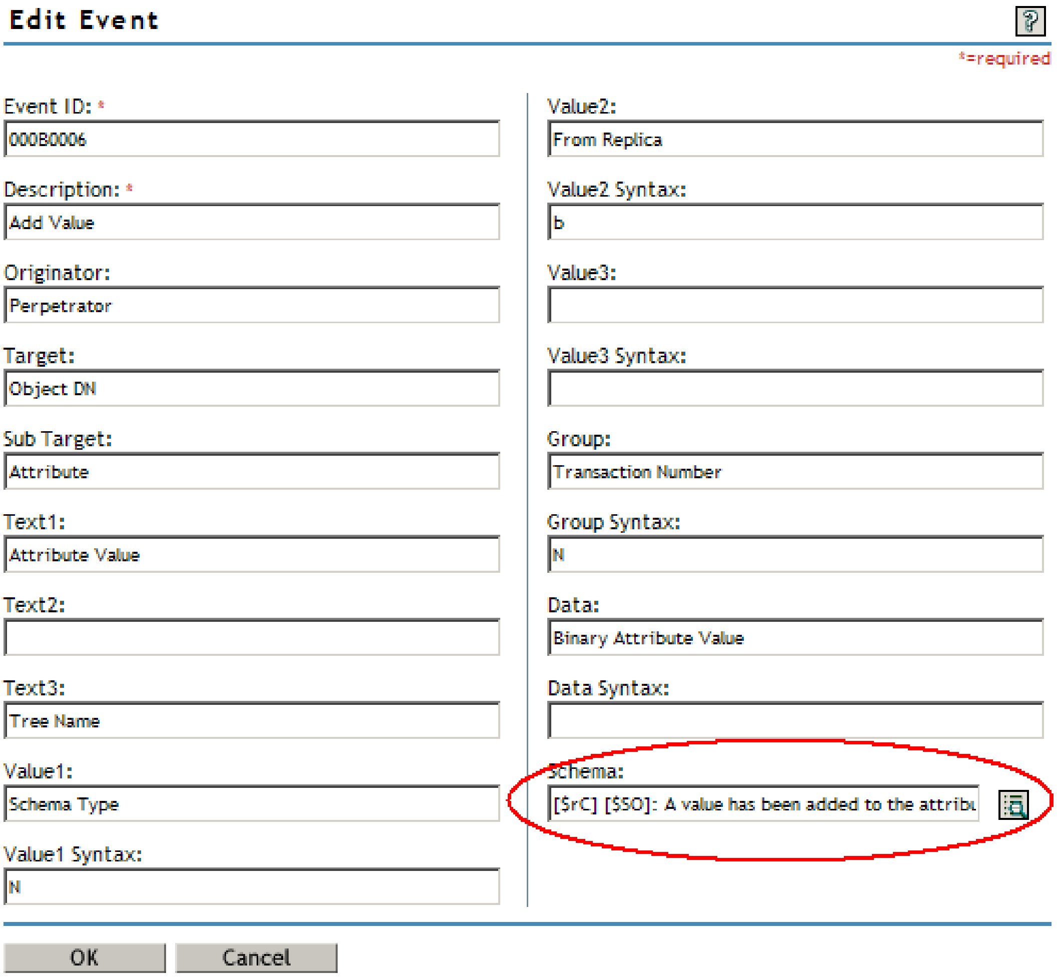
8 To modify the item order in the event schema:

8a In the *Expression* frame, select the text or event field you want to move.

8b Click the *Up* or *Down* ▼ buttons in the *Expression* frame to modify the item order.

9 When you have completed the event schema definition, click *OK* to save your changes.

iManager returns you to the event menu.



Edit Event

*=required

Event ID: * 000B0006

Description: * Add Value

Originator: Perpetrator

Target: Object DN

Sub Target: Attribute

Text1: Attribute Value

Text2:

Text3: Tree Name

Value1: Schema Type

Value1 Syntax: N

Value2: From Replica

Value2 Syntax: b

Value3:

Value3 Syntax:

Group: Transaction Number

Group Syntax: N

Data: Binary Attribute Value

Data Syntax:

Schema: [$rC] [$SO]: A value has been added to the attribu

The defined event schema appears in the *Schema* field as a series of event field and format variables. For information on the event schema syntax, see Section A.3, "Managing Event Data," on page 181.

Deleting Product Events

To delete an event from an application's log schema:

1 Open the *Query Options* task.

1a Click the *Roles and Tasks* button on the iManager toolbar.

1b In the Roles and Tasks view, expand the *Auditing and Logging* Role.

1c Click the *Query Options* task.

2 In the Query Options page, click *Product Events*.

3 In the Product Events page, click the plus icon ⊞ next to the product name to display the application's log events.

NOTE: Only those events defined in the application's LSC file appear in the Product Events page.

4 Select the event you want to modify, then click *Delete*.

5 Click *OK* to confirm you want to delete the event.

The event is removed from the LSC file.

9.1.3 Setting Your Global Options in iManager

The Global Options page allows you to set a default limit on the number of rows returned in the query results.

To set a default query limit in iManager:

1 Open the *Query Options* task.

  1a Click the *Roles and Tasks* button on the iManager toolbar.

  1b In the Roles and Tasks view, expand the *Auditing and Logging* Role.

  1c Click the *Query Options* task.

2 In the Query Options page, click *Global Options*.

3 In the Global Options page, specify your default query limit.

4 When finished, click *OK*.

The Global Options page also includes the default sort order and date time format; however, these options cannot be modified in the current release.

The following table provides an explanation of each option in the Global Options page.

IMPORTANT: These are global settings. This means that iManager automatically adds these parameters to all database queries unless the parameter is expressly defined in the query statement.

Table 9-2 *Query Global Options*

Option	Description
Limit Query Result To	This option limits the number of rows (that is, records) that are returned from a database query.
Default Sort Order	iManager does not have a default sort order. Consequently, This option cannot be modified in the current release. You can sort the records in the query results page by clicking a column heading.
Date/Time Format	iManager formats all time and date information in RFC-822 UTC format. RFC-822 is the Internet standard format for electronic mail message headers. All time and date values are expressed in UTC rather than local time. This option cannot be modified in the current release.
Import existing reporting configuration	This option allows you to import queries from other iManager servers. The queries must be defined in XML format.

Option	Description
Export current reporting configuration and save	This option exports the queries defined on the current imanager server. The queries are exported in XML format.

9.1.4 Defining Queries in iManager

iManager uses queries to request information from MySQL and Oracle databases. All queries are defined in SQL. Although you must be familiar with the SQL language to create SQL query statements, this is the most powerful and flexible query method.

iManager includes several predefined queries and it includes a Query Builder to help you define basic query statements. Of course, you can also build your own query statements.

You can create two kinds of queries in iManager: manual queries and saved queries. Manual queries are simply queries that are not saved; they only run one time. Saved queries are saved in the Query list and can be run again and again against different databases.

IMPORTANT: Saved queries are stored in the User object you use to log in to iManager. Therefore, they are not available to other users on the system.

The following sections provide the information you need to perform the following tasks:

- Use Predefined Queries
- Create Manual Queries
- Create Saved Queries
- Create Saved Queries Using the Query Builder
- Modify and Delete Saved Queries

Using Predefined Queries

iManager includes several predefined queries. You can modify these queries or run them “as is.”

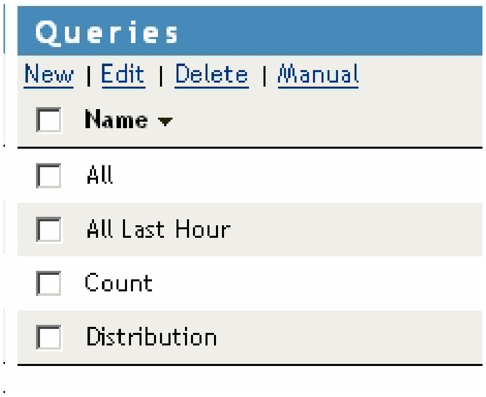
1 Open the *Queries* task.

1a Click the *Roles and Tasks* button on the iManager toolbar.

1b In the Roles and Tasks view, expand the *Auditing and Logging* Role.

1c Click the *Queries* task.

2 Select the predefined queries from the *Query* list.



□ Name

□ All

□ All Last Hour

□ Count

□ Distribution

The following table lists the queries that ship with iManager and their functions.

Table 9-3 *iManager Predefined Queries*

Query	Function
All	Returns all events in the current data store.
All last hour	Returns all events that occurred within the last hour.
Count	Returns the total number of events logged to the current data store.
Distribution	Returns the number of times each Event ID has occurred in the current data store.

Creating Manual Queries

Manual queries are simply queries that are not saved; they only run one time.

To create a manual query in iManager:

1 Open the *Queries* task.

   1a Click the *Roles and Tasks* button on the iManager toolbar.

   1b In the Roles and Tasks view, expand the *Auditing and Logging* Role.

   1c Click the *Queries* task.

2 In the *Database* drop-down list, select the database you want to query.

3 Click *Manual*.

4 In the *Name* field, specify the name you want to appear as the title in the query results.

5 Define the query statement in the *Query* box.

   For basic information on building SQL queries, see the *MySQL Reference Manual* (http://www.mysql.com/documentation/index.html).

   You do not need to include a FROM clause in your query statement. iManager dynamically builds the FROM clause using the table specified in the database definition you select when you run the query. However, if the query statement does include a FROM clause, iManager queries the table defined in the query statement.

6. Click *Run Query* to run the query.

iManager Query Macros

The following table contains macros that can be used when creating iManager queries:

Table 9-4 *iManager Query Macros*

Macro	Description
[TIME]= [LAST_HOUR] [TODAY] [YESTERDAY] [LAST_24_HOURS] [LAST_7_DAYS] [THIS_MONTH]	Limits results to those occurring in a specific time frame.
HexToDec[hex#]	Converts a number from hexidecimal to decimal.
IP[192.168.0.5]	Enables you to use an IP address in a query.
[TABLE]	Replaced with the actual table name during the query.

Creating Saved Queries

Saved queries are saved in the Query list and can be run again and again against different databases.

To create a saved query in iManager:

1. Open the *Queries* task.

   1a. Click the *Roles and Tasks* button on the iManager toolbar.

   1b. In the Roles and Tasks view, expand the *Auditing and Logging* Role.

   1c. Click the *Queries* task.

2. Click *New*.

3. In the *Name* field, specify the name you want to use to refer to this query.

   The query name appears in the *Query* list and in the query results' title.

4. In the *Select Field* drop-down list, select the field information (columns) you want to return in the query.

   Use Shift+click or Ctrl+click to select multiple fields.

5. Define the query statement.

   - Create the query using the Query Builder.

     For specific information on the Query Builder, see "Creating Saved Queries Using the Query Builder" on page 123.

     or

   - Write the query statement in the *Query SQL Statement* window.

For basic information on SQL query statements, see the *MySQL Reference Manual* (http://www.mysql.com/documentation/index.html).

You do not need to include a FROM clause in your query statement. iManager dynamically builds the FROM clause using the table specified in the database definition you select when you run the query. However, if the query statement does include a FROM clause, iManager queries the table defined in the query statement.

6 Select *Translate Column Titles* if you want to label the column headings in the query results page with the field titles defined in the log schema.

Select this option only for queries that return one type of event. If you select this option for queries that return multiple types of events, Nsure Audit Report labels the column headings with the field titles from the last event returned in the query.

IMPORTANT: For this option to work, you must import each application's log schema. For information, see "Importing Log Schemas" on page 110.

7 When finished, click *OK*.

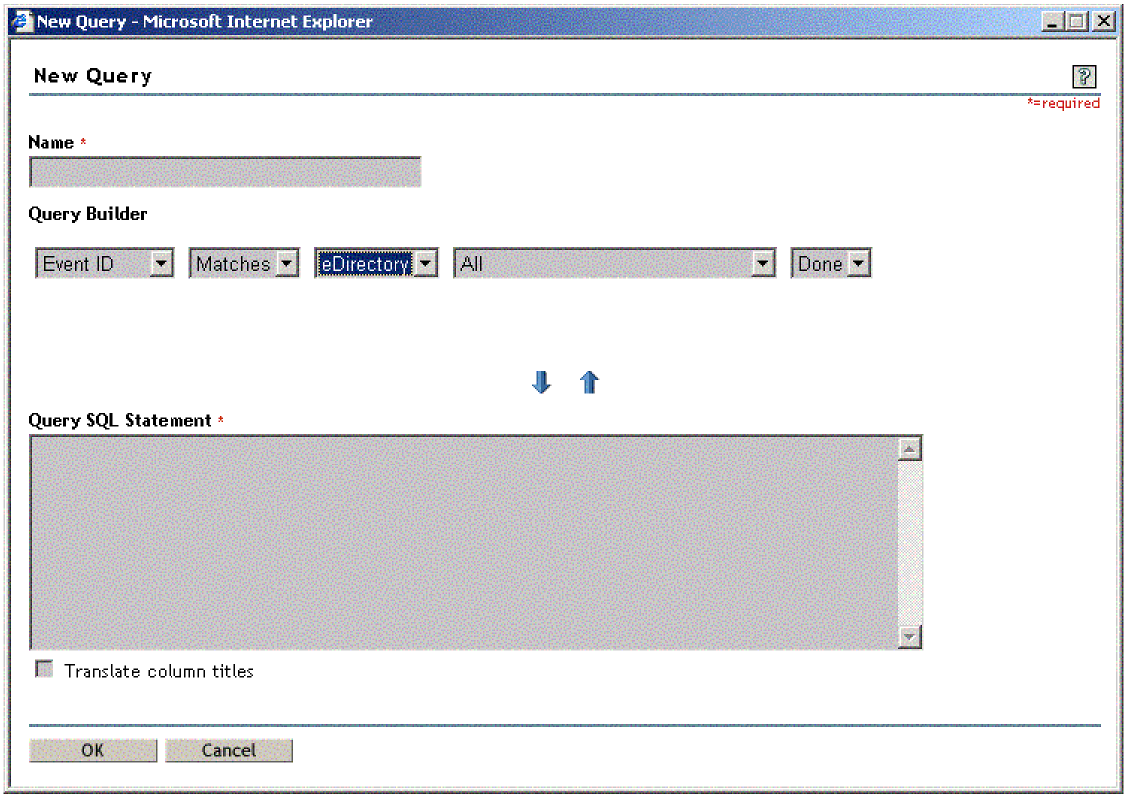
The query now appears in the *Query* list.

Creating Saved Queries Using the Query Builder

If you are unfamiliar with the SQL query language, you can use the Query Builder to help you define basic saved queries. The Query Builder simplifies the process of creating a query by allowing you to choose from lists of predefined parameters. The Query Builder then constructs the query statement from the parameters you select.

To open the Query Builder fields, select *And* in the initial drop-down list.

Figure 9-1 *Query Builder in iManager*



Because the Query Builder can provide only a limited set of parameters, the queries it creates are very simple. However, it is the easiest way to create saved queries and it is capable of creating most base-level queries.

The following table reviews the options in the Query Builder.

***Table 9-5** Query Builder Options*

Parameter	Description
Event Field	The event field you want to query. You can select the following options from the drop-down list: • *Event ID* • *Time Frame* • *Component* • *Originator* • *Originator Type* • *Target* • *Target Type* • *Sub Target* • *Text1* • *Text2* • *Text3* • *Source IP* • *Severity* • *Value1* • *Value2* • *Value3* For more information on event fields, see Section A.1, “Event Structure,” on page 175.
Condition	The condition under which the logging server applies the Value to the Event Field. Depending on the *Event Field*, you can select the following conditions from the drop-down list box: • *matches* • *less than* • *greater than* • *begins with* • *contains* • *is between* __________ *and* ___________
Product	Limits the query results to a specific logging application. This option is available only if you select the *Event ID* field.

Parameter	Description
Value	The value for the designated event field. The query statement applies the Value to the designated Event Field under the defined conditions. If an event matches the criteria, it is returned in the query results. If you select *Event ID* in the Event field and if the designated product’s log schema provides event descriptions, iManager displays the event descriptions rather than the Event IDs; however, the events are still sorted by their numeric Event ID. Therefore, the event descriptions are not listed in alphabetical order, but related events are grouped together.
Operator	To narrow the query results, you can define values for multiple event fields. Using standard and/or operators, you can define multiple event conditions. The done operator indicates the end of the query statement. The conditions are accumulative; that is, the logging server applies the first condition, then the second, then the third, etc., to progressively narrow the results.
Arrows	The down-arrow moves the query down into the Query SQL Statement box. iManager builds an SQL query statement from the parameters you define in the Query Builder. The up-arrow moves an SQL query statement from the Query SQL Statement box to the Query Builder. If the query statement includes clauses that are outside the scope of the Query Builder, iManager returns the error `SQL statement is too complex to use builder.`

Modifying and Deleting Saved Queries

After you have created a saved query, you can edit or delete the query by selecting the query name and clicking *Edit* or *Delete*.

9.1.5 Running Queries in iManager

1 Open the *Queries* task.

1a Click the *Roles and Tasks* button on the iManager toolbar.

1b In the Roles and Tasks view, expand the *Auditing and Logging* Role.

1c Click the *Queries* task.

2 In the *Database* drop-down list, select the database you want to query.

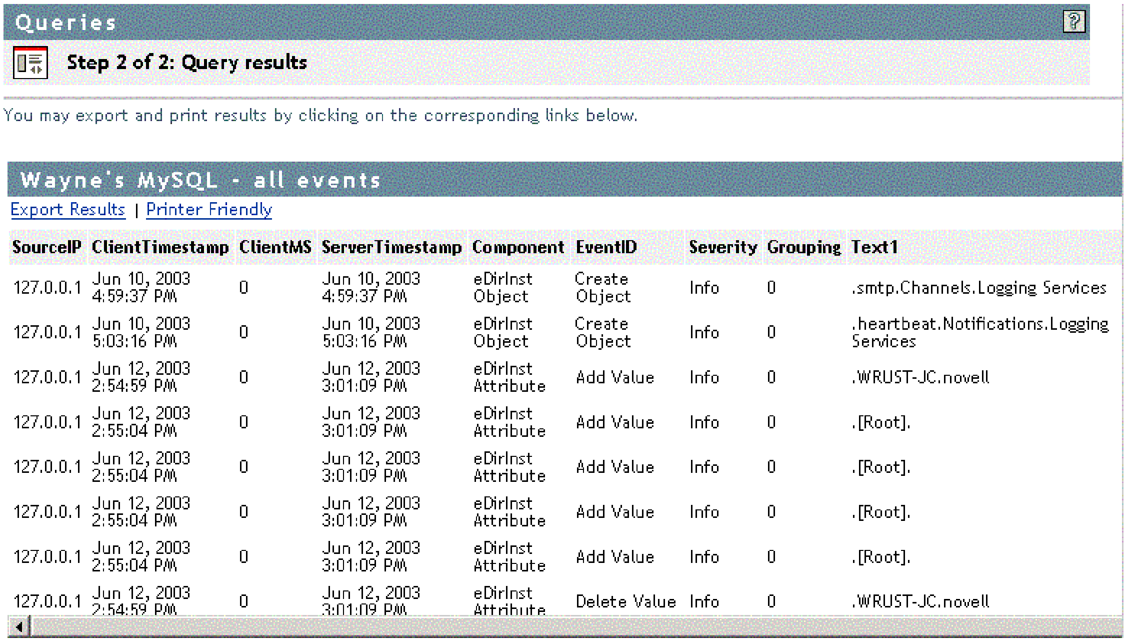
For information on creating Database definitions, see “Defining Your Query Databases in iManager” on page 107.

3 In the *Queries* list, select the query you want to run.

4 Click *Run Query*.

iManager returns the query results in a data table; rows represent individual records and columns represent fields within those records. You can click any of the column headings to sort the results by that field.

Figure 9-2 *iManager Query Results*



Queries

Step 2 of 2: Query results

You may export and print results by clicking on the corresponding links below.

Wayne's MySQL - all events

Export Results | Printer Friendly

SourceIP	ClientTimestamp	ClientMS	ServerTimestamp	Component	EventID	Severity	Grouping	Text1
127.0.0.1	Jun 10, 2003 4:59:37 PM	0	Jun 10, 2003 4:59:37 PM	eDirInst Object	Create Object	Info	0	.smtp.Channels.Logging Services
127.0.0.1	Jun 10, 2003 5:03:16 PM	0	Jun 10, 2003 5:03:16 PM	eDirInst Object	Create Object	Info	0	.heartbeat.Notifications.Logging Services
127.0.0.1	Jun 12, 2003 2:54:59 PM	0	Jun 12, 2003 3:01:09 PM	eDirInst Attribute	Add Value	Info	0	.WRUST-JC.novell
127.0.0.1	Jun 12, 2003 2:55:04 PM	0	Jun 12, 2003 3:01:09 PM	eDirInst Attribute	Add Value	Info	0	.[Root].
127.0.0.1	Jun 12, 2003 2:55:04 PM	0	Jun 12, 2003 3:01:09 PM	eDirInst Attribute	Add Value	Info	0	.[Root].
127.0.0.1	Jun 12, 2003 2:55:04 PM	0	Jun 12, 2003 3:01:09 PM	eDirInst Attribute	Add Value	Info	0	.[Root].
127.0.0.1	Jun 12, 2003 2:55:04 PM	0	Jun 12, 2003 3:01:09 PM	eDirInst Attribute	Add Value	Info	0	.[Root].
127.0.0.1	Jun 12, 2003 2:54:59 PM	0	Jun 12, 2003 3:01:09 PM	eDirInst Attribute	Delete Value	Info	0	.WRUST-JC.novell

If you selected the *Translate Column Titles* option when you defined the query, iManager labels the query results with the field titles defined in the log schema. iManager also displays each event's field titles as you mouse over the event fields.

NOTE: It is recommended that you only select the *Translate Column Titles* option for queries that return one type of event. If you select this option for queries that return multiple types of events, Nsure Audit Report labels the column headings with the field titles from the last event returned in the query. For more information on the *Translate Column Titles* option, see "Creating Saved Queries" on page 122.

9.1.6 Verifying Event Authenticity in iManager

To provide non-repudiable logs, Nsure Audit can digitally sign each event that is logged to the data store. To sign an event, the logging application or the Platform Agent hashes the event data and signs the hash with the Logging Application's private key. The signature is then stored as part of the event. This signature allows the auditor or investigator to determine if an event has been changed.

To allow auditors to determine if an event has been deleted or the sequence of events has been changed, Nsure Audit can also chain its event signatures. That is, if event chaining is enabled, each event's signature includes its own data as well as the signature from the previous event.

Event chaining is enabled in the Platform Agent's configuration file, `logevent`. For information on configuring this option, see "Logevent" on page 42. It can also be configured through the Secure Logging Server object's *Sign Event* attribute. For more information, see Section 4.2.2, "Logging Server Objects," on page 45.

If event chaining is enabled, iManager can verify that all the events logged to the data store for each logging application are authentic; that is, it can validate the event signatures to determine if an application's events have been tampered with, deleted, or if the sequence of events has been changed.

The following sections review how to define a verification query and how to verify logged events:

- "Verifying Logged Events" on page 128
- "Defining Verification Queries" on page 129

Verifying Logged Events

To verify that an application's logged events are authentic:

1. Open the *Queries* task.
   - **1a** Click the *Roles and Tasks* button on the iManager toolbar.
   - **1b** In the Roles and Tasks view, expand the *Auditing and Logging* Role.
   - **1c** Click the *Verification* task.
2. In the *Database* drop-down list, select the database where the events you want to verify are logged.
3. Select the Logging Application for which you want to verify events.

   NOTE: Nsure Audit provides a pre-defined verification query for Nsure Audit events. For all other logging applications, you must define your own verification query. For more information, see "Defining Verification Queries" on page 129.
4. Click *Verify*.

After Nsure Audit Report verifies the application's events, it returns the verification results.

If the event chain is authentic, iManager returns a message that the table's events have been verified as authentic.If the event chain is not authentic, iManager lists each problem and its associated event.

There following table provides an explanation for each signature error.

Table 9-6 *Signature Errors*

Signature Error	Explanation
Logging application restarted. This is the first event after the restart, but it cannot be verified if events have been removed at the end of the previous chain.	The logging application shut down and restarted, so the event count field (ClientMS) started again at 0; therefore, the event chain was broken. You can determine if the application restart was malicious or not by looking at the last event in the previous event chain. Logging applications send an event when they are unloaded, so if the last event in the previous event chain is an application unload event, you know that no events have been deleted.

Signature Error	Explanation
This is the first event in the database, but not the first event in the chain. Earlier events are missing.	The current event is the first event in the database, however, the event count field (ClientMS) indicates this is not the first event in the chain. This message occurs if you have rolled or expired your data store. You can use the following methods to determine if any events are missing: • If you expired your data store, you can look at the current event's time stamp to see if it occurred at the time you expired the data store. • If you rolled the data store, you can look at the event count field for the last event in the archived data store to determine if it preceded the current event.
The previous event is missing.	The current event's signature does not include the signature from the previous event. Using the event count field (*ClientMS*), Nsure Audit Report can determine that only the previous event is missing.
x previous events are missing.	The current event's signature does not include the signature from the previous event. Using the event count field (ClientMS), Nsure Audit Report can determine approximately how many previous events are missing.
Event has been tampered with.	The current event's signature is not valid. Although it includes the signature from the previous event, the event data in the signature does not match the current data.

Defining Verification Queries

To define a verification query:

1 Open the *Queries* task.

1a Click the *Roles and Tasks* button on the iManager toolbar.

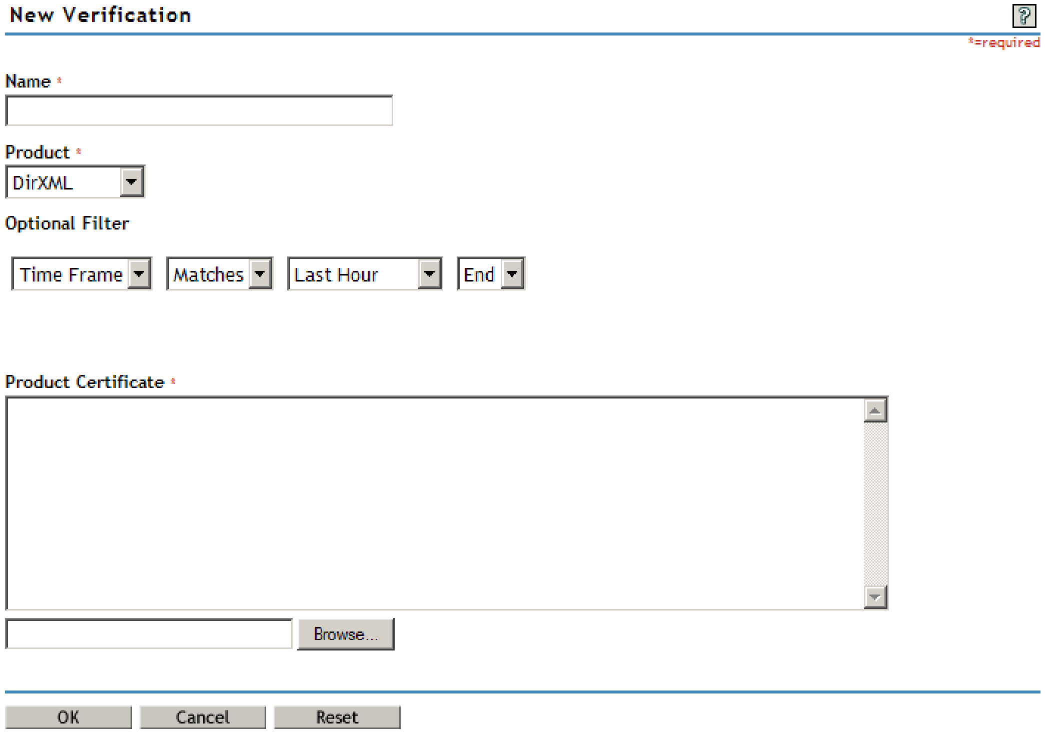
1b In the Roles and Tasks view, expand the *Auditing and Logging* Role.

1c Click the *Verification* task.

2 In the *Database* drop-down list, select the database where the events you want to verify are logged.

3 Click *New* to define the verification query.

The *New Verification* menu appears.



New Verification

*=required

Name *

Product *: DirXML

Optional Filter: Time Frame | Matches | Last Hour | End

Product Certificate *

4 In the *Name* field, specify the name you want to use to refer to this query.

The query name appears in the *Query* list and in the query results' title.

5 In the *Product* drop-down list, select the logging application for the events you want to verify.

6 To narrow the query, select *And* or *Or* in the *Optional Filter* drop-down list.

This expands the filter options so you can narrow the verification query to a specific time frame or IP address range.

Using standard and/or operators, you can define multiple event conditions. The done operator indicates the end of the query statement.

The conditions are accumulative; that is, the logging server applies the first condition, then the second, then the third, etc., to progressively narrow the results.

7 Include the Logging Application Certificate in the *Product Certificate* window.

Click *Browse* to locate the Logging Application Certificate in the directory tree.

By default, the Logging Application Certificates are available in the following directories:

- `sys:\system\naudit` (NetWare)
- `\program files\novell\nsure audit\logschema\` (Windows)
- `/opt/novell/naudit//logschema/` (Linux)
- `/opt/NOVLnaudit/logschema/` (Solaris)

8 Click *OK*.

The query now appears in the *Query* list.

9.1.7 Exporting Query Results in iManager

iManager can export query results in the following formats:

- HTML file (`*.htm`)
- Comma-Separated Text file (`*.csv`)
- Tab Delimited Text file (`*.txt`)

To export query results in iManager:

1. Run a query.

   For step-by-step instructions, see "Running Queries in iManager" on page 126.
2. Within the query results page, click *Export Results*.
3. Select the export format, then click *OK*.

   iManager brings up a *Save As* dialog box.
4. Select the directory location and specify the filename.
5. Click *Save*.

9.1.8 Printing Query Results in iManager

1. Run a query.

   For step-by-step instructions, see "Running Queries in iManager" on page 126.
2. Within the query results page, click *Printer Friendly*.

   iManager opens another page with the query results formatted in an HTML table.
3. Click *File* > *Print*.

The query results are printed to your default printer.

9.2 Using Nsure Audit Report

NOTE: Nsure Audit Report is included with NetWare 6.5 for evaluation purposes only. It displays a licensing notice and terminates after 10 minutes of use until you install a valid license.

Nsure Audit Report is a Windows-based, ODBC-compliant application that can use SQL query statements or Crystal Decisions Reports to query Oracle and MySQL data stores (or any other database that has ODBC driver support). You can define your own SQL query statements or import existing query statements and reports. Some logging applications might also include their own predefined queries or reports. Query results are returned in simple data tables; rows represent individual records and columns represent fields within those records.

This section provides the information you need to use Nsure Audit Report to generate queries and reports. It includes

- Section 9.2.1, "Installing Nsure Audit Report," on page 132
- Section 9.2.2, "Launching Nsure Audit Report," on page 132
- Section 9.2.3, "Nsure Audit Report Interface," on page 133

9.2.1 Installing Nsure Audit Report

Nsure Audit Report is available only in the Nsure Audit Windows installation. For information on the Windows install, see "Installing Nsure Audit on Windows" in the *Novell Nsure Audit Installation Guide*.

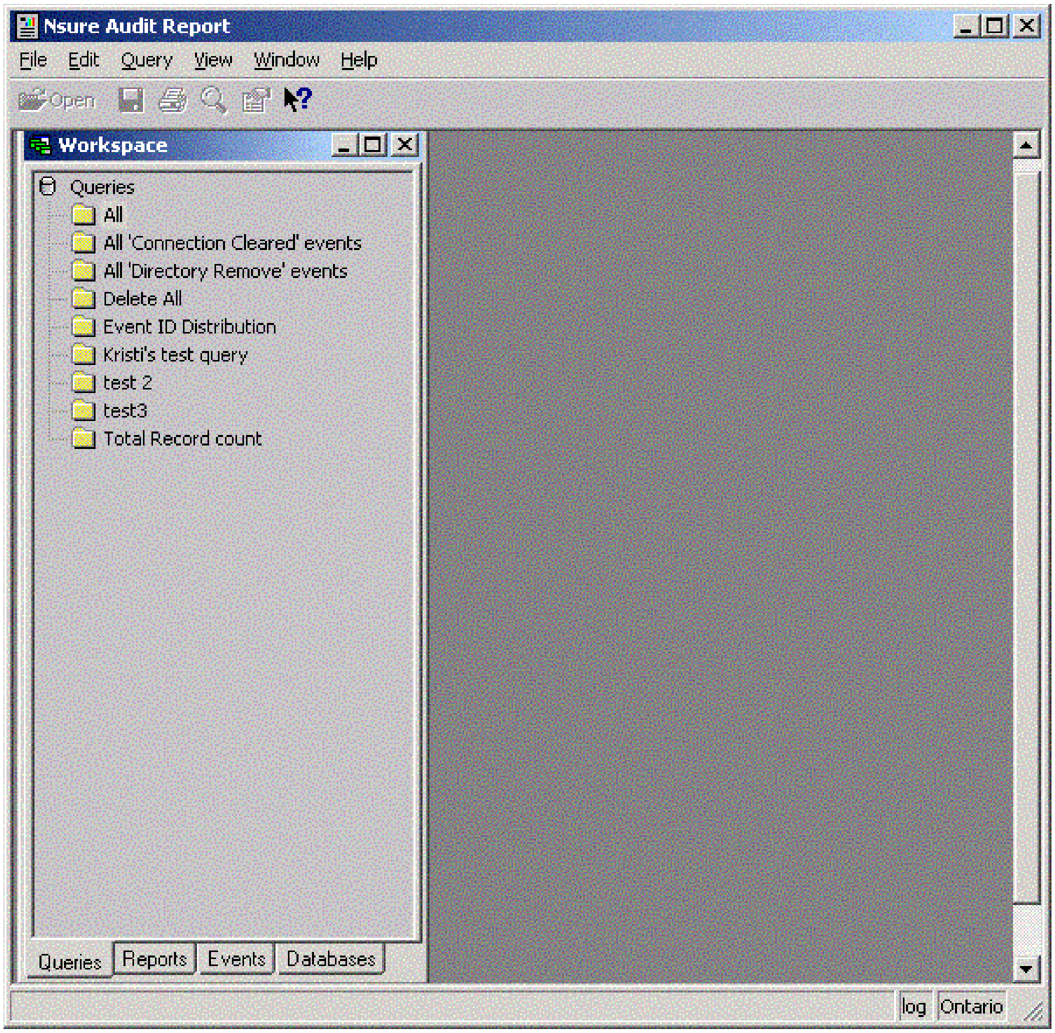
The Nsure Audit Report program file, `lreport.exe`, is installed to the `\program files\novell\nsure audit` directory.

9.2.2 Launching Nsure Audit Report

During installation, Nsure Audit Report is added to the *Start* menu. To start Nsure Audit Report from the *Start* menu, click *Start* > *Programs* > *Nsure Audit Reporting Application*.

9.2.3 Nsure Audit Report Interface

Figure 9-3 *Nsure Audit Report Interface*



The Workspace window includes four panes:

- The Queries pane lists your defined queries. You can create, delete, or run queries in this pane.
- The Reports pane lists the predefined Crystal Reports. You can import or run reports in this pane.
- The Events pane lists your logging applications and their associated events. You can run distribution reports or count an application's overall events in this pane.

  You must import the applications' log schemas before they appear in the Events pane. For more information, see "Importing Log Schemas" on page 110.
- The Databases pane lists the databases that you can query.

  You must define your databases before they appear in the Database pane. For more information, see "Defining Your Databases in Nsure Audit Report" on page 134.

To move between panes, simply click the tabs at the bottom of the Workspace window. You can also open each pane from the *View* menu.

The status bar displays the currently selected database and table. You can click these fields to select another database or table.

9.2.4 Defining Your Databases in Nsure Audit Report

Nsure Audit Report allows you to run queries and reports on any ODBC Data Source defined in your Windows registry.

To define your Windows Data Sources, go to the Windows Start menu and click *Settings* > *Control Panel* > *Administrative Tools* > *Data Sources*. There you can add, remove, and configure your ODBC Data Sources. For more information, see Windows Help.

After you have defined an ODBC Data Source in Windows, you can add the Data Source to your list of available databases in Nsure Audit Report.

1. Click *File* > *New* > *Database*.

   You can also right-click in the Database pane and select *Insert*.
2. In the Name field, specify the name you want to use to refer to this database.
3. Click *Browse*.
4. In the Select Data Source window, click *Machine Data Source*.
5. Select the Windows Data Source Name (DSN) you want to be able to query in Nsure Audit Report.
6. Specify the username and password Nsure Audit Report can use to authenticate with the database.

   If you leave this field blank, Nsure Audit Report uses the username and password defined in the Data Source.

   IMPORTANT: The only time you need to specify a username and password is if your Data Source driver does not define the username and password or if the username defined in the Data Source does not have rights to log in from the current workstation.
7. When finished, click *OK*.

The Data Source now appears in the Database pane and can be selected for queries and reports.

The following table provides further information on the options in the Define Database menu.

Table 9-7 *Define Database Menu Options*

Option	Description
Name	The name you want to use to refer to this Data Source. This name appears in the Databases pane.
DSN	The Windows Data Source Name (DSN) you want to add to your database list. For information on defining your Windows Data Sources, refer to Windows Help.

Option	Description
Username	The username Nsure Audit Report uses to authenticate with the database. If you leave this field blank, Nsure Audit Report uses the username and password defined in the Data Source. The only time you need to provide a username and password is if your Data Source driver does not define the username and password or if the username defined in the Data Source does not have rights to log in from the current workstation. In some cases, the username and password defined in the Data Source might not have rights to log in from the current workstation. For example, in Secure Mode, the default MySQL administrative account, Root, only has rights to log in at the database server. Therefore, if MySQL is running in Secure Mode, you either need to create a new user account with rights to log in from the current workstation or you must modify the rights for the Root account. (By default, MySQL installs in Secure Mode on NetWare 6.5.) The default username for the NetWare 6.5 data store is auditusr. (This default can be changed during the installation of Nsure Audit.) This account has all privileges to the default database (`naudit`) and can log in from any IP address.
Password	The password Nsure Audit Report uses to authenticate with the database. **NOTE:** The default password for the NetWare 6.5 data store is auditpwd. (This default can be changed during the installation of Nsure Audit.)
Do Not Store Password	By default, Nsure Audit Report stores the password the logging server uses to authenticate with the database. This enables Nsure Audit Report to automatically log in to the database. **IMPORTANT:** If you select the *Do Not Store Password* option, you must specify the password each time you run a query on the current database.

Editing and Deleting Data Sources

To modify or delete a Data Source in Nsure Audit Report:

1 In the Databases pane, select the Data Source you want to modify or delete.

2 Right-click to bring up the shortcut menu.

3 Modify or delete the Data Source.

3a Select *Properties* to modify the Data Source.

3b Select *Delete* to delete the Data Source

NOTE: Deleting a Data Source does not affect the Data Source definition in the Windows registry. It only removes the Data Source from the Database list, which means Nsure Audit Report can no longer query or run reports on the database.

Defining Your Default Database and Table

Your default database is automatically used for all queries and reports unless a specific database is designated in the query statement or another database is selected in the Database pane.

To define your default database:

1. Click the *Database* field on the status bar.
2. Select the default database from the list.

You can also select the database in the Database pane, right-click, and select *Set As Default Database* or you can choose *Query* > *Default Database* from the menu bar.

To define your default table:

1. Click the *Table* field on the status bar.
2. Type the name of the default table you want to use, or select it if it is already present in the list, then press Enter.

   The default table name defined by the Oracle and MySQL channels is NAUDITLOG.

   IMPORTANT: If you are using an Oracle database, the default table must correspond to the name of the table view, NAUDITLOG. For more information, see Section D.9, "Creating a View in Oracle," on page 224.

You can also click *Query* > *Default Table* from the menu.

9.2.5 Setting Default Options in Nsure Audit Report

The *Options* menu in Nsure Audit Report allows you to set your default sort order, query limits, date/time format, and Clipboard options.

To bring up the *Options* menu, click *View* > *Options* from the menu bar.

The following table reviews the options in the *Options* menu.

Table 9-8 *Nsure Audit Report Options Menu Options*

Option	Description
General	
Default Sort Order	The order in which records are sorted in the query results window. If you select ascending or descending, the records are sorted by the first column in the output which is the source IP address by default.
Result	
Clipboard	The following Clipboard settings determine how query data is copied to the Clipboard.
Copy All If No Entries Selected	If no entries are selected when you press Ctrl+C in the Query Results window, all of the query information is copied to the Clipboard.

Option	Description
Copy 'Raw' Instead of Translated Data	Information is copied to the Clipboard exactly as it appears in the database. This means that all translated data (including IP addresses, signatures, dates, Severity level, Event ID, and other numeric values) are copied in their raw format. For example, a Severity level of Emergency would be copied as a 1.
Copy Field Headers	The field titles defined in the log schema files are copied to the Clipboard along with the data in their associated fields. **IMPORTANT:** This option requires that you import each logging application's log schema. For more information, see "Importing Log Schemas" on page 138.
Column Separation String	When records are copied to the Clipboard, the text string provided in this field is used to delineate the fields within each record. This option facilitates cut and paste functions. For example, if you wanted to cut and paste data into an Excel spreadsheet, you would use a comma (,) delimiter. If you wanted to paste the data into tabular columns, you would use a tab delimiter.
Limits	
Limit Results To __ Rows	This option limits the number of rows (that is, records) that are returned from a database query. This is a global setting. This means that Nsure Audit Report automatically adds this parameter to all database queries unless the parameter is expressly defined in the query statement.
Printing	The following options determine which fonts are used to print query results.
Header Font	The font used to print the column headers.
Data Font	The font used to print the query data (that is, the fields within each record).
Footer Font	The font used to print the footer in the query results page. The footer contains the page number and query string.
Translation	
Date/Time Format	The following options determine how Nsure Audit Report presents date and time information in the query results. This is a global setting. This means that Nsure Audit Report automatically adds this parameter to all database queries unless the parameter is expressly defined in the query statement.
RFC822 UTC	All time and date information is formatted in RFC-822 format, which is the Internet standard format for electronic mail message headers. All values are expressed in UTC.
RFC822	All time and date information is formatted in RFC-822 format. All values are expressed in local time as defined in the workstation's Windows settings.

Option	Description
Locale	All time and date information is formatted according to the standards and formats selected in the workstation's Windows settings. All values are expressed in local time as defined in the workstation's Windows settings.
Locale Date	All date information is formatted according to the standards and formats selected in the workstation's Windows settings. (Time information is not included.) All values are expressed in local date format as defined in the workstation's Windows settings.
Locale Time	All time information is formatted according to the standards and formats selected in the workstation's Windows settings. (Date information is not included.) All values are expressed in local time as defined in the workstation's Windows settings.
Binary Data	Events logged to Nsure Audit include a data field that can contain up to 3072 bytes of data. The following options determine how Nsure Audit Report handles the information in the event's data field.
Don't Display	The data field is not included in the query results.
Display ASCII	The data field is included in the query results and displays in ASCII format.
Display hex	The data field is included in the query results and displays in hexadecimal format.
Display first __ bytes	Limits the amount of information that is included from the data field. The maximum is 3072 bytes.

9.2.6 Importing and Viewing Events in Nsure Audit Report

The Events pane allows you to view logging application properties as well as event data. The information provided on applications and their associated events can be used to define query statements, Notification Filters, and Heartbeat Notifications.

NOTE: For an explanation of event fields, see Section A.1, "Event Structure," on page 175. For information on defining Notification Filters and Heartbeat Notifications, see Chapter 8, "Configuring Filters and Event Notifications," on page 101.

Before you can view a logging application's events, however, you must first import its log schema. The log schema catalogs the events that can be logged for a given application. It also provides the event descriptions and field labels that Nsure Audit Report uses in its reports. For more information, see Section A.4, "Log Schema Files," on page 187.

Importing Log Schemas

Nsure Audit stores each application's log schema (LSC) file in its respective Application object. (English LSC files are stored under the NAuditAppSchemaEn attribute.) Therefore, when you import log schemas, Nsure Audit Report reads the information from the Application objects on the designated logging server.

To import a logging application's log schema in Nsure Audit Report:

1 Click *File* > *Import* > *Application Schema*.

2 Specify the IP address or host name of the Secure Logging Server.

   If you have multiple logging servers, select the Secure Logging Server that loads the logging application associated Application object at startup.

   NOTE: To determine which Secure Logging Server loads the Application object, refer to the logging server's Log Applications page. For more information, see Section 6.2, "Creating Application Objects," on page 65.

3 From the drop-down list box, select the language version of the log schema you want to import.

   If an application does not have a log schema for the selected language, Nsure Audit Report imports the application's English log schema.

   When you click *OK*, Nsure Audit Report connects to the logging server and imports the log schemas from all Application objects in the logging server's supported Application containers. For more information, see Section 6.3, "Application Objects," on page 67.

   IMPORTANT: Nsure Audit Report writes the log schema files to its cache file, `lreport.lsc`, in the Windows Home Directory. Therefore, the account you use to log in to Windows XP must have right to the account's Windows Home Directory. The Windows Home Directory is defined in the User Profile. For more information, see Home Directory in Windows Help.

4 Click *OK* in the confirmation dialog box.

   The *Import Confirmation* dialog box notes that you must restart Nsure Audit Report before the new schemas appear in the Events tab.

5 Restart Nsure Audit Report.

The new applications and their associated events now appear in the Events pane.

Viewing Application Properties and Events

IMPORTANT: To view events in the Events pane, the account you use to log in to Windows XP must have rights to the Windows Home Directory. The Windows Home Directory is defined in the User Profiles. For more information, see "Home Directory" in Windows Help.

To view a logging application's properties in Nsure Audit Report:

1 In the Events pane, select an application.

2 Right-click, then select *Properties*.

The *Application Properties* window displays the following information:

***Table 9-9** Application Properties Window Options*

Attribute	Description
Application Identifier	The name assigned to the current application. The Application Identifier is also stored in the application's certificate. The Application Identifier is part of the Component string for every event logged from the current application. For more information, see Section 6.3, "Application Objects," on page 67.
Application ID	The four-digit hex value assigned to the current application. The Application ID is part of the Event ID for every event logged from the current application. All Application IDs are assigned through Novell Developer Support and are maintained in the Nsure Audit central registry. For more information, see Section 6.3, "Application Objects," on page 67.
Description	The application description provided in the application's log schema.

To view a logging application's events in Nsure Audit Report:

1 In the Events pane, expand the application's folder.

This exposes the application's associated events. Only those events cataloged in the application's log schema appear in this list.

2 Right-click, then select *Properties*.

You can also double-click an event to bring up the Event Properties window.

The Event Properties window displays the Event ID, description, and field information.

For more information on event fields, see Section A.1, "Event Structure," on page 175.

9.2.7 Verifying Event Authenticity in Nsure Audit Report

To provide non-repudiable logs, Nsure Audit can digitally sign each event that is logged to the data store. To sign an event, the logging application or the Platform Agent hashes the event data and signs the hash with the Logging Application's private key. The signature is then stored as part of the event. This signature allows the auditor or investigator to determine if an event has been changed.

To allow auditors to determine if an event has been deleted or the sequence of events has been changed, Nsure Audit can also chain its event signatures. That is, if event chaining is enabled, each event's signature includes its own data as well as the signature from the previous event.

Event chaining is enabled in the Platform Agent's configuration file, `logevent`. For information on configuring this option, see "Logevent" on page 42. It can also be configured through the Secure Logging Server object's *Sign Event* attribute. For more information, see Section 4.2.2, "Logging Server Objects," on page 45.

If event chaining is enabled, Nsure Audit Report can verify that all the events logged to the data store for each logging application are authentic; that is, it can validate the event signatures to determine if an application's events have been tampered with, deleted, or if the sequence of events has been changed.

NOTE: Future iterations of Nsure Audit Report will allow you to verify individual events.

To verify that an application's logged events are authentic:

1. Click *Query* > *Verify Authenticity*.
2. Select the database where the events are logged.
3. Specify the table where the events are logged.
4. Select the logging application for the events you want to validate.
5. If the application is running on multiple systems, specify the IP address or host name of the Platform Agent that logged the events you want to verify in the Source Address field.
6. If you want to filter events, you can specify the component string for the events you want to validate in the *Optional Filter* field.
7. Specify the path and filename for the Logging Application Certificate.

   Click *Browse* to locate the Logging Application Certificate in the directory tree.
8. Click *Verify*.

The following table provides more information on the Verify options.

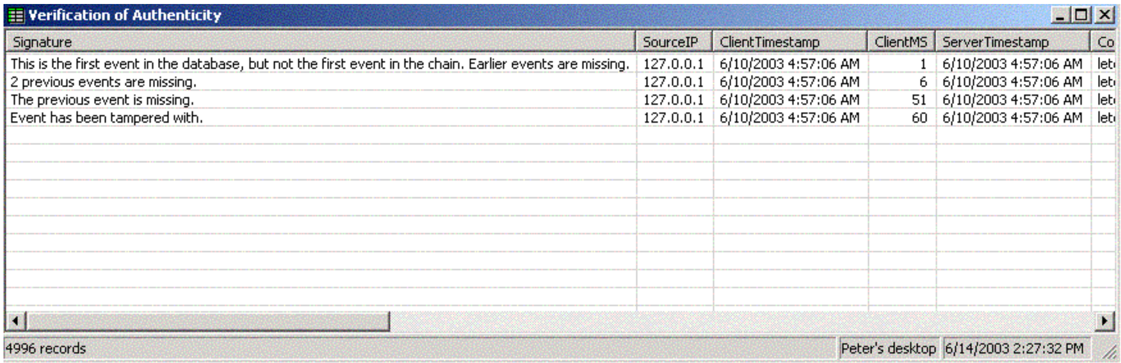
Table 9-10 *Verify Options*

Option	Description
Database	The database containing the events you want to verify.
Table	The table containing the events you want to verify.
Application	The logging application's events you want to verify. In most cases, each logging application has its own certificate. This means that event signatures are typically application-specific. Therefore, Nsure Audit Report verifies event signatures on a per application basis.
Source Address	The IP address or host name of the Platform Agent that logged the events you want to verify. The only time you need to provide a source address is if the logging application is running on multiple systems. This allows Nsure Audit Report to identify which event chain to verify.
Optional Filter	If you only want to verify the events for a specific component or module within a logging application, you can specify the component's component string. For more information on component strings, see Section A.1, "Event Structure," on page 175 and Section A.2, "Component Strings," on page 179.

Option	Description
App Certificate	The path and filename for the Logging Application Certificate used to sign the application's events. By default, the Logging Application Certificates are available in the following directories: • `sys:\system\naudit` (NetWare) • `\program files\novell\nsure audit\logschema\` (Windows) • `/opt/novell/naudit//logschema/` (Linux) • `/opt/NOVLnaudit/logschema/` (Solaris)

After Nsure Audit Report verifies the application's events, it returns the verification results.

Figure 9-4 *Nsure Audit Report Verification Results*



Signature	SourceIP	ClientTimestamp	ClientMS	ServerTimestamp	Co
This is the first event in the database, but not the first event in the chain. Earlier events are missing.	127.0.0.1	6/10/2003 4:57:06 AM	1	6/10/2003 4:57:06 AM	let
2 previous events are missing.	127.0.0.1	6/10/2003 4:57:06 AM	6	6/10/2003 4:57:06 AM	let
The previous event is missing.	127.0.0.1	6/10/2003 4:57:06 AM	51	6/10/2003 4:57:06 AM	let
Event has been tampered with.	127.0.0.1	6/10/2003 4:57:06 AM	60	6/10/2003 4:57:06 AM	let

If the event chain is authentic, Nsure Audit Report returns a message that the table's events have been verified as authentic. If the event chain is not authentic, Nsure Audit Report lists each problem and its associated event.

There following table provides an explanation for each signature error.

Table 9-11 *Signature Errors*

Signature Error	Explanation
Logging application restarted. This is the first event after the restart, but it cannot be verified if events have been removed at the end of the previous chain.	The logging application shut down and restarted, so the event count field (ClientMS) started again at 0; therefore, the event chain was broken. You can determine if the application restart was malicious or not by looking at the last event in the previous event chain. Logging applications send an event when they are unloaded, so if the last event in the previous event chain is an application unload event, you know that no events have been deleted.

Signature Error	Explanation
This is the first event in the database, but not the first event in the chain. Earlier events are missing.	The current event is the first event in the database, however, the event count field (ClientMS) indicates this is not the first event in the chain. This message occurs if you have rolled or expired your data store. You can use the following methods to determine if any events are missing: • If you expired your data store, you can look at the current event's time stamp to see if it occurred at the time you expired the data store. • If you rolled the data store, you can look at the event count field for the last event in the archived data store to determine if it preceded the current event.
The previous event is missing.	The current event's signature does not include the signature from the previous event. Using the event count field (*ClientMS*), Nsure Audit Report can determine that only the previous event is missing.
x previous events are missing.	The current event's signature does not include the signature from the previous event. Using the event count field (ClientMS), Nsure Audit Report can determine approximately how many previous events are missing.
Event has been tampered with.	The current event's signature is not valid. Although it includes the signature from the previous event, the event data in the signature does not match the current data.

9.2.8 Working with Reports in Nsure Audit Report

In Nsure Audit Report, the term "reports" refers specifically to Crystal Decisions Report Template Files (`*.rpt`). Crystal Decisions Reports graphically summarize specific sets of log data in pie charts, bar charts, and so forth.

Nsure Audit Report allows you to import and run Crystal Decisions Reports. Some logging applications might also include their own predefined reports. You do not need Crystal Reports to run reports; however, if you have Crystal Reports, you can customize the predefined reports or you can design your own reports and import those reports into Nsure Audit Report.

Nsure Audit Report stores the report name and path in the Windows registry under `HKEY_CURRENT_USER\Software\Novell\Log Report Application\1.0\Reports`; however, the actual report file is stored in the directory structure.

The following sections discuss working with reports in Nsure Audit Report:

- "Importing Reports" on page 144
- "Deleting Reports" on page 144
- "Running Reports" on page 144

- "Drilling Down on Report Data" on page 144
- "Exporting Reports" on page 145
- "Printing Reports" on page 145

Importing Reports

To import Crystal Decisions Reports:

1. In the Reports pane right-click, then select *Insert*.
2. In the *Name* field, specify the name you want to appear in the Reports pane.
3. In the *File* field, specify the path and filename for the Crystal Decisions Report.

   Click *Browse* to locate the file in the directory tree.
4. When finished, click *OK*.

   Nsure Audit Report adds the report to the Reports pane.

Nsure Audit Report stores the report name and path in the Windows registry under `HKEY_CURRENT_USER\Software\Novell\Log Report Application\1.0\Reports`; however, the actual report file is stored in the directory structure.

Deleting Reports

To delete a Report in Nsure Audit Report:

1. In the Reports pane, select the report you want to delete.
2. Right-click, then click *Delete*.

Running Reports

You do not need Crystal Reports to run Crystal Decisions Reports in Nsure Audit Report.

To run a report in Nsure Audit Report:

1. In the Databases pane, select the database you want to run the report on.

   NOTE: If you do not select a database, the report runs against the default database. For more information, see "Defining Your Default Database and Table" on page 136.
2. In the Reports pane, select the report you want to run.
3. Right-click, then select *Run*.

   Nsure Audit Report opens the report in the Report window.
4. To update the report with live data, click the *Refresh Data* button ⚡ on the Report toolbar.

Drilling Down on Report Data

After you run a report, Nsure Audit Report allows you to drill down on a specific field value. A drill-down report returns all records that match the selected field value.

To run a drill-down report, simply double-click the field value you want to query.

TIP: The mouse pointer appears as a magnifying glass over drill-down fields.

When you run a drill-down report, Nsure Audit Report returns all records that match the given field value. For example, if you drill down on a SourceIP field that has a value of 192.65.102.159, Nsure Audit Report returns all records that have a value of 192.65.102.159 in their SourceIP field.

Exporting Reports

Nsure Audit Report can export reports in a variety of formats including Adobe* Acrobat*, HTML, Microsoft Excel, ODBC, Rich Text Format (RTF), Microsoft Word, text, comma-separated values (CSV), and XML.

To export a report in Nsure Audit Report:

1. Run a report.

   For step-by-step instructions, see "Running Reports" on page 144.
2. Click *File* > *Export*.
3. Select the file format that you want to export the report to (for example, `.pdf`, `.txt`, `.xml`, and so forth).
4. Select the export destination.
5. Click *OK*.

   Nsure Audit Report brings up the *Save As* dialog box.
6. Specify the export filename, then click *Save*.

Nsure Audit Report exports the file in the designated format to the designated path and filename.

Printing Reports

To print a report to your default printer:

1. Run a report.

   For step-by-step instructions, see "Running Reports" on page 144.
2. Click the *Print* button on the Report toolbar.

9.2.9 Working with Queries in Nsure Audit Report

Nsure Audit Report uses queries to request information from MySQL and Oracle databases. All Nsure Audit Report queries are defined in SQL. Although you must be familiar with the SQL language to create SQL query statements, this is the most powerful and flexible query method.

If you are unfamiliar with the SQL language, Nsure Audit Report includes a Query Expert to help you define basic query statements. You can also import existing query sets and run them within Nsure Audit Report.

Nsure Audit Report stores all queries in the Windows registry under `HKEY_CURRENT_USER\Software\Novell\Log Report Application\1.0\Queries`.

The following sections provide details on working with queries in Nsure Audit Report:

- "Importing Queries" on page 146
- "Creating Manual Queries" on page 121

- “Creating Queries Using the Query Expert” on page 147
- “Modifying and Deleting Saved Queries” on page 126
- “Custom Query Macros” on page 150
- “Running Queries” on page 151
- “Managing Query Results in Nsure Audit Report” on page 154
- “Exporting Query Results in Nsure Audit Report” on page 155
- “Printing Query Results in Nsure Audit Report” on page 156

Importing Queries

To import existing queries, you must save the queries in a text-based query file. The query file format requires that you enclose the title of each query in square brackets [ ]. The query statement is on the line following the title. If you want to enable the *Translate Column Titles* option, enter Translate=1 on the line following the query statement. Empty lines are not legal and any line that starts with a hash (#) is commented out.

For information on the *Translate Column Titles* option, see “Manually Creating Queries” on page 146.

The following is a sample query file. It contains two queries: All Connection Cleared events and All Directory Remove events.

```
# Query File
#
[All 'Connection Cleared' events]
SQL=SELECT * FROM log WHERE eventid=655622
Translate=1
[All 'Directory Remove' events]
SQL=SELECT * FROM log WHERE eventid=655368
Translate=1
```

To import SQL queries in Nsure Audit Report:

1. Click *File* > *Import* > *Query Set*.
2. Select the query file in the directory tree, then click *Open*.

The queries contained in the query file now appear in the Query pane.

All imported queries are stored in the Windows registry under `HKEY_CURRENT_USER\Software\Novell\Log Report Application\1.0\Queries`.

Manually Creating Queries

All queries are stored in the Windows registry under `HKEY_CURRENT_USER\Software\Novell\Log Report Application\1.0\Queries`.

To manually create a query in Nsure Audit Report:

1. Click *Query* > *Manual*.

   You can also right-click in the Queries pane and select *Insert*.

2 In the *Name* field, specify the name you want to use to refer to this query.

3 Define the query statement in the Query window.

4 Select *Translate Column Titles* if you want Nsure Audit Report to label the query results with the field titles defined in the log schema.

5 When finished, click *OK*.

NOTE: When using mySQL, use the LIMIT statement to prevent an excessively large number of records from being returned. Even with a maximum value set in Nsure Audit Report, mySQL returns up to LIMIT records.

The query now appears in the Queries pane.

The following table provides further information on the *Direct Query* options.

Table 9-12 *Direct Query Options*

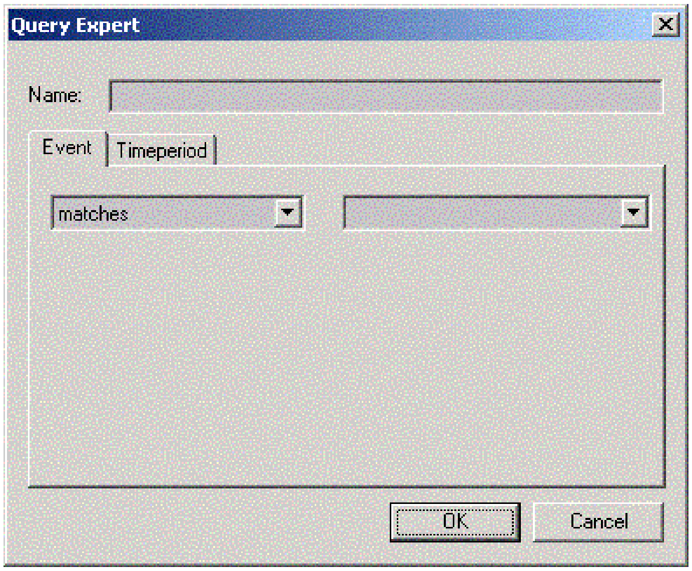
Query Option	Description
Name	The name you want to use to refer to this query. The query name is listed in the Queries pane and it appears as the title in the query results window.
Query	The query statement. Queries are defined using the SQL query language. For basic information on SQL queries, see the *MySQL Reference Manual* (http://www.mysql.com/documentation/index.html). If you want your query to dynamically build the FROM clause using the currently selected database and table, enter FROM $I in the query statement.
Translate Column Titles	Select *Translate Column Titles* if you want Nsure Audit Report to label the column headings with the field titles defined in the log schema. We recommend that you select this option only for queries that return one type of event. If you select this option for queries that return multiple types of events, Nsure Audit Report labels the column headings with the field titles from the last event returned in the query. **IMPORTANT:** For this option to work, you must import each application's log schema. For information, see "Importing Log Schemas" on page 138.

Creating Queries Using the Query Expert

IMPORTANT: To use the Query Expert, the account you use to log in to Windows XP must have rights to the Windows Home Directory. The Windows Home Directory is defined in the User Profiles. For more information, see Home Directory in Windows Help.

If you are unfamiliar with the SQL query language, you can use the Query Expert to help you define basic database queries. The Query Expert simplifies the process of creating a query by allowing you to choose from lists of predefined parameters. The Query Expert then constructs the query statement from the parameters you select.

Figure 9-5 *Nsure Audit Report Query Expert*



Because the Query Expert can provide only a limited set of parameters, the queries it creates are very simple. However, it is the easiest way to create queries and it is capable of creating most base-level queries.

To create a query using the Query Expert in Nsure Audit Report:

1. Click *Query* > *Expert*.
2. Specify the name you want to use to refer to this query in the *Name* field.
3. Define the query parameters.
   - **3a** In the *Event* tab, define the condition and the event.
   - **3b** In the *Timeperiod* tab, select a time parameter.
4. When finished, click *OK*.

NOTE: When using mySQL, use the LIMIT statement to prevent an excessively large number of records from being returned. Even with a maximum value set in Nsure Audit Report, mySQL returns up to LIMIT records.

The query now appears in the Queries pane.

The following table provides further information on the Query Expert's parameters.

Table 9-13 *Query Expert Parameters*

Parameter	Description
Event	

Parameter	Description
matches is less than is more than is between	The query condition.
Events	The event parameter. The drop-down menu includes all the events for every application listed in the Events pane. If the application's log schema provides event descriptions, Nsure Audit Report displays those descriptions in the *Event* list; however, the events are still sorted by their numeric Event ID. Therefore, the event descriptions are not listed in alphabetical order, but related events are grouped together.
Timeperiod	The time parameter. The query returns only events that occurred in the designated time frame.

The basic structure of query statements created with the Query Expert is as follows:

```
SELECT * FROM table WHERE eventid condition event(s) AND
clienttimestamp> $time_parameter
```

Shortcuts

The Query Expert provides some quick and easy ways to create queries for all the events in a single application or for a single event.

To create a query for all the events in a single application:

1. Select an application in the Events pane.
2. Right-click, then select *Define Query*.
3. In the Timeperiod tab, select a time parameter.
4. Click *OK*.

To create a query for a single event:

1. Expand one of the application folders in the Events pane.
2. Select an event.
3. Right-click, then select *Define Query*.
4. In the Timeperiod tab, select a time parameter.
5. Click *OK*.

Modifying Queries

To modify a Query in Nsure Audit Report:

1. In the Queries pane, select the query you want to modify or delete.
2. Right-click, then select *Properties*.
3. Modify the query.

For information on the query options, see "Creating Saved Queries" on page 122.

Deleting Queries

To delete a Query in Nsure Audit Report:

1. In the Queries pane, select the query you want to delete.
2. Right-click, then select *Delete*.

Custom Query Macros

Nsure Audit Report has some powerful custom macros that simplify data queries. The following table lists the custom query macros that can used in Nsure Audit Report.

IMPORTANT: All query macros must be preceded by a dollar sign ($).

Table 9-14 *Nsure Audit Report Custom Query Macros*

Function	Description
I	The currently selected database and table. If you want your query to dynamically build the FROM clause using the currently selected database and table, enter FROM $I in the query statement.
Now	The current date and time in local time as defined in the workstation's Windows settings.
ThisMonth	The current month in local time as defined in the workstation's Windows settings
Today	The current day in local time as defined in the workstation's Windows settings.
LastWeek	The previous week in local time as defined in the workstation's Windows settings.
Yesterday	Yesterday in local time as defined in the workstation's Windows settings.
Date(*mm-dd-yyyy*)	The given date in local time.
Hex(*x*)	The decimal value of hex value `x`
IP(*`address`*)	An IP address in IPv4 dotted address form, or in host name form.
Prompt(*`message`*)	Prompts the user for a value. When a query that contains this variable is run, the user is prompted to input a value based on `text`. For example, `select * from $1 where text2='$Prompt(Please Provide the User Name for which you want to query:)';`, prompts the user to provide a value based on the prompt provided by `message`. This value is then used in the query.
PromptDate(*mm-dd-yyyy*)	Prompts the user for a date value, then converts it into seconds since 1970, making it consistent with the way dates are stored in the Nsure Audit database. The date must be in the form mm-dd-yyyy.

Function	Description
PromptHex(*hex_value*)	Prompts the user for a hex value, and converts it to an integer value.
PromptIP(*host_or_ip*)	Prompts the user for a host name or dot-formatted IP address, and converts it to an integer consistent to the way IP addresses are stored in the Nsure Audit database.

The following sample query statement illustrates how the query macros can be used:

```
SELECT * FROM $1 WHERE eventid='$Prompt(Please provide the Event ID to
query:)' AND clienttimestamp>$WEEK
```

Running Queries

To run a query in Nsure Audit Report:

1 Select the database you want to query.

  1a Click the *Database* field on the status bar.

  1b Select the default database from the list.

2 Select the table you want to query.

  2a Click the *Table* field on the status bar.

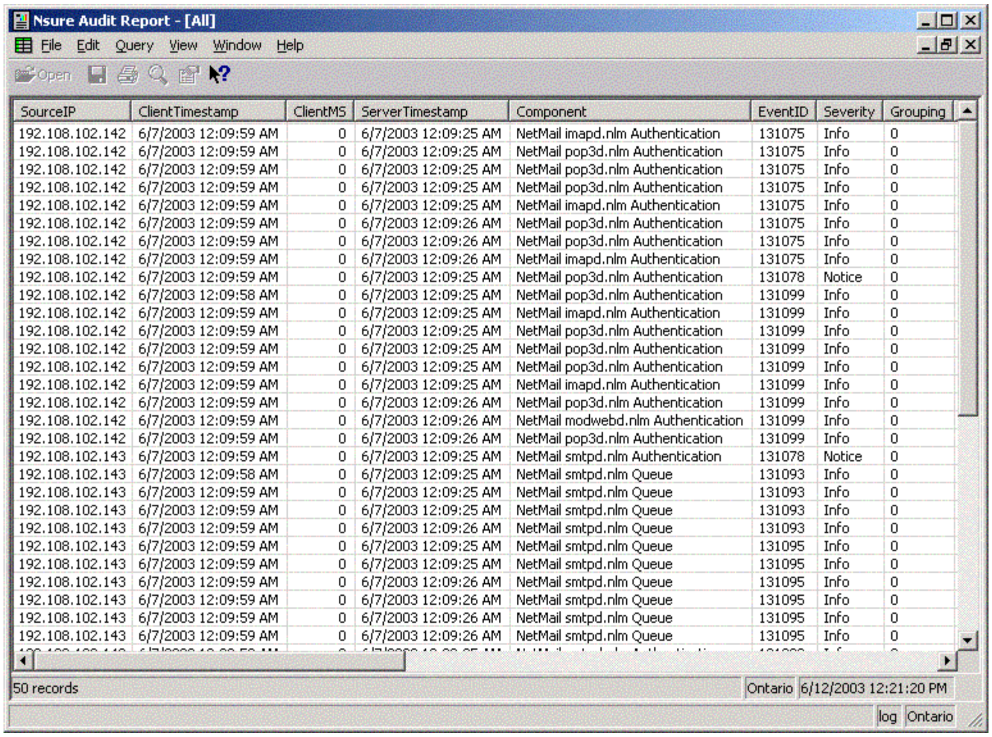
  2b Select the default table from the drop-down list, then press Enter.

3 In the Queries pane, select the query you want to run.

4 Right-click, then select *Run*.

Nsure Audit Report returns the query results in a data table; rows represent individual records and columns represent fields within those records. You can click any of the column headings to sort the results by that field.

Figure 9-6 *Query Results in Nsure Audit Report*



SourceIP	ClientTimestamp	ClientMS	ServerTimestamp	Component	EventID	Severity	Grouping
192.108.102.142	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:25 AM	NetMail imapd.nlm Authentication	131075	Info	0
192.108.102.142	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:25 AM	NetMail pop3d.nlm Authentication	131075	Info	0
192.108.102.142	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:25 AM	NetMail pop3d.nlm Authentication	131075	Info	0
192.108.102.142	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:25 AM	NetMail pop3d.nlm Authentication	131075	Info	0
192.108.102.142	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:25 AM	NetMail imapd.nlm Authentication	131075	Info	0
192.108.102.142	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:26 AM	NetMail pop3d.nlm Authentication	131075	Info	0
192.108.102.142	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:26 AM	NetMail pop3d.nlm Authentication	131075	Info	0
192.108.102.142	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:26 AM	NetMail imapd.nlm Authentication	131075	Info	0
192.108.102.142	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:25 AM	NetMail pop3d.nlm Authentication	131078	Notice	0
192.108.102.142	6/7/2003 12:09:58 AM	0	6/7/2003 12:09:25 AM	NetMail pop3d.nlm Authentication	131099	Info	0
192.108.102.142	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:25 AM	NetMail imapd.nlm Authentication	131099	Info	0
192.108.102.142	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:25 AM	NetMail pop3d.nlm Authentication	131099	Info	0
192.108.102.142	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:25 AM	NetMail pop3d.nlm Authentication	131099	Info	0
192.108.102.142	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:25 AM	NetMail pop3d.nlm Authentication	131099	Info	0
192.108.102.142	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:25 AM	NetMail imapd.nlm Authentication	131099	Info	0
192.108.102.142	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:26 AM	NetMail pop3d.nlm Authentication	131099	Info	0
192.108.102.142	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:26 AM	NetMail modwebd.nlm Authentication	131099	Info	0
192.108.102.142	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:26 AM	NetMail pop3d.nlm Authentication	131099	Info	0
192.108.102.143	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:25 AM	NetMail smtpd.nlm Authentication	131078	Notice	0
192.108.102.143	6/7/2003 12:09:58 AM	0	6/7/2003 12:09:25 AM	NetMail smtpd.nlm Queue	131093	Info	0
192.108.102.143	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:25 AM	NetMail smtpd.nlm Queue	131093	Info	0
192.108.102.143	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:25 AM	NetMail smtpd.nlm Queue	131093	Info	0
192.108.102.143	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:26 AM	NetMail smtpd.nlm Queue	131093	Info	0
192.108.102.143	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:25 AM	NetMail smtpd.nlm Queue	131095	Info	0
192.108.102.143	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:25 AM	NetMail smtpd.nlm Queue	131095	Info	0
192.108.102.143	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:26 AM	NetMail smtpd.nlm Queue	131095	Info	0
192.108.102.143	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:26 AM	NetMail smtpd.nlm Queue	131095	Info	0
192.108.102.143	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:26 AM	NetMail smtpd.nlm Queue	131095	Info	0
192.108.102.143	6/7/2003 12:09:59 AM	0	6/7/2003 12:09:26 AM	NetMail smtpd.nlm Queue	131095	Info	0

50 records | Ontario | 6/12/2003 12:21:20 PM

log | Ontario

If you selected the *Translate Column Titles* option when you defined the query, Nsure Audit Report labels the query results with the field titles defined in the log schema. Nsure Audit Report also displays each event's field titles as you mouse over the event fields.

NOTE: We recommend that you select only the *Translate Column Titles* option for queries that return one type of event. If you select this option for queries that return multiple types of events, Nsure Audit Report labels the column headings with the field titles from the last event returned in the query. For more information on the Translate Column Titles option, see "Manually Creating Queries" on page 146.

Running Event Distributions

Event Distributions tell you how many times each type of event has occurred for a given application. For example, if you run an Event Distribution for NetWare, Nsure Audit Report returns the number of times each event listed in the NetWare log schema occurred.

To run an Event Distribution in Nsure Audit Report:

1 Select the database you want to query.

1a Click the *Database* field on the status bar.

1b Select the default database from the list.

2 Select the table you want to query.

2a Click the *Table* field on the status bar.

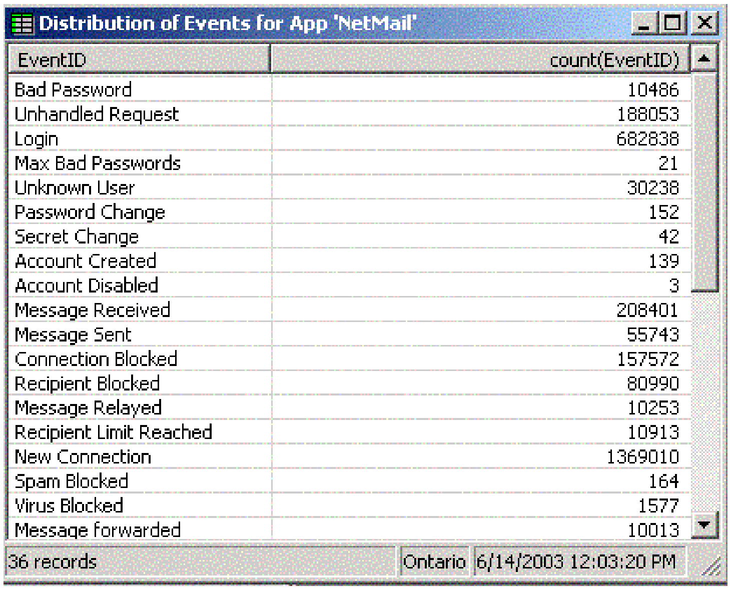
2b Select the default table from the drop-down list, then press Enter.

3 In the Events pane, select an application.

4 Right-click, then select *Distribution*.

Nsure Audit Report returns the number of times each of the application's events occurred in the selected database.

Figure 9-7 *Event Distribution Report in Nsure Audit Report*



Distribution of Events for App 'NetMail'

EventID	count(EventID)
Bad Password	10486
Unhandled Request	188053
Login	682838
Max Bad Passwords	21
Unknown User	30238
Password Change	152
Secret Change	42
Account Created	139
Account Disabled	3
Message Received	208401
Message Sent	55743
Connection Blocked	157572
Recipient Blocked	80990
Message Relayed	10253
Recipient Limit Reached	10913
New Connection	1369010
Spam Blocked	164
Virus Blocked	1577
Message Forwarded	10013

36 records | Ontario | 6/14/2003 12:03:20 PM

The Distribution window lists the Event ID and how many times that Event ID occurred. To sort on the Event ID, click the *Event ID* column. To sort by the number of occurrences, click the Count column.

If the application's log schema provides event descriptions, Nsure Audit Report displays those descriptions in the Event ID column. However, when you sort on the Event ID column, events are sorted by their numeric Event ID, not by their description. Consequently, the event descriptions are not listed in alphabetical order, but related events are grouped together.

Counting Events

If you want to know how many events have been logged for a given application, or the number of occurrences for a specific event, you can run an event count. An event count simply returns the number of events logged to the current database.

To run an event count for a logging application in Nsure Audit Report:

1 Select the database you want to query.

1a Click the *Database* field on the status bar.

1b Select the default database from the list.

2 Select the table you want to query.

2a Click the *Table* field on the status bar.

2b Select the default table from the drop-down list, then press Enter.

3 Select an application in the Events pane.

4 Right-click, then select *Count*.

Nsure Audit Report returns the total number of events that have been logged for the selected application.

To run an event count for a single event:

1 Select the database you want to query.

1a Click the *Database* field on the status bar.

1b Select the default database from the list.

2 Select the table you want to query.

2a Click the *Table* field on the status bar.

2b Select the default table from the drop-down list, then press Enter.

3 Expand an application folder in the Events pane.

4 Select one of the application's events.

5 Right-click, then select *Count*.

Nsure Audit Report returns the total occurrences of the selected event.

Managing Query Results in Nsure Audit Report

After it returns the query results, Nsure Audit Report allows you to further process the data by dynamically sorting records, running drill-down queries, copying specific records, and viewing event properties.

Sorting Records

To sort the query results by a specific field, click the corresponding field heading. Nsure Audit Report toggles between ascending and descending order. The first time you click, Nsure Audit Report sorts the records in ascending order. If you click again, it sorts the record in descending order, and so forth.

Drilling Down on Query Data

After you run a report, Nsure Audit Report allows you to drill down on a specific field value. A drill-down report returns all records that match the selected field value.

To run a drill-down query:

1 Position your mouse pointer over the field value you want to query.

2 Right-click, then select *Drill-down*.

Nsure Audit Report returns all records that match the selected field value. For example, if you drill-down on a *SourceIP* field that has a value of 192.65.102.159, Nsure Audit Report returns all records that have a value of 192.65.102.159 in the *SourceIP* field.

Copying Records

To copy specific records from the query results:

1 Select the records you want to copy.

  1a Shift-click to select contiguous records.

  1b Ctrl+click to select non-contiguous records.

2 Right-click, then select *Copy*.

  You can also press Ctrl+C.

The query data can be copied to the Windows Clipboard in raw format, it can have field delimiters, and it can include field headers. How information is copied to the Windows Clipboard is managed through the Clipboard settings in the Nsure Audit Report Options menu. For more information, see Section 9.2.5, "Setting Default Options in Nsure Audit Report," on page 136.

NOTE: If the *Copy All if No Entries Selected* option is selected in the *Options* menu, you can copy all the records in the query results window by not selecting any records and pressing Ctrl+C.

Viewing Individual Records

To view a specific a specific record's properties:

1 Select the record you want to view.

2 Right-click, then select *Properties*.

  You can also double-click the record.

Exporting Query Results in Nsure Audit Report

Nsure Audit Report can export query results in the following formats:

- HTML (`*.htm`)
- Comma-separated values (`*.csv`)
- Text (tab-delimited) (`*.txt`)

To export query results in Nsure Audit Report:

1 Run a query.

  For step-by-step instructions, see "Running Queries" on page 151.

2 Click *File* > *Export*.

3 In the *Export Results* menu, define the export file's path and filename.

4 Click the *Save As Type* drop-down list, then select the export format.

5 Click *Save*.

Exporting Specific Records

1 Select the records you want to export.

  1a Shift-click to select contiguous records.

  1b Ctrl+click to select non-contiguous records.

2 Click *File* > *Export*.

Printing Query Results in Nsure Audit Report

To print the query results to your default printer:

1 Run a query.

For step-by-step instructions, see "Running Queries" on page 151.

2 Click *File* > *Print*.

You can also right-click, then select *Print*.

Printing Specific Records

To print specific records from the query results:

1 Select the records you want to print.

1a Shift-click to select contiguous records.

1b Ctrl+click to select non-contiguous records.

2 Click *File* > *Print*.

You can also right-click, then select *Print*.

9.3 Using Other Utilities to Access Log Data

Depending on your system's data store, there are other ways to access your log data. The following sections review alternative methods to access your log data:

- Section 9.3.1, "Using LETrans to Access Data Logged by the File Channel," on page 156
- Section 9.3.2, "Using Third-Party Product to Access Log Data," on page 157

9.3.1 Using LETrans to Access Data Logged by the File Channel

The File channel allows the logging server to log events directly to file in raw format or to translate those events to a human-readable log file.The advantage of using the File channel to log system events is that it can log a large number of events per second; however, it cannot be queried using iManager or Nsure Audit Report.

LETrans is a command line utility that allows users to access data logged by the File Channel. Its primary function is to translate raw text log files into human-readable form. However, it also provides the ability to query an ODBC data source on your Windows machine, then translate and format the output.

The LEtrans utility takes no parameters; it is configured using the `letrans.cfg` file. The `letrans.cfg` file contains a description of each LETrans configuration option.

To Launch LETrans:

1 Open `letrans.cfg` in a text editor. LETrans and `letrans.cfg` are located in the following directories:

- NetWare: `sys:\system\naudit`

- Windows: `\program files\novell\nsure audit`
- Linux: `/opt/novell/naudit`
- Solaris: `/opt/NOVLnaudit`

2 List the path and name of each untranslated log file in the source files section.

3 Add the path to the log schema file (`*.lsc`) for any additional instrumentations you are using in the schema section.

4 Save `letrans.cfg`, then execute LETrans from the server.

9.3.2 Using Third-Party Product to Access Log Data

Because Nsure Audit logs events to standard systems (MySQL, Oracle, Microsoft SQL Server, syslog, and delimited text files), you can directly access log data using any tool that is standardized to those systems. For example, you can access data in MySQL and Oracle systems using ODBC or JDBC tools. Text files can be opened with a standard text reader such as Windows Notepad or VIM (UNIX) or an application that supports delimited text files such as Microsoft Excel.

NOTE: You can reference the logging applications' log schema (LSC) files to identify the log data event fields. For more information, see "How LSC Files Are Used" on page 188.

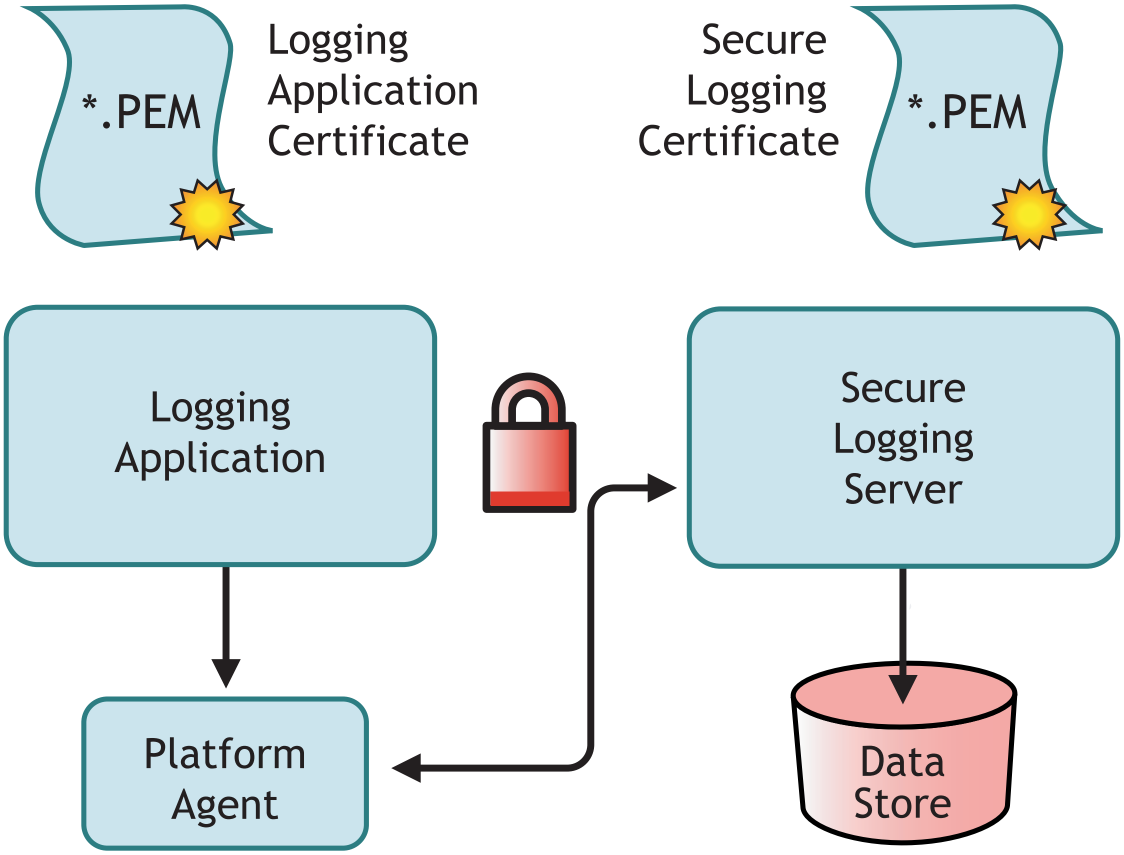
Security and Non-Repudiation 10

Novell® Nsure™ Audit leverages digital certificates and signatures to ensure your log files are valid and non-repudiable. This section includes the information you need to authenticate your logging applications and validate your data store's record of events.

10.1 Authenticating Logging Applications

The Secure Logging Server uses digital certificates and Application IDs to verify the identity of all its logging applications. In fact, the Secure Logging Server only accepts connections from applications that have a valid Logging Application Certificate and Application Identifier. This ensures that unknown or spoofed entities cannot submit events to the data store.

NOTE: The Application Identifier is the name the logging application uses to identify itself to the logging server. The Application Identifier is stored in the application's certificate and Application object. For more information, see Section 6.3, "Application Objects," on page 67.



The basic authentication process is as follows:

1. The logging application calls the Platform Agent.
2. The Platform Agent submits the application's certificate to the Secure Logging Server.
3. The Secure Logging Server validates the certificate and verifies the Application Identifier stored in the certificate.
   - A valid Logging Application Certificate must be signed with the Secure Logging Server's own certificate.
   - A valid Application Identifier must be associated with an Application object in one of the Secure Logging Server's supported Application containers.
4. If the certificate and the Application ID are valid, the Secure Logging Server accepts the logging application's connection.
5. The logging application begins to log events.

The Secure Logging Server's certificate (the Secure Logging Certificate) is the logging system's root certificate; that is, it is used to sign certificates for all the logging applications. Every instrumented application must have a certificate signed by the Secure Logging Server's certificate.

The Secure Logging Server and all logging applications ship with their own embedded certificates. Using these certificates, the Secure Logging Server is able to validate each logging application's identity; however, the embedded certificates are not necessarily "secure" because the same certificates are distributed with every copy of the software.

If you want to further secure your logging system, you can use custom certificates generated with the AudCGen utility. To generate your own certificates, see Section 10.3, “Creating the Secure Logging Certificate,” on page 162 or Section 10.4, “Creating Logging Application Certificates,” on page 163.

10.2 Signing Events

Logging is an effective and useful way to determine what is happening on your system. The resulting data can be used for many administrative activities, including troubleshooting, usage characterizations (determining how people are using a system), and determining the cause (or responsible party) of an undesirable event.

However, if you are trying to hold someone accountable for an undesirable event (such as accessing unauthorized information or sabotaging data), standard unprotected logs can be insufficient. This is because individuals can tamper with logs to delete any trace of their actions or to imply another cause for the event. Alternatively, an individual might simply claim that the logs have been tampered with, even if they haven’t. Either case makes it difficult to hold an individual accountable for an event, especially if the individual in question is a system administrator who has access to the logs or has the ability to generate counterfeit events.

For this reason, any logging system that is used for forensic evidence must provide non-repudiation. In other words, the system must, at a minimum, be able to prove that logs have not been tampered with so that a clear tie to the responsible party can be made.

Nsure Audit achieves non-repudiation by digitally signing each event that is logged to the data store. To sign an event, the logging application or the Platform Agent hashes the event data and signs the hash with the Logging Application's private key. The signature is then stored as part of the event. This signature allows the auditor or investigator to determine if an event has been changed. To validate an event, the signature process is simply repeated and the two signatures are compared. If the signatures match, the event is valid; if not, the event has been tampered with.

To allow auditors to determine if an event has been deleted or if the sequence of events has been changed, Nsure Audit can chain its event signatures. That is, if event chaining is enabled, each event's signature includes its own data as well as the signature from the previous event. To validate the event chain, all the logged events for a single application are re-signed and the event signatures are compared. If an event has been deleted or the sequence of events has been changed, the signatures in the chain will not match.

NOTE: Event chaining is enabled in the Platform Agent's configuration file, logevent. For information on configuring this option, see “Logevent” on page 42. For information on validating events in Nsure Audit Report, see Section 9.2.7, “Verifying Event Authenticity in Nsure Audit Report,” on page 140.

Using digital signatures and event chaining, auditors can prove with certainty that specific events did in fact occur. If a log includes the events and the events' digital signatures confirm that the log has not been modified, that log can be used as non-repudiable forensic evidence in disciplinary or legal proceedings.

For example, let's say that an employee (Joe) has been accused of insider trading. To prove this charge, the company must be able to provide evidence that Joe accessed financial information that then influenced his stock trades. The investigator audits the logs to determine if and when Joe accessed the financial information. The log events show that Joe did indeed access the financial information before making the stock trades in question. If Joe denies the allegation, the investigator can show that digital signatures have been used, that the signatures on the events are valid, and that the logged events indicate that Joe performed the action.

10.3 Creating the Secure Logging Certificate

In the current iteration of Novell Nsure Audit, the Secure Logging Certificate is the system's Certificate Authority (CA); that is, it is the trusted, root certificate that is used to validate all other certificates. Therefore, the Secure Logging Certificate is self-signed and it is used to sign all Logging Application Certificates.

NOTE: Future iterations will be able to use secure certificates from an external CA.

To generate a Secure Logging Certificate, enter the following command at the command prompt:

```
audcgen -cert:filename -pkey:filename [-f] [-bits:number] [-
serial:number] -ss
```

The following table reviews each of the command parameters:

Parameter	Description
`-cert:filename`	The output path and filename for the Secure Logging Certificate. The default path and filename is \cacert.pem .
`-pkey:filename`	The output path and filename for the Secure Logging Certificate's private key. The default path and filename is \capkey.pem .
[-f]	Force overwrite. AudCGen overwrites any existing certificates or private keys of the same name (for example, cacert.pem or capkey.pem) in the output directory. This parameter is optional. If you do not use the -f parameter and there is an existing file, AudCGen aborts creation of the certificate.
`[-bits:number]`	The number of bits for the certificate. The default is 512; however, Nsure Audit can handle certificates up to 1472 bits. The Platform Agent rejects certificates larger than 1472 bits.
`[-serial:number]`	This parameter assigns a serial number to the current certificate. You can use this option to keep track of your system's certificates. This parameter is optional.
-ss	Self-sign. AudCGen generates a self-signed CA certificate and key.

The following is a sample command to create a Secure Logging Certificate:

```
audcgen -cert:c:\cacert.pem -pkey:c:\capkey.pem -f -bits:512 -
serial:12345 -ss
```

10.3.1 Configuring the Secure Logging Server to Use a Custom Certificate

To enable the Secure Logging Server to use a custom certificate and private key, you must configure the Secure Logging Certificate File and Secure PrivateKey File attributes on the Logging Server object. For more information, see "Logging Server Objects" on page 45.

10.4 Creating Logging Application Certificates

To generate a Logging Application Certificate, enter the following command at the command prompt:

```
audcgen -cert:filename -pkey:filename [-f] [-bits:number] [-
serial:number] -appcert:filename
-apppkey:filename -app:Application_Identifier
```

The following table reviews each of the command parameters:

Parameter	Description
`-cert:filename`	The path and filename to the Secure Logging Certificate that AudCGen uses to sign the Logging Application Certificate. The default path and filename is \cacert.pem .
`-pkey:filename`	The path and filename to the Secure Logging Certificate's private key. The default path and filename is \capkey.pem .
[-f]	Force overwrite. AudCGen overwrites any existing certificates or private keys of the same name (for example, appcert.pem or apppkey.pem) in the output directory. This parameter is optional. If you do not use the -f parameter and there is an existing file, AudCGen aborts creation of the certificate.
`[-bits:number]`	The number of bits for the certificate. The default is 512; however, Nsure Audit can handle certificates up to 1472 bits.
`[-serial:number]`	This parameter assigns a serial number to the current certificate. You can use this option to keep track of your system's certificates. This parameter is optional.

Parameter	Description
`-appcert:`*`filename`*	The output path and filename for the Logging Application Certificate. The default path and filename is /appcert.pem .
`-apppkey:`*`filename`*	The output path and filename for the Logging Application Certificate. The default path and filename is /apppkey.pem .
`–app:`*`Application_Identifier`*	The logging application's Application Identifier. This value must match the Application Identifier stored the logging application's Application object.

The following is a sample command to create a Logging Application Certificate for the Novell eDirectory™ Instrumentation:

```
audcgen -cert:c:\cacert.pem -pkey:c:\capkey.pem -f -bits:512 -
serial:12345
-appcert:c:\appcert.pem -apppkey:c:\apppkey.pem -app:eDirInst
```

10.4.1 Enabling Logging Applications to Use Custom Certificates

The process of enabling a logging application to use a custom Logging Application Certificate can vary per application. Please refer to the logging application's documentation.

To enable the eDirectory Instrumentation to use a custom Logging Application Certificate, the path and filename for the certificate and private key files must be as follows:

Platform	Certificate Path and Filename	PrivateKey Path and Filename
NetWare®	\system\dsicert.pem	\system\dsipkey.pem
Windows	\windows_directory\dsicert.pem	\windows_directory\dsipkey.pem
Linux and Solaris	/etc/dsicert.pem	/etc/dsipkey.pem

The NetWare Instrumentation requires \system\nwicert.pem and \system\nwipkey.pem .

The NAudit Instrumentation uses the Secure Logging Certificate and private key configured on the Logging Server object.

10.5 Validating Certificates

In the Nsure auditing system, all certificates must be signed by the Secure Logging Certificate and they must contain an Application Identifier.

To determine if a certificate is valid, enter the following command:

```
audcgen -cert:filename -v -appcert:target_certificate
```

The following table reviews each of the command parameters:

Parameter	Description
`-cert:`*`filename`*	The path and filename for the Secure Logging Certificate that AudCGen uses to validate the certificate. The default path and filename is /cacert.pem .
`-v`	Validate. AudCGen validates the certificate designated in the -appcert parameter.
`-appcert:`*`filename`*	The path and filename for the Logging Application Certificate to validate.

The following is a sample command to validate the Logging Application Certificate for the eDirectory Instrumentation:

```
audcgen -cert:c:\cacert.pem -v -appcert:c:\windows\dsicert.pem
```

Troubleshooting NSure Audit

11

This section reviews the common issues you might encounter in Novell® Nsure™ Audit or its associated utilities. It also covers how to uninstall Novell Nsure Audit.

11.1 Common Issues

This section lists common issues you might encounter with Novell Nsure Audit. These include:

11.1.1 Secure Logging Server Does Not Load

If the Secure Logging Server does not load, check the log for any errors that occurred during startup.

- On NetWare®, check the console log (4.x/5.x) or logger screen (6.x) for any errors during load.
- On Windows, check the \nproduct.log file for any errors that were logged during startup.
- On Linux and Solaris, check the screen for any errors that were printed during load.

The following table reviews the problems that might prevent the Secure Logging Server from loading.

Cause	Explanation/Solution
The driver for the default log channel could not be loaded/Could not initialize logging subsystem	If this is the case, the Secure Logging Server will abort loading. This is done to ensure that the administrator is aware of this potentially serious condition. `Solution` Check the error message and fix the indicated problem. For example, if the MySQL driver cannot connect to the MySQL server, load the MySQL server.
Could not authenticate to MDB.	This error occurs if the Secure Logging Server cannot successfully initialize the MDB interface or if the host is not configured to run the Secure Logging Server. MDB is the directory interface used by Nsure Audit. For more information, see Section H.1, "Program Files and Directories," on page 245. `Solution` • Ensure that eDirectory is working properly on the host. • Configure the host to run the Secure Logging Server.
Not enough memory	During startup, the Secure Logging Server allocates memory for its own operation and for the event cache. If this memory cannot be allocated, the Secure Logging Server will terminate. `Solution` • Check your Secure Logging Server configuration and adjust the cache settings accordingly. • Unload unneeded modules loaded during startup.

The server cannot log in to the database.	The Secure Logging Server cannot log in to the data store.
	`Solution`
	Ensure that your MySQL password is correct. Try logging into the MySQL monitor using the exact settings that are configured on the MySQL channel. For example if the MySQL server IP address is 151.155.167.249, the surname is auditusr, and the password is auditpwd, log in to the MySQL monitor with the following syntax:
	`mysql -h 151.155.167.249 -u auditusr -p`
	If you cannot log in, then the MySQL rights are probably not set up correctly.

11.1.2 Events Are Not Being Logged

The following table reviews the problems that might prevent events from being logged.

Cause	Explanation/Solution
The logging application is logging events to cache	If a logging application loads the Platform Agent while the Secure Logging Server is not, or not yet, loaded, its events will be logged to the Platform Agent's Disconnected Mode Cache. By default, the Platform Agent tries to reconnect to the Secure Logging Server only every ten minutes. The Logging Cache Module, on the other hand, tries to upload its cached files to the Secure Logging Server every two minutes. `Solution` Wait for the Platform Agent to reconnect to the Secure Logging Server. For information on changing the Platform Agent's reconnect interval, see "Logevent" on page 42. If you don't see the events after the configured upload interval, verify that your configuration is correct.
A component is misconfigured	There can be a misconfiguration on the Platform Agent or the Secure Logging Server that prevents events from being logged. `Solution` • On the Platform Agent, verify that the LogHost specified in the logevent file points to the intended server. For information on configuring the Platform Agent's LogHost parameter, see "Logevent" on page 42. • Verify that the computer where the platform agent is running can actually reach the loghost via TCP/IP. (A ping will quickly tell.) • If the logging application logs only debug level events, check whether the LogDebug option is set to Never, in which case the Platform Agent will not send debug level events to the server. For information on configuring the Platform Agent's LogDebug parameter, see "Logevent" on page 42. • If the logging application and Secure Logging Server are using custom certificates, make sure that the Logging Application Certificate has been signed with the Secure Logging Certificate. For more information, see Chapter 10, "Security and Non-Repudiation," on page 159.

Cause	Explanation/Solution
A component is misconfigured *continued*	If a single application is not logging events, • Make sure that the application has an Application object in one of the Secure Logging Server's supported Application containers. For more information on Application objects, see Chapter 6, “Managing Applications that Log to Nsure Audit,” on page 65. • Verify that the Application Identifier property in the Application object matches the Application Identifier stored in the logging application's certificate. For information on Application Identifiers, see Section 6.3, “Application Objects,” on page 67. • If you recently created an Application object, you must restart the logging server to load the Application object's configuration. For information on restarting the logging server, see Section G.3, “Secure Logging Server Startup Commands,” on page 236. • Make sure that the Application container where your Application object is located is included in the logging server's list of supported Application containers. For information on the logging server's Application container property, see “Logging Server Objects” on page 45.

11.1.3 Notifications Are Not Being Executed

The following table reviews the problems that might prevent notifications from being executed.

Cause	Explanation/Solution
A component is misconfigured.	There can be a misconfiguration on the Platform Agent or the Secure Logging Server that prevents events from being logged. `Solution` • Make sure that the Notification container where your Notification Filter object is located is included in the logging server's list of supported Notification containers. For information on the logging server's Notification container property, see “Logging Server Objects” on page 45. • Ensure that the Notification object has a notification channel. For more information on the Notification Channels property, see Section 8.3, “Notification Filters,” on page 102. • Check the console or the startup log to determine if the driver for the notification channel is being loaded. • Verify that the Notification Filter is configured to select the events you want to trigger the notification.

11.1.4 Volume Quickly Runs Out of Disk Space

The following table reviews the problems that might cause your data store volume to fill up quickly.

Cause	Explanation/Solution
You aren't cycling the data store.	Implement the available log-management functions. • Configure the File Channel object to roll and purge logs. For more information, see "File Channel Object" on page 76. • Configure the MySQL Channel object to expire the database. For more information, see "MySQL Channel Object" on page 87.
You are logging ubiquitous events.	Review your file system, eDirectory, and NetWare event settings on the NCP Server object and possibly disable some often-occurring events to avoid filling up your disk subsystem too quickly.
The data store volume is too small to accommodate your logging system traffic.	The space you need for your database depends on a number of factors which include, but are not limited to, how many events per second you are storing and how long you want to keep the data. To determine the required volume size for your logging system, log the desired events for about an hour during your host's peak utilization. Use the consumed disk space as a basis to calculate your logging system's required volume sizes. For some general statistics on MySQL data stores, a logging system that generates around 80 events per second with an average event size of 80 bytes will consume approximately 500 MB of disk space for the database table and 150 MB for the index in a 24-hour period.

11.1.5 The Host Running the Platform Agent is Running Out of Memory

The following table reviews the problems that might cause the host running the Platform Agent to run out of memory.

Cause	Explanation/Solution
The Disconnected Mode Cache has run out of disk space.	When the Secure Logging Server is not available, the Platform Agent's Logging Cache module writes incoming events to the Disconnected Mode Cache on disk. If the Disconnected Mode Cache runs out of disk space, the Logging Cache module falls back to memory.

Cause	Explanation/Solution
The Disconnected Mode Cache and data store are on the same volume.	If you are running the eDirectory Instrumentation or the NetWare Instrumentation on the same host as the Secure Logging Server, you should ensure that the Platform Agent's Disconnected Mode Cache and the Secure Logging Server's data store are not on the same volume. Otherwise, if the volume fills up, you will have a total logging failure. The Platform Agent will have no room for the Disconnected Mode Cache and the Secure Logging Server will have no place to log events.

11.1.6 Nsure Audit MySQL Returns a Cannot Open File Error

If the MySQL database returns error number 145, Cannot Open File, you might need to repair the MySQL database. See Appendix C, "Using MySQL with Nsure Audit," on page 211 for details on accessing the MySQL documentation to perform this procedure.

11.1.7 MySQL on Linux Returns a Socket Connection Error

If the MySQL database on Linux returns the following error message:

```
Can't connect to local MySQL server through socket '/temp/mysql.sock'
(2) 2002
```

Open /etc/my.cnf in a text editor and change the socket path to `/tmp/mysql.sock`.

11.1.8 Oracle on Linux Causes a Cannot Initialize the Logging Subsystem Error

If you are using Oracle, and you receive a "cannot initialize the logging subsystem" error when loading the Secure Logging Server on Linux, make sure you have correctly configured the database. For more information, see Section D.2, "Preparing the Oracle Database," on page 218.

11.1.9 Nsure Audit Events Sent During Initialization are Not Logged to the Data Store

During initialization, several events are sent to the Secure Logging Server by Nsure Audit. These events are not reported by platform agents, and are not intended to be logged to the datastore.

This can be confusing, as the Secure Logging Server console might show these events as having occured, but they are not logged to the data store. Only events logged your platform agents are logged to the datastore.

When troubleshooting your connection, make sure that you cause an event that will be reported by a platform agent.

11.1.10 Nsure Identity Manager 2 DR1 Update required to Use Nsure Audit 1.0.3

If you are using Nsure Identity Manager 2 without the DR1 update, do not upgrade to Nsure Audit 1.0.3 until you have applied the DR1 update. Without the DR1 update, your Identity Manager server might be unable to log events to Nsure Audit 1.0.3.

11.2 Verifying the Secure Logging Server Configuration

To verify the configuration used by the Secure Logging Server, review the Audit events logged to the data store.

The NAudit Instrumentation logs an event every time the Secure Logging Server loads a Channel, Notification, or Application object. It also logs an event each time a Channel driver fails to load or there is a bad Heartbeat or Notification configuration. Therefore, by reviewing your system's Audit the Auditor events, you can determine if your logging server is performing the way you expect.

For a complete list of Audit the Auditor events, see the NAudit LSC File (http://www.novell.com/documentation/lg/nsureaudit/html/naud_en.htm)

To review the Audit the Auditor events, you can query the events in iManager or Nsure Audit Report. For more information, see "Defining Queries in iManager" on page 120 or Section 9.2.9, "Working with Queries in Nsure Audit Report," on page 145.

11.3 Using the NetWare Instrumentation with Anti-Virus Products

The NetWare Instrumentation, AuditNW, must be loaded any anti-virus product or the server will appear to hang and will have to be hard-reset.

11.4 Uninstalling Novell Nsure Audit

1 Launch Auditext at the server console.

   1a On NetWare, enter `sys:\system\auditext.nlm`

   1b On Windows, enter `\program files\novell\nsure audit\auditext.exe`

   1c On Linux, enter `/opt/novell/naudit/auditext`

   1d On Solaris, enter `/opt/NOVLnaudit/auditext`

2 Specify your admin username and password.

3 Select Remove Schema Extensions, then press Enter.

   AuditExt removes the Logging Services container and all of its objects.

   IMPORTANT: Nsure Audit objects located outside the Logging Services container must be manually deleted.

4 Exit the AuditExt utility.

5 Remove the Nsure Audit Program files.

5a On NetWare, delete the following files and directories:

- sys:\system\naudit\
- auditagt.ncf
- nauditDS.nlm
- auditext.nlm
- auditNW.nlm
- auditsvr.ncf
- Disconnected Mode Cache directory
- lcache.nlm
- lengine.nlm
- channel drivers (lgd*.nlm)
- logevent.nlm
- logevent.cfg
- LSC files (*.lsc)
- mdb.nlm

For the location of these files and directories, see Section H.2, "Program Directories," on page 249.

5b On Windows:

- Click Start > Settings > Control Panel > Add or Remove Programs to launch the Add or Remove Programs wizard.
- Select Novell Nsure Audit and click Change/Remove.
- In the InstallShield Wizard, select Remove and click Next.
- In the confirmation dialog, click OK to remove Novell Nsure Audit and its installed components from your system.

5c On Linux, enter the following:

- `novell-AUDTlogserver-1.0.3-*.i586.rpm`
- `novell-AUDTplatformagent-1.0.3-*.i586.rpm`
- `novell-mdb-1.0-5.i386.rpm`
- `novell-AUDTedirinst-1.0.3-*.i586.rpm`

5d On Solaris, enter the following:

- `pkgrm NOVLaudin`
- `pkgrm NOVLaudit`
- `pkgrm NOVLaudpa`
- `pkgrm NOVLaudpl`
- `pkgrm NOVLmdb`

Event Structure

A

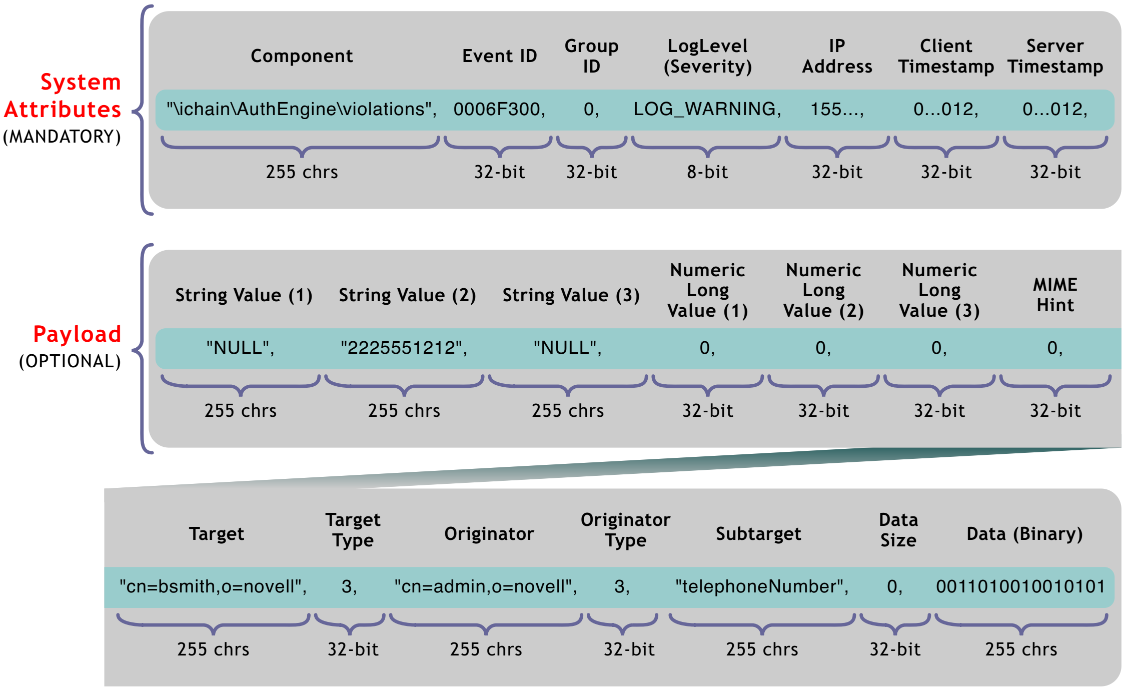
All events logged through Novell® Nsure Audit have a fixed set of fields. This section reviews event structure and its corollary information including event variables, application component strings, and log schema files.

A.1 Event Structure

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

The following diagram calls out the fields that make up a logged event. It also indicates the maximum size of each field.

Figure A-1 *Nsure Audit Event Structure*



Component	Event ID	Group ID	LogLevel (Severity)	IP Address	Client Timestamp	Server Timestamp
"\ichain\AuthEngine\violations",	0006F300,	0,	LOG_WARNING,	155...,	0...012,	0...012,
255 chrs	32-bit	32-bit	8-bit	32-bit	32-bit	32-bit

String Value (1)	String Value (2)	String Value (3)	Numeric Long Value (1)	Numeric Long Value (2)	Numeric Long Value (3)	MIME Hint
"NULL",	"2225551212",	"NULL",	0,	0,	0,	0,
255 chrs	255 chrs	255 chrs	32-bit	32-bit	32-bit	32-bit

Target	Target Type	Originator	Originator Type	Subtarget	Data Size	Data (Binary)
"cn=bsmith,o=novell",	3,	"cn=admin,o=novell",	3,	"telephoneNumber",	0,	0011010010010101
255 chrs	32-bit	255 chrs	32-bit	255 chrs	32-bit	255 chrs

The following table explains each event field.

Table A-1 *Nsure Audit Event Fields*

Event Field	Description
Component	The component string is formatted like a DOS pathname, with a backslash (\ ) separating component parts. For example: \eDirectory\Database\Lookup \iChain\Connection Manager\Authentication \NetMail\POP3\Authentication The first part of the component string is the Application Identifier. The Application Identifier is the string the logging application uses to identify itself to the logging server. The Application Identifier is stored in the application's certificate and Application object. When the Secure Logging Server authenticates an application's connection with the Platform Agent, it associates the Application Identifier with that connection. Thereafter, it automatically adds the Application Identifier to the component string for every event coming from that connection. For more information on application certificates and authentication, see Chapter 10, "Security and Non-Repudiation," on page 159.

Event Field	Description
Component continued	The subsequent portions of the component string are defined by the application. Typically, they identify modules within the application, types of events, etc. The intent of the component string is to facilitate queries across various products and events. For example, using wildcard characters, you can search for all iChain® violations (\ichain*\violations), all iChain events (\ichain*), or violations from every logging application (*\violations). You can also use the component string to filter events event chains. See Section 9.2.7, "Verifying Event Authenticity in Nsure Audit Report," on page 140. For a listing of the Nsure Audit, eDirectory™ and NetWare® component strings, see Section A.2, "Component Strings," on page 179.
EventID	The EventID is comprised of two elements: the HiWord and the LoWord. • The HiWord is the four-digit hex value assigned to the current application. All Application IDs are assigned through Novell Developer Support and are maintained in the Nsure Audit central registry. Before instrumenting a new application, developers should obtain an AppID through Novell Developer Support (http://developer.novell.com/devres/ss/resource.htm). • The LoWord is the AppEventID assigned by the person instrumenting the application. Typically, these values are assigned in ascending order. For more information, see the Nsure Audit SDK (http://developer.novell.com/ndk/naudit.htm).
GroupID	An ID that can be used to identify related events. For example, the NetMail® instrumentation of Nsure Audit uses this field to store the temporary filename assigned to each message as it passes through the message queue. By sorting on the Group ID, NetMail administrators can view all events that occurred as that particular message passed through the message queue.
Log Level (Severity)	The log level is an indicator of the severity of the reported event. • Emergency events cause the system to shut down. • Alert events require immediate attention. • Critical events might cause parts of the system to malfunction. • Error events are errors that can be handled by the system. • Warnings are negative events that do not represent a problem. • Notices are positive or negative events that an administrator can use to understand or improve the use and operation of the current system. • Info represents positive events of any importance. • Debug events are used by support technicians or engineers to debug the current system.
IP Address	The IP address of the Platform Agent that logged the event. By default, Nsure Audit stores IP address values in network byte order.
Client Timestamp	The time the Platform Agent received the event from the logging application.

Event Field	Description
ClientMS	The event count field. When a logging application makes a connection to the Platform Agent, the Secure Logging Server begins counting the events the come over that connection. The count begins at 0 for the initial event and increments by one for every event. If the logging application is restarted, the event count is reset to 0. Nsure Audit Report uses this field to determine how many events are missing if the event signatures are not to valid. For more information, see Section 9.2.7, “Verifying Event Authenticity in Nsure Audit Report,” on page 140.
Server Timestamp	The time the logging server received the event.
Text1	The value of this field depends upon the event. It can contain any text string up to 255 characters. The Text1 field is vital to the function of the CVR driver. The CVR driver looks in the event’s Text1 and Text2 fields to identify the defined attribute and object for a given policy. For more information, see “CVR Channel Driver” on page 72.
Text2	The value of this field depends upon the event. It can contain any text string up to 255 characters. The Text2 field is vital to the function of the CVR driver. The CVR driver looks in the event’s Text1 and Text2 fields to identify the defined attribute and object for a given policy. For more information, see “CVR Channel Driver” on page 72.
Text3	The value of this field depends upon the event. It can contain any text string up to 255 characters.
Value1	The value of this field depends upon the event. It can contain any numeric value up to 32 bits.
Value2	The value of this field depends upon the event. It can contain any numeric value up to 32 bits.
Value3	The value of this field depends upon the event. It can contain any numeric value up to 32 bits.
Mime hint	This field identifies the type of data contained in the Data field.
Target	This field captures the event target. All eDirectory events store the event’s object in the Target field.
Target Type	This field specifies which predefined format the target and originator are represented in. Defined values for this type are currently: • 0: None • 1: Slash Notation • 2: Dot Notation • 3: LDAP Notation
Originator	This field captures who or what caused the event to happen.

Event Field	Description
Originator Type	This field specifies which predefined format the target and originator are represented in. Defined values for this type are currently: • 0: None • 1: Slash Notation • 2: Dot Notation • 3: LDAP Notation
Sub Target	This field captures the sub-component of the target which was affected by the event. All eDirectory events store the event's attribute in the Sub Target field.
Data Size	This field identifies the size of the data contained in the Data field.
Data	The value of this field depends upon the event. The default size of this field is 3072 characters. You can configure the size of this field in the LogMaxBigData value in `logevent.cfg`. This value does not set the size of the Data field, but it does set the maximum size that the Platform Agent can log. For more information, see "Logevent" on page 42. The maximum size of the Data field is defined by the database where the data is logged. Thus the size varies for each database that is used. If the size of the data field logged by the Platform Agent exceeds the maximum size allowed by the database, the channel driver truncates the data in the Data field. If an event has more data than can be stored in the String and Numeric Value fields, it is possible to store up to 3 KB of binary data in the Data field.
Signature	The event signature. Nsure Audit digitally signs each event that is logged to the data store. To sign an event, the logging application or the Platform Agent hashes the event data and signs the hash with the Logging Application's private key. The signature is then stored as part of the event. This signature allows the auditor or investigator to determine if an event has been changed. If event chaining is enabled, each event's signature includes its own data as well as the signature from the previous event. This allows auditors to determine if an event has been deleted or if the sequence of events has been changed. Event chaining is enabled in the Platform Agent's configuration file, `logevent`. For information on configuring this option, see "Logevent" on page 42. For information on validating events in Nsure Audit Report, see Section 9.2.7, "Verifying Event Authenticity in Nsure Audit Report," on page 140.

A.2 Component Strings

The first part of all events logged to Nsure Audit is the component string. The intent of the component string is to facilitate queries across various products and events. For example, using wildcard characters, you can search for all iChain violations (\ichain*\violations), all iChain events (\ichain*), or violations from every logging application (*\violations).

Each logging application has its own set of components. The following sections list the components strings for Novell eDirectory, NetWare, and Audit the Auditor events. For information on other component strings, refer to the logging application's product documentation.

A.2.1 eDirectory Component Strings

The components for the eDirectory Instrumentation are as follows:

Table A-2 *eDirectory Component Strings*

Object	Attribute
Debug	Misc
Agent	Connection
Bindery	Schema
Partition	Replica
Meta	

These components are combined with the eDirectory Application Identifier (eDirInst) to form the component strings for all eDirectory events. For example,

eDirInst\Object
eDireInst\Replica

A.2.2 NetWare Component Strings

The components for the NetWare Instrumentation are as follows:

Table A-3 *NetWare Component Strings*

File	Directory
Trustee	Volume
Module	Network
Connection	Alert
Server	

These components are combined with the NetWare Application Identifier (NetWareInst) to form the component strings for all NetWare events. For example,

NetWareInst\File
NetWareInst\Network

A.2.3 Audit the Auditor Component Strings

The components for Nsure Audit Instrumentation are as follows:

***Table A-4** Audit the Auditor Component Strings*

Generic	License
Authentication	Log
Channel	Configuration
Heartbeat	Engine

These components are combined with the Nsure Audit Application Identifier (NsureAuditInst) to form the component strings for all Nsure Audit events. For example,

NsureAuditInst\Channel
NsureAuditInst\Log

A.3 Managing Event Data

Nsure Audit provides several variables that are used to determines what event fields are reported and how the event field data is displayed when logging to the File or Syslog channel in Translated Mode.

The event variables are constructed by specifying a dollar sign ($), followed by a two-character code representing the variable format (F) and event field value (V). For example:

```
$FV
```

The event field variable (V) references a specific field within a logged event. The format variable (F) determines how the data from the event field is displayed.

For example, event field R returns the IP address of the Platform Agent. Using different format variables, the IP address appears as follows:

$XR returns 1B043982

$NR returns 453261698

$iR returns 130.57.4.27

The Argument Builder simplifies the process of defining your event variables. It provides a graphical interface from which you can select which event fields you want to display in the translated log file and how you want the field data to display. Based on your selections, the Argument Builder defines the event schema using the event field and format variables.

The following sections review the *event_field* and *format* variables and how you can use the Argument Builder to define the event schema:

- Section A.3.1, “Event Field Variables (V),” on page 181
- Section A.3.2, “Format Variables (F),” on page 183
- Section A.3.3, “Using the Argument Builder to Define Event Schema,” on page 184

A.3.1 Event Field Variables (*V*)

IMPORTANT: Event variables are case sensitive and all variable strings must be preceded by a dollar sign ($).

Table A-5 *Event Field Variables*

Variable	Event Field
O	Component
I	EventID
G	GroupID
L	Log Level (Severity)
R	IP Address
C	Client Timestamp
A	Server Timestamp
S	Text1
	NOTE: To use the $S variable in the SMTP Channel object’s Recipient field, this value must be an e-mail address. For more information, see “SMTP Channel Object” on page 93.
T	Text2
	NOTE: To use the $T variable in the SMTP Channel object’s Recipient field, this value must be an e-mail address. For more information, see “SMTP Channel Object” on page 93.
F	Text3
	NOTE: To use the $F variable in the SMTP Channel object’s Recipient field, this value must be an e-mail address. For more information, see “SMTP Channel Object” on page 93.
1	Value1
2	Value2
3	Value3
M	Mime hint
U	Target
V	Target Type
Y	Sub Target
B	Originator
H	Originator Type
X	Data Size
D	Data

Variable	Event Field
SE	Description This variable returns the value of the Notification object’s Description field.The value is unique in that it is not provided by the logging application, but by the Notification object that directed the event to the current Channel driver. The Notification object’s description is sent with the event to the Channel driver. For more information on Notification object’s Description field, see Section 6.3, “Application Objects,” on page 67 or Section 8.4, “Heartbeat Objects,” on page 104.

A.3.2 Format Variables (*F*)

IMPORTANT: Format variables are case sensitive and all variable strings must be preceded by a dollar sign ($).

Table A-6 *Format Variables*

Variable	Format	Description
T	Local Time	Displays the time in the format defined on the local computer (UTC localized).
D	Local Date	Displays the date in the format defined on the local computer (UTC localized).
N	Numeric Format	Displays the current value in standard numeric format (32bit unsigned).
n	Signed Numeric Format	Displays the current value in standard numeric format (32bit signed). However, if the value is greater than 2 billion, it is displayed as a negative number.
S	String Format	Displays string values. **IMPORTANT:** This format variable can only be used with the O (Component), S (Text1), T (Text2), F (Text3), D (data), B (Originator), U (Target), and SE (Description) event variables.
X	Hexadecimal Number	Displays the current value in hexadecimal format.
R	RFC-822	Displays the current value in RFC-822 format. This variable is used to format time and date values. **NOTE:** RFC-822 is the Internet standard format for electronic mail message headers. All time values are expressed in UTC.
r	RFC-822 local	Displays the current value in RFC-822 format; however, the time and date values are expressed in local time rather than UTC.

Variable	Format	Description
I	IPv4 internet Address (network order)	Displays the current value as an IP address. This variable assumes the value is in network byte order. **NOTE:** By default, Nsure Audit stores IP address values in network byte order.
i	IPv4 Internet Address (host order)	Displays the current value as an IP address. This variable assumes the value is in host byte order.
B	Boolean Yes/No	If the value of the field is 0, this variable returns No. If the value is not 0, this variable returns Yes.
b	Boolean True/ False	If the value of the field is 0, this variable returns False. If the value is not 0, this variable returns True.

A.3.3 Using the Argument Builder to Define Event Schema

The Argument Builder is a tool that simplifies the process of defining the event schema. The event schema determines what event fields are reported and how the event field data is displayed when logging to the File or Syslog channel in Translated Mode.

The Argument Builder provides a graphical interface from which you can select which event fields you want to display in the translated log file and how you want the field data to display. Based on your selections, the Argument Builder defines the event schema using a series of event field and format variables. For information on the event schema syntax, see Section A.3, "Managing Event Data," on page 181.

To define an event's schema:

1 Open the *Query Options* task.

- **1a** Click the *Roles and Tasks* button on the iManager toolbar.
- **1b** In the Roles and Tasks view, expand the *Auditing and Logging* Role.
- **1c** Click the *Query Options* task.

2 In the Query Options page, click *Product Events*.

3 Open the event menu:

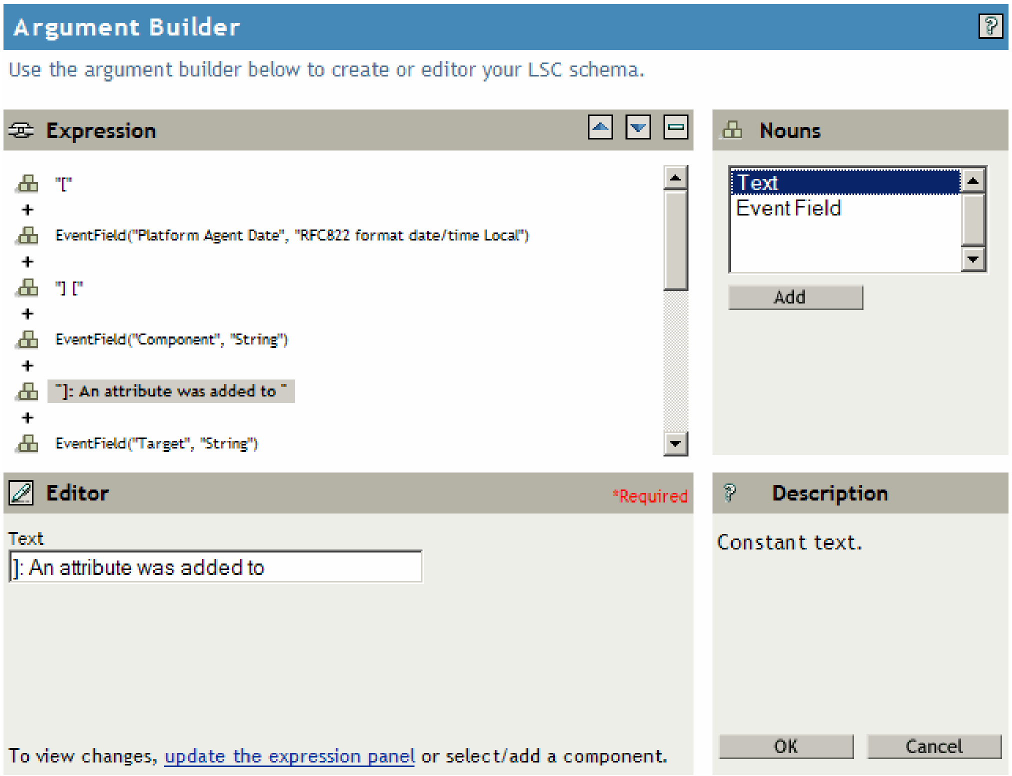
- In the Product Events page, select the logging application to which you want to add an event, click *New*, then click *OK* to confirm you want to create a new event.
- Click the plus icon ⊞ next to the product name to display the application's log events, select the event you want to modify, then click *Edit*.

4 In the event menu, click the *Argument Builder* button to open the Argument Builder.

5 To add a text field to the event schema:

- **5a** In the *Noun* frame, select *Text*, then click *Add*.
- **5b** In the *Editor* frame, specify the text string in the *Text* field.
- **5c** In the *Noun* frame, click *Add*.

The new text field appears in the *Expression* frame.



Argument Builder

Use the argument builder below to create or editor your LSC schema.

Expression

"["
+
EventField("Platform Agent Date", "RFC822 format date/time Local")
+
"] ["
+
EventField("Component", "String")
+
"]: An attribute was added to "
+
EventField("Target", "String")

Nouns

Text
Event Field

Editor *Required

Text:]: An attribute was added to

Description

Constant text.

To view changes, update the expression panel or select/add a component.

6 To add an event field to the event schema:

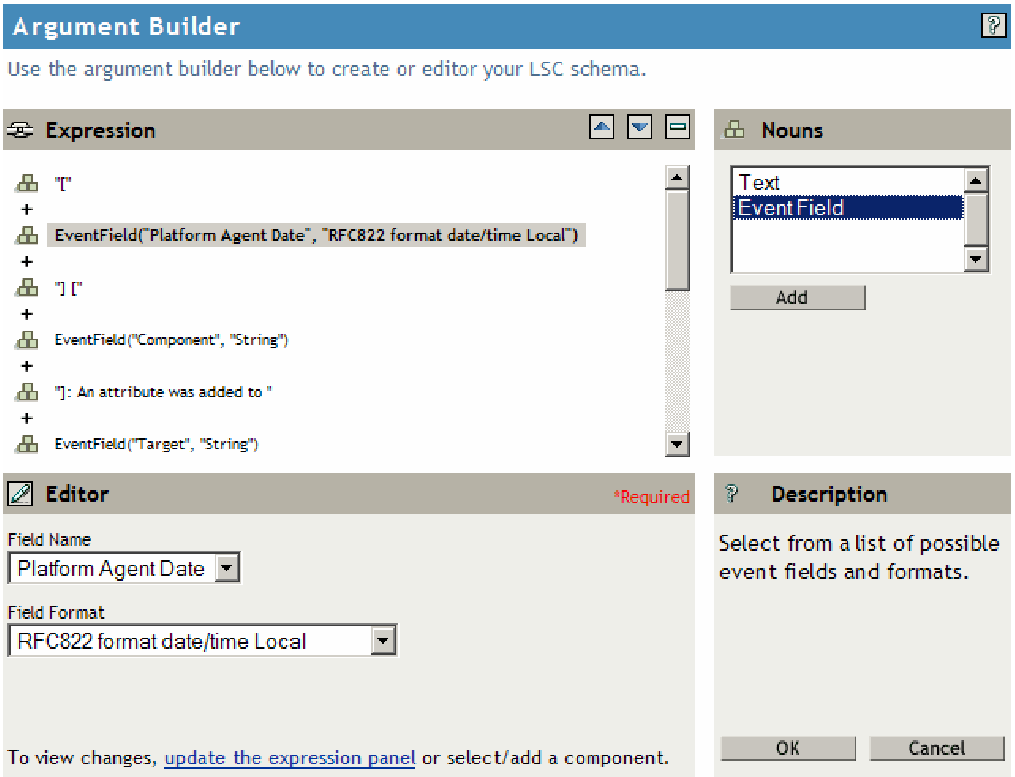
6a In the *Noun* frame, select *Event Field*, then click *Add*.

6b In the *Editor* frame, select an event field from the *Field Name* drop-down list.

6c Select the event field's associated format from the *Field Format* drop-down list.

6d In the *Noun* frame, click *Add*.

The new event field appears in the *Expression* frame.



"["
+
EventField("Platform Agent Date", "RFC822 format date/time Local")
+
"] ["
+
EventField("Component", "String")
+
"]: An attribute was added to "
+
EventField("Target", "String")

Nouns: Text, Event Field

Field Name: Platform Agent Date

Field Format: RFC822 format date/time Local

To view changes, update the expression panel or select/add a component.

Select from a list of possible event fields and formats.

7 To remove an item from the event schema:

7a In the *Expression* frame, select the text or event field you want to remove.

7b Click the *Remove Token* button ⊟ in the *Expression* frame.

The text or event field is removed from the *Expression* frame.

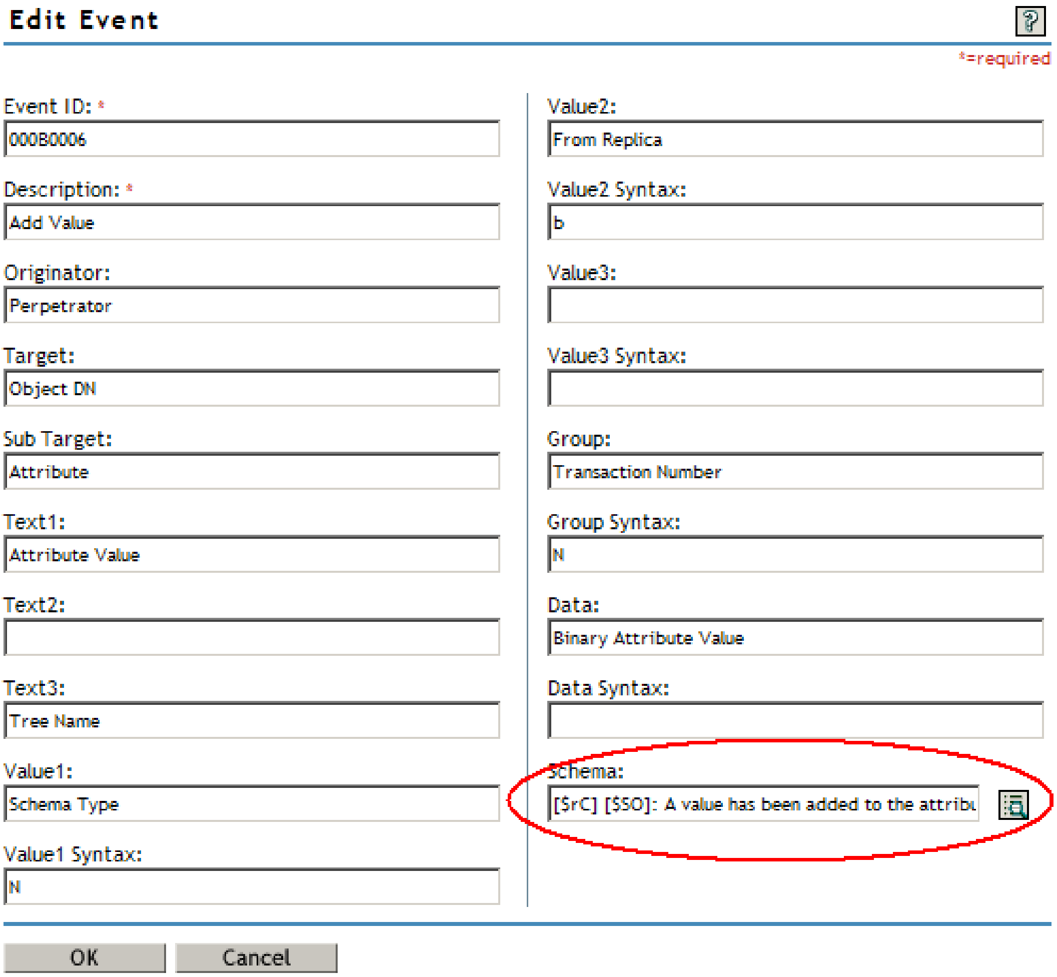
8 To modify the item order in the event schema:

8a In the *Expression* frame, select the text or event field you want to move.

8b Click the *Up* or *Down* ▼ buttons in the *Expression* frame to modify the item order.

9 When you have completed the event schema definition, click *OK* to save your changes.

iManager returns you to the event menu.



Edit Event

*=required

Event ID: * 000B0006

Description: * Add Value

Originator: Perpetrator

Target: Object DN

Sub Target: Attribute

Text1: Attribute Value

Text2:

Text3: Tree Name

Value1: Schema Type

Value1 Syntax: N

Value2: From Replica

Value2 Syntax: b

Value3:

Value3 Syntax:

Group: Transaction Number

Group Syntax: N

Data: Binary Attribute Value

Data Syntax:

Schema: [$rC] [$SO]: A value has been added to the attribu

The defined event schema appears in the *Schema* field as a series of event field and format variables. For information on the event schema syntax, see Section A.3, "Managing Event Data," on page 181.

A.4 Log Schema Files

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

Nsure Audit stores LSC files as attributes in their respective Application object. (English LSC files are stored under the NAuditAppSchemaEn attribute.) Typically, logging applications use the AuditExt utility to automatically create their associated Application object and to populate the Application object's log schema attribute; however, if you modify or localize an LSC file, you can manually add it to the Application object using the AuditExt utility. For more information on this procedure, see "Using AuditExt to Add LSC Files to Application Objects" on page 241.

- "How LSC Files Are Used" on page 188

- “Localized Log Schema Files” on page 189
- “Adding LSC Files to Application Objects” on page 189

A.4.1 How LSC Files Are Used

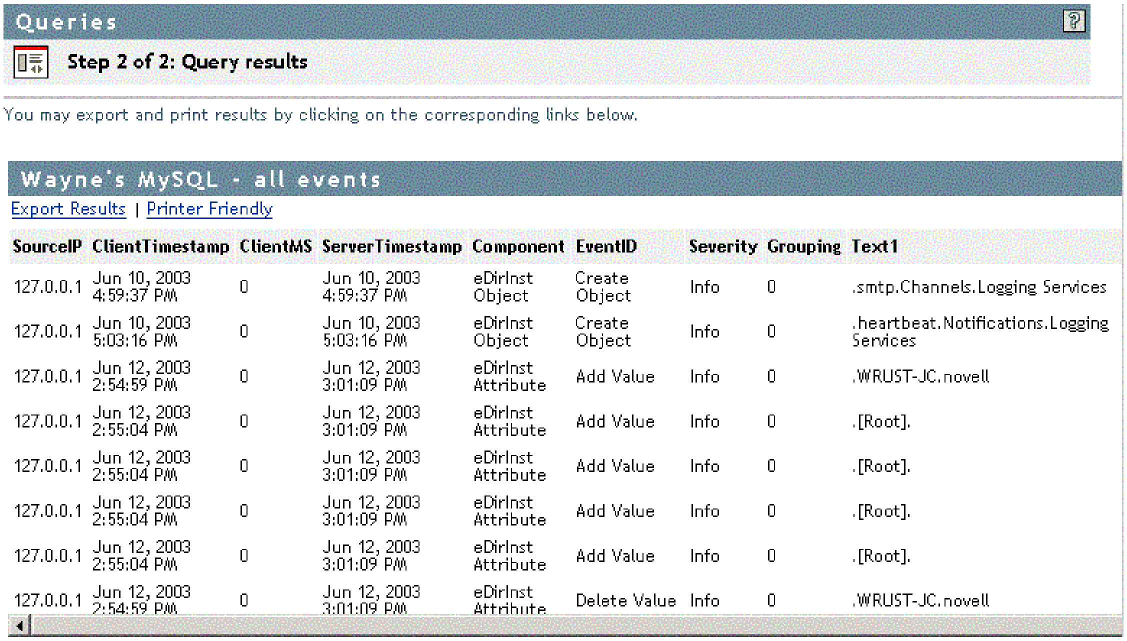
The information stored in the log schema files—specifically Event IDs, Group IDs, Text and Numeric field values—is useful in defining query statements, Notification Filters, and Heartbeat Notifications. For example, if you want to receive a notification anytime a server goes down, you must first look up the Event ID for the Server Down event in the NetWare log schema. You can then configure a Notification Filter that selects events with an Event ID of 000A0103.

The File and Syslog drivers use the log schemas to create localized, human-readable log files. At startup, the File and Syslog channel drivers load each application’s log schema in memory. If a logging application has multiple language versions of its log schema, the drivers load the schema for the language designated in their respective Channel objects. The File and Syslog channel drivers then reference the log schemas to write localized event descriptions to their log files.

NOTE: If the File and Syslog Channel objects reference the same language, the drivers independently load the log schema in their own memory. The only time the log schema is shared is between multiple instances of the same driver. For example, if you have two File channels configured to write Translated log files in English, the English log schema for each application is loaded only once. For more information, see “File Channel Driver” on page 75 and “Syslog Channel Driver” on page 98.

iManager and Nsure Audit Report also reference the log schemas. They use the Field titles defined in the log schemas to label column headings and event fields in the query results.

***Figure A-2** Query Result Column Headings*



Step 2 of 2: Query results

You may export and print results by clicking on the corresponding links below.

Wayne's MySQL - all events

Export Results | Printer Friendly

SourceIP	ClientTimestamp	ClientMS	ServerTimestamp	Component	EventID	Severity	Grouping	Text1
127.0.0.1	Jun 10, 2003 4:59:37 PM	0	Jun 10, 2003 4:59:37 PM	eDirInst Object	Create Object	Info	0	.smtp.Channels.Logging Services
127.0.0.1	Jun 10, 2003 5:03:16 PM	0	Jun 10, 2003 5:03:16 PM	eDirInst Object	Create Object	Info	0	.heartbeat.Notifications.Logging Services
127.0.0.1	Jun 12, 2003 2:54:59 PM	0	Jun 12, 2003 3:01:09 PM	eDirInst Attribute	Add Value	Info	0	.WRUST-JC.novell
127.0.0.1	Jun 12, 2003 2:55:04 PM	0	Jun 12, 2003 3:01:09 PM	eDirInst Attribute	Add Value	Info	0	.[Root].
127.0.0.1	Jun 12, 2003 2:55:04 PM	0	Jun 12, 2003 3:01:09 PM	eDirInst Attribute	Add Value	Info	0	.[Root].
127.0.0.1	Jun 12, 2003 2:55:04 PM	0	Jun 12, 2003 3:01:09 PM	eDirInst Attribute	Add Value	Info	0	.[Root].
127.0.0.1	Jun 12, 2003 2:55:04 PM	0	Jun 12, 2003 3:01:09 PM	eDirInst Attribute	Add Value	Info	0	.[Root].
127.0.0.1	Jun 12, 2003 2:54:59 PM	0	Jun 12, 2003 3:01:09 PM	eDirInst Attribute	Delete Value	Info	0	.WRUST-JC.novell

iManager and Nsure Audit Report manually import log schemas from the Application objects. For information on importing and viewing log schema files, see “Managing Product Events in iManager” on page 110 and Section 9.2.6, “Importing and Viewing Events in Nsure Audit Report,” on page 138.

A.4.2 Localized Log Schema Files

Although there is only one log schema for a given application, there can be many localized versions of the LSC file. Nsure Audit stores each language version of the LSC file as an attribute in its respective Application object. For example, English LSC files are stored under the NAuditAppSchemaEn attribute; German LSC files are stored under the NAuditAppSchemaDe attribute; and so forth.

IMPORTANT: Each language version of the LSC file must be added to the Application object before it can be used by the Syslog and File channel drivers to create localized log files.

Nsure Audit Report and iManager only support one language version of each log schema file at a time.

A.4.3 Adding LSC Files to Application Objects

During installation, logging applications use the AuditExt utility to automatically create their associated Application objects and to populate the Application objects’ log schema attribute. However, if you modify or localize an LSC file, you can manually add it to the Application object using the AuditExt utility at the server console.

To add a log schema to an Application object using AuditExt, enter the following command at the server console:

```
auditext -lsc -u:username -p:password "-a:Application_object" -
f:LSC_file -l:language
```

For example,

```
auditext -lsc -u:admin -p:argl "-a:eDirectory Instrumentation" "-
f:\temp\edir.lsc" -l:en
```

If the path to the LSC file contains spaces, enclose the path and the `-f` flag in quotation marks. For example, `"-f:c:/my files/myapp.lsc"`.

AuditExt requires that the first line of all LSC files is formatted as follows:

```
#^object_name^Application_ID^Application_Identifier^language_identifie
r
```

Each parameter is explained below.

Table A-7 AuditExt Parameters

Parameter	Description
object_name	The string that is used as the name of the Application object.

Parameter	Description
Application_ID	The four-digit hex value assigned to the current application. All Application IDs are assigned through Novell Developer Support and are maintained in the Nsure Audit central registry.
Application_Identifier	The name the logging application uses to identify itself to the logging server. The Application Identifier is stored in the application's certificate.
language_identifier	A two character code for the current LSC file's language. The following is a list of language codes: • EN = English • ES = Spanish • FR = French • DE = German • IT = Italian • PT = Portuguese • RU = Russian

If no path is given, AuditExt looks for the log schema files in the working directory of AuditExt. By default, schema log files are contained following directories:

Table A-8 *Schema Log File Directories*

Operating System	Directory
NetWare	`sys:\system\naudit\*.lsc`
Windows	`\program files\novell\nsure audit\logschema\*.lsc`
Linux	`/opt/novell/naudit/logschema/*.lsc`
Solaris	`/opt/NOVLnaudit/logschema/*.lsc`

Application Events

B

The following sections provide descriptions of the events logged for the following applications:

- Section B.1, "eDirectory Events," on page 191
- Section B.2, "File System Events," on page 206
- Section B.3, "NetWare Events," on page 207
- Section B.4, "Nsure Audit Events," on page 209

B.1 eDirectory Events

You configure which eDirectory™ events are logged in the NCP™ Server object's *Nsure Audit > eDirectory* page and in the eDirectory Instrumentation. For more information, see Section 5.2, "Configuring eDirectory, File System, and NetWare Events," on page 62.

IMPORTANT: eDirectory events such as login and logout are ubiquitous and can quickly fill your data store. Therefore, you should monitor your system's event traffic and configure your data store's expiration or roll policies accordingly. For information on the MySQL channel's expiration properties, see "MySQL Channel Object" on page 87. For information on configuring the File channel to purge or roll its log files, see "File Channel Object" on page 76.

Table B-1 *eDirectory Events*

EventID	Property Label	Description
000B0001	Create Object	A new eDirectory object has been created. To generate this event, create a new eDirectory object.
000B0002	Delete Object	An existing eDirectory object has been deleted. To generate this event, delete an existing eDirectory object.
000B0003	Rename Object	An existing eDirectory object has been renamed. To generate this event, rename an existing eDirectory object.
000B0004	Move Object (Source)	An existing eDirectory object has been moved. This event indicates that an object has been deleted from its source location. It is paired with a Move Object (Destination) event. To generate this event, move an existing eDirectory object from one eDirectory container to another.
000B0005	Move Object (Destination)	An existing eDirectory object has been moved. This event indicates that an object has been created in its destination location. It is paired with a Move Object (Source) event. To generate this event, move an existing eDirectory object from one eDirectory container to another.

EventID	Property Label	Description
000B0006	Add Value	A value has been assigned to an attribute of an eDirectory object. This event is common. To generate this event, add a value to a multi-valued attribute or assign a value to any other attribute.
000B0007	Delete Value	A value has been removed from an attribute of an eDirectory object. This event often occurs as a by-product of meta operations. This event is common. To generate this event, clear an attribute's value or remove a single value in a multi-valued attribute.
000B00081	Close Stream	A stream attribute has been closed.
000B0009	Delete Attribute	An attribute has been deleted from an eDirectory object. Some object, attribute, and value modification processes will remove an object entirely, then re-create it. A user cannot invoke this directly, but rather this occurs as a side-product of another user-initiated process. This event is rare. To generate this event, change the password for an existing eDirectory user object. The object's password attribute is deleted and re-created.
000B000A	Set Bindery Context	A server's bindery context has been set. To generate this event, type the following from a NetWare® console: `SET BINDERY CONTEXT=edir_context`
000B000B	Create Bindery Object	A bindery object has been created. Only a few applications continue to utilize bindery-only APIs. This might occur during SAP broadcasts on a server with IPX™ connectivity. Using bindery APIs and events is discouraged and should be discontinued. This event is rare.
000B000C	Delete Bindery Object	A bindery object has been deleted. Only a few applications continue to utilize bindery-only APIs. This might occur during SAP broadcasts on a server with IPX connectivity. Using bindery APIs and events is discouraged and should be discontinued. This event is rare.

EventID	Property Label	Description
000B000D	Delete Unused External Reference	An object's unused external reference has been deleted. This might occur during eDirectory's Backlink process. (It executes automatically every 25 hours.) This event indicates that an object possesses a reference to an original object that either no longer exists or has no valid reason to exist (that is, the original object has been deleted). This event is rare. To generate this event, simulate an eDirectory replica ring with poor communication among the replica's members. Delete an object, then manually invoke the Backlinker (`SET DSTRACE=ON, +BLINK, *B`).
000B000F[1]	Remote Server Down	A remote server has gone down.
000B0010[1]	Remote Connection Cleared	A remote connection has been cleared.
000B0011	Check SEV	An object's Security Equivalence Vector (SEV) has been checked. This event is generated when an object's SEV requires verification. This event is rare. To trigger this event, change the Security Equals To attribute on an existing eDirectory User object. Verification occurs immediately after changing the SEV.
000B0012	Update SEV	An object's Security Equivalence Vector (SEV) has been updated. This is often represented as the Security Equals To attribute in some object classes. This event is rare. To trigger this event, change the Security Equals To attribute on an existing eDirectory User object.
000B0013[1]	NCP Retry Expended	The maximum number of retries for an NCP request has been reached.
000B0014	DHost Module Loaded	A Loadable Module (DLM, NLM, XLM, etc.) has changed to the “loaded” state. This event type is used to notify dependent processes that one of its dependent modules has loaded. To trigger this event, load any NLM (`DS.NLM`, `EDIT.NLM`, and so forth) on NetWare. On Windows, start the Directory Services service (`DS.DLM`).
000B0015	DHost Module Unloaded	A loadable module (DLM, NLM™, XLM, etc.) has changed to the “unloaded” state. This event type is used to notify dependent processes that one of its dependent modules has unloaded. To trigger this event, unload any NLM (`DS.NLM`, `EDIT.NLM`, and so forth) on NetWare. On Windows, stop the Directory Services service (`DS.DLM`).

EventID	Property Label	Description
000B0201	Local Agent Opened	The local Directory agent has been opened. To trigger this event, perform a local database repair on a NetWare server, during which the local database is locked. This event is triggered when the repair concludes and the DB is unlocked for use.
000B0202	Local Agent Closed	The local Directory agent has been closed. To trigger this event, perform a local database repair on a NetWare server, during which the local database is locked. This event is triggered when the DB is locked.
000B0203[1]	Error Via Bindery	An error was returned via the Bindery.
000B0204[1]	DSA Bad Verb	An incorrect verb number was given in a DSAgent request.
000B0207[1]	Move (Subtree)	A container and its subordinate objects have been moved.
000B0208[1]	No Replica Pointer	A replica exists that has no replica pointer associated with it.
000B0209[1]	Inbound Sync End	Inbound synchronization has finished.
000B020A	Backlink SEV	A backlink operation has changed an object's SEV. This event should not occur under normal circumstances.
000B020B[1]	Backlink Operator	A backlink operation has changed an object's console operator privileges.
000B020C[1]	Delete Subtree	A container and its subordinate objects have been deleted.
000B020D[1]	New Master Sets	A new master replica has been designated.
000B020F	Partition State Change Request	A request has been made to change the state of an eDirectory partition. This event type is not used and should be excluded or ignored.
000B0210	Referral Created	A referral has been created. This might occur when part of a subtree is not maintained on the local server, as is the case with Subref replica types. To generate this event, delete the R/W replica from a subpartition after correctly partitioning a tree. A Subref replica is created and referrals are created to objects in that replica.
000B0211[1]	Update Class Definition	A schema class definition has been updated.
000B0212[1]	Update Attribute Definition	A schema attribute definition has been updated.
000B0213[1]	Lost Entry	eDirectory has encountered a lost entry. A lost entry is an entry for which updates are received, but no entry exists on the local server.

EventID	Property Label	Description
000B0214	Purge Entry Failed	An eDirectory object could not be purged. This is likely to occur if there are restrictions that prevent the proper removal of an unused object or its attribute. The Purge process is an automated background process eDirectory uses to maintain the health of a replica. It purges unused objects, processes object obituaries, and so forth. The Purge process can be initiated by the NDS Replica Synchronization background process, which is often invoked by various changes to eDirectory objects. This event is rare. To trigger this event, manually invoke the Purge process (`SET DSTRACE=ON, +J, *J`). Reproducing this event is difficult and requires a special set of circumstances that involve different object states on different replicas.
000B0215	Purge Start	An eDirectory Purge operation has begun. The Purge process is an automated background process that eDirectory uses to maintain the health of a replica. It purges unused objects, processes object obituaries, and so forth. The Purge process can be initiated by either the NDS Replica Synchronization background process or by an administrator using DSTrace. To trigger this event, manually invoke the Purge process using the Janitor (`SET DSTRACE=ON, +J, *J`).
000B0216	Purge End	An eDirectory Purge operation has concluded. The Purge process is an automated background process that eDirectory uses to maintain the health of a replica. It purges unused objects, processes object obituaries, and so forth. The Purge process can be initiated by either the NDS Replica Synchronization background process or by an administrator using DSTrace. To trigger this event, manually invoke the Purge process using the Janitor (`SET DSTRACE=ON, +J, *J`). This event is invoked when the process completes.
000B02171	FlatCleaner End	A Flatcleaner operation has completed.
000B02181	One Replica	A partition has been encountered that has only one replica. Novell recommends that each partition have at least three replicas for greater fault tolerance.
000B0219	Limber Done	A Limber operation has concluded. This process is responsible for maintaining high-level eDirectory tree connectivity. Every server in an eDirectory tree occasionally verifies its tree names and the addresses for other servers in the tree. The Limber process executes when eDirectory processes are restored after a reboot and when the server receives a Limber request from another server. It also runs when the eDirectory tree name changes and when an IP or IPX address change is reported. The Limber process invokes this event after completion. To trigger this event, manually invoke the Limber process (`SET DSTRACE=ON, +LIMBER, *L`).

EventID	Property Label	Description
000B021A1	Split Done	A Split Partition operation has completed.
000B021B	Outbound Sync (Server) Start	Outbound synchronization has begun. This occurs whenever replica synchronization is requested and the server submits its replica information. To trigger this event, manually invoke the Synchronization process (`SET DSTRACE=ON, +S, *S`).
000B021C	Outbound Sync (Server) End	Outbound synchronization has concluded. This occurs whenever replica synchronization is requested and the server submits its replica information. To trigger this event, manually invoke the Synchronization process (`SET DSTRACE=ON, +S, *S`). This event is invoked when the process completes.
000B021D	Sync Partition Start	The synchronization of a particular eDirectory partition has begun. To trigger this event, type the following at a NetWare console: `SET DSTRACE=*S`
000B021E	Sync Partition End	The synchronization of a particular eDirectory partition has concluded. To trigger this event, type the following at a NetWare console: `SET DSTRACE=*S` This event is invoked when the synchronization process completes.
000B021F1	Move Tree (Start)	A Move Subtree operation has started.
000B02201	Move Tree (End)	A Move Subtree operation has finished.
000B0221	Recertified Public Key	An eDirectory NCP Server object's public key has been regenerated. This event is meant for use by internal Novell support tools only. It should be excluded or ignored. This event is rare.
000B02221	Generated CA Keys	Certificate of Authority keys have been generated.
000B0223	Join Done	A Join Partitions operation has completed. This occurs when you merge an eDirectory partition with its parent. This event is rare. To trigger this event, merge an eDirectory partition. The status of that partition and its parent reside in a join state until the merge completes.

EventID	Property Label	Description
000B0224	Partition Locked	An eDirectory partition has been locked. A partition locks prior to a major partition operation, including creating a new partition (and splitting the parent partition), merging a partition into its parent, and changing a replica's type. This event is rare. To trigger this event, select a subcontainer within a partitioned container. Create a new partition using the subcontainer as its root.
000B0225	Partition Unlocked	An eDirectory partition has been unlocked. A partition unlocks after a major partition operation, including creating a new partition (and splitting the parent partition), merging a partition into its parent, and changing a replica's type. This event is rare. To trigger this event, select a subcontainer within a partitioned container. Create a new partition using the subcontainer as its root. This event is invoked after the operation completes.
000B0226	Schema Synchronized	An eDirectory schema has been successfully synchronized. This occurs whenever schema changes are propagated throughout a tree or when schema synchronization is forced using DSTrace or DSRepair. To trigger this event, manually invoke a schema synchronization (`SET DSTRACE=ON, +S, *SS`).
000B0227	Name Collision	A name collision has occurred between two eDirectory objects. This signifies a problem in eDirectory that should be corrected using DSRepair. Name collisions can be a by-product of synchronization problems. eDirectory invokes this event if a problem is detected during its automated synchronization routines. This event is rare.
000B0228	NLM Loaded	A NetWare Loadable Module (NLM) has been loaded. This is an internal Novell event that indicates eDirectory (DS) was loaded through another support module. This event can be excluded or ignored. To trigger this event, load a NetWare module such as DSRepair. This event is triggered immediately after the module is loaded.
000B022B1	Lumber Done	A Lumber operation has completed.
000B022C	Backlink Procedure Done	A backlink process has completed. This is triggered by eDirectory's internal automated maintenance routines. To trigger this event, manually invoke the backlink process (`SET DSTRACE=ON, +BLINK, *B`).

EventID	Property Label	Description
000B022D	Server Renamed	A server has been renamed. This occurs when the name of an NCP server changes in eDirectory. To trigger this event, change a NetWare server's name using the `autoexec.ncf` file, then reboot the server.
000B022E	Synthetic Time Issued	An eDirectory server has issued synthetic time for the purposes of synchronization. This occurs when configured time sources are no longer functional. To trigger this event, configure an eDirectory server to use time sources that do not exist.
000B022F	Server Address Changed	A server's address has changed. This event occurs when eDirectory updates an NCP Server object with the new address. This update is triggered by eDirectory's Limber process. An administrator can change the IP address on a NetWare server and reboot it. Limber executes after the reboot and triggers this event. This event is rare. To trigger this event, change a NetWare server's IP address and reboot. Prior to rebooting, the NetWare server must be configured to auto-load the Nsure Audit eDirectory Instrumentation.
000B0230	DSA Read	A read operation has been performed on an eDirectory object. Because the majority of eDirectory actions read object data, this event is very common. To trigger this event, open an eDirectory object and view any of its attributes.
000B0301	Login	A user has successfully logged in to eDirectory. This is without regard to a particular server. To trigger this event, log in to eDirectory using a valid account and credentials.
000B0302	Change Password	A user's password has been changed. To trigger this event, assign a new password to an existing eDirectory user object.
000B0303	Logout	A user has logged out from eDirectory. This is without regard to a particular server. To trigger this event, log in to eDirectory using a valid account and credentials. After successfully authenticating, log out of eDirectory.
000B03041	Added Replica	A replica of a partition has been added to a server.
000B03051	Removed Replica	A replica of a partition has been removed from a server.
000B03061	Split Partition	A partition has been split.
000B03071	Join Partitions	A parent partition has been joined with a child partition.

EventID	Property Label	Description
000B0308	Changed Replica Type	A replica's type has been changed. To trigger this event, promote a R/W replica to Master In a healthy replica ring.
000B0309	Remove Object	An object has been deleted from a container. To trigger this event, delete an eDirectory object within any container object.
000B030A	Abort Partition Operation	A request has been made to abort a partition operation. To trigger this event, initiate any partition operation (merge, new, split, and so forth), then immediately attempt to cancel the process.
000B030B1	Received Replica Updates	A replica has received an update during synchronization.
000B030C1	Repaired Timestamps	A replica's time stamps have been repaired.
000B030D1	Send Replica Updates	A replica has sent an update during synchronization.
000B030E1	Verify Password	A password has been verified.
000B030F	Backup Object	An eDirectory object has been backed up. Only eDirectory-aware backup applications are known to invoke this event. To trigger this event, configure your backup software to back up eDirectory.
000B0310	Restore Object	An eDirectory object has been restored. Only eDirectory-aware backup applications are known to invoke this event. To trigger this event, configure your backup software to restore objects from eDirectory.
000B0311	Define Attribute	An attribute definition has been added to the schema. To trigger this event, update an eDirectory tree's schema with new attribute definitions for existing objects.
000B0312	Remove Attribute	An attribute definition has been removed from the schema. This occurs only when eDirectory has been instructed to remove an attribute definition from its schema. This event can be invoked using schema modification tools that require root-level eDirectory credentials. This event is very rare. To trigger this event, you must use a schema modification utility or application to delete a particular object's attribute definition.
000B0313	Remove Class	A class definition has been removed from the schema. To trigger this event, use a schema editor to remove a class definition.

EventID	Property Label	Description
000B0314	Define Class	A class definition has been added to the schema. To trigger this event, update an eDirectory tree's schema with new object class definitions.
000B0315	Modify Class	An eDirectory class definition has been modified.
000B0316[1]	DS Counters Reset	The internal NDS® counters have been reset.
000B0317	Remove Assoc. Directory	A directory associated with an entry has been removed. This is triggered almost exclusively by queue-based objects (such as Print Queues) where the object relies upon the existence of a server or directory path on a particular file server. (This does not apply to a User object's Home Directory attribute.) This event signifies that a server has been notified to delete the server or directory path. This event is rare. To trigger this event, create a Print Queue object in an environment where multiple servers host replicas within a particular replica ring. Ensure that the Print Queue has a valid Queue Directory assignment. Delete the Print Queue. This event occurs if the eDirectory server handling the request differs from the server that hosts the queue directory.
000B0318	Compare Attribute Value	An eDirectory object's attribute has been subjected to a Compare operation. Many meta-operations incorporate value comparisons. To trigger this event, add a replica to an eDirectory server. Several comparison checks are performed before and after this Partition operation.
000B0319	DS Stream Opened	An eDirectory object’s stream attribute has been opened. To trigger this event, attempt to log in to eDirectory using a valid eDirectory account and credentials. **NOTE:** The Login Script attribute is a stream attribute.
000B031A	List Subordinates	A request has been made to list a container object’s subordinate entries. To trigger this event, expand a container object to view its subordinate objects.
000B031B[1]	List Containable Classes	A List Containable Classes operation has been performed on an entry.
000B031C	Inspected Entry	An Inspect Entry operation has been performed on an eDirectory object. This event type is exclusively intended for older versions of Novell DSRepair. This event is rare.

EventID	Property Label	Description
000B031D	Resent Entry	A Resend Entry operation has been performed on an eDirectory object. A flag on an object has been changed to instruct eDirectory to request that the object be re-sent from an authoritative source. This event type might be invoked by internal Novell support tools. It should be excluded or ignored. This event is rare.
000B031E1	Mutate Entry	A Mutate Entry operation has been performed on an entry.
000B031F	Merged Entries	One eDirectory object has been merged with another. This event type is invoked almost exclusively by internal Novell support tools. It can be excluded or ignored. This event is rare.
000B0320	Merge Trees	This event signifies that two eDirectory trees have been merged. Because of the significant implications of merging trees, this event should rarely occur, if ever. To trigger this event, use iManager to select and merge different eDirectory trees.
000B03211	Create Subref	A subordinate reference has been created.
000B0322	List Partitions	A request has been made to enumerate partitions. To trigger this event, use iManager to locate an NCP server and list all partitions for which that server hosts a replica.
000B03231	Read Attribute	An entry's attributes have been read.
000B0324	Read References	The references on an eDirectory object have been read. Although this event is designed for backup applications and internal Novell support tools, it can be used by anyone programming against the eDirectory API. It can be excluded or ignored. This event is rare.
000B0325	Updated Replica	An eDirectory replica has been updated. To trigger this event, use DSRepair to force the replicas to synchronize. Although there are separate events to designate the beginning and end of replica synchronization, this event indicates that a replica has been updated, not the process itself.
000B0326	Start Update Replica	An Update Replica operation has begun on a partition replica. In a replica ring with multiple replicas, one replica notifies another that an update is pending. The target server then invokes this event at the beginning of its synchronization cycle. To trigger this event, load DSRepair on a Master replica holder. Select a partition, then select *Synchronize the replica on all servers*. Non-Master replicas invoke this event.

EventID	Property Label	Description
000B0327	End Update Replica	An Update Replica operation has concluded on a partition replica. To trigger this event, use DSRepair to force the replicas to synchronize. This event is invoked upon completion.
000B0328[1]	Sync Partition	A Synchronize Partition operation has been performed on a partition replica.
000B0329[1]	Synchronized Schema	The schema has been synchronized.
000B032A[1]	Create Backlink	A backlink has been created.
000B032B[1]	Check Console Operator	An object has been checked for Console Operator rights.
000B032C	Change Tree Name	An eDirectory tree's name has been changed. To trigger this event, rename an existing eDirectory tree.
000B032D	Start Join	A Join Partitions operation has begun. This occurs after someone with administrative rights attempts to merge one eDirectory partition with its parent. Because Partition operations are meant for maintenance and redesign tasks, this should occur infrequently. To trigger this event, select an eDirectory partition and merge it with its parent. The status of that partition and its parent is in a Join state until the merge completes.
000B032E	Abort Join	A request has been made to abort a Partition Join operation. To trigger this event, merge one partition into its parent, then immediately attempt to cancel the process.
000B032F	Update Schema	An eDirectory schema has been updated. To trigger this event, use DSRepair on a non-Master replica holder to Request schema from Tree. This event indicates that a replica has been updated, not the process itself.
000B0330	Start Update Schema	An Update Schema operation has begun within a partition eDirectory tree. The target server then invokes this event at the beginning of its schema synchronization cycle. To trigger this event, load DSRepair on a Master replica holder. Select *Global Schema Options*, then select *Request schema from Tree*. Non-Master replicas invoke this event.
000B0331	End Update Schema	An Update Schema operation has concluded for an eDirectory tree. To trigger this event, use DSRepair to request the schema from the eDirectory tree for which the current server holds a replica. This event is invoked upon completion.
000B0332[1]	Move Tree (Source)	A Move Tree operation has been performed.

EventID	Property Label	Description
000B0333	DS Reloaded	eDirectory has been reloaded. **IMPORTANT:** This event is implemented only in eDirectory for NetWare. To trigger this event, type the following at a NetWare console: `SET DSTRACE=*`
000B0334[1]	Add Property	An attribute (property) has been added to an object.
000B0335[1]	Delete Property	An attribute (property) has been removed from an object.
000B0336	Add Group Member	A member has been added to a Group object's Members list. To trigger this event, modify an existing eDirectory Group object by adding a member to the Group object's Members list.
000B0337	Delete Group Member	A member has been removed from a Group object's Members list. To trigger this event, modify an existing eDirectory Group object by removing a member from the Group object's Members list.
000B0338	Change Property Security	Security for a property on a bindery object has been changed. Only a few applications continue to utilize bindery-only APIs. This might occur during SAP broadcasts on a server with IPX connectivity. Using bindery APIs and events is discouraged and should be discontinued. This event is rare.
000B0339[1]	Change Object Security	A bindery object's security has been changed.
000B033A[1]	Read Object Info	A Read Object Info operation has been performed on an object.
000B033C	Search	An eDirectory search operation has been performed. To trigger this event, use iManager to search for all objects of a particular class type.
000B033D	Partition State Changed	The state of an eDirectory partition has successfully changed. Various partition states include On, Transition On, Move, and Join. To trigger this event, merge an eDirectory partition with its parent. The status of that partition and its parent changes to Join until the merge completes.
000B033E[1]	Remove Backlink	A backlink has been removed.
000B033F[1]	Low Level Join	A low-level join has been performed.
000B03401	Create Namebase	The Directory namebase has been created.

EventID	Property Label	Description
000B0341	Change Security Equals	An eDirectory object's Security Equals To attribute has been changed. This signifies either an addition or reduction of rights for a particular object. To trigger this event, add or remove entries from an existing object's Security Equals To attribute.
000B0342	CRC Failure	A CRC failure was encountered during the re-assembly of NCP packets. The NCP packets were determined to be fragmented. This can be caused by poorly-written NIC drivers, a hardware-layer network communications issue, or in the unlikely event of a Denial of Service attack. The easiest scenario to trigger this event requires the use of a packet generation utility that can simulate fragmented NCP packets.
000B0343	Add Entry	An eDirectory object has been created beneath a container. To trigger this event, create an eDirectory object beneath any container object.
000B0344	Modify Object	An eDirectory object's attribute has been modified. To trigger this event, change the value of an object's attribute such as the First name of a User object.
000B0345	Open Bindery	The local bindery has been opened. This is often the result of an attempt to close a server's local eDirectory database for maintenance purposes. To trigger this event, perform a database repair on an eDirectory server using DSRepair. The bindery opens after performing a repair.
000B0346	Close Bindery	The local bindery has been closed. This is often the result of an attempt to close a server's local eDirectory database for maintenance purposes. To trigger this event, perform a database repair on an eDirectory server using DSRepair. The bindery closes prior to performing a repair.
000B0347	Connection State Changed	An eDirectory server has detected a change in an object's connection state. This reflects changes in a connection to a server, that is adding or terminating a connection. It can be generated if eDirectory is unloaded, terminating all local connections. This event is not implemented in eDirectory for NetWare. To trigger this event, log in to eDirectory using a valid account and credentials. (Disconnect and log out first if necessary.) The resulting events indicate the addition of a connection to the target eDirectory server.
000B03481	New Schema Epoch	A new schema epoch has been declared.
000B03491	Modify RDN	A Modify RDN operation has been performed.

EventID	Property Label	Description
000B034A1	Created Orphaned Partition	A Create Orphan Partition operation has been performed.
000B034B1	Removed Orphaned Partition	A Remove Orphan Partition operation has been performed.
000B034C1	Linked Orphaned Partition	A Link Orphan Partition operation has been performed.
000B034D1	Unlinked Orphaned Partition	An Unlink Orphan Partition operation has been performed.
000B034E1	EntryIDs Swapped	A Swap Entry ID operation has been performed.
000B034F1	Low Level Split	A low-level partition split has been performed.
000B0351	Allow Login	A user has been allowed to log in to eDirectory. The Login Time Restrictions attribute (on some eDirectory object types) is checked at the top (hh:00) and bottom (hh:30) of every hour to validate all authenticated connections. An object's connection is invalidated if that object is not configured to have access during the next 30-minute segment. **IMPORTANT:** This event is implemented only in eDirectory for NetWare. A NetWare server validates its authenticated connections every 30 minutes. This event is triggered once for each authenticated connection per half hour.
000B0352	DS Stream Closed	The stream attribute of an eDirectory object has been closed. To trigger this event, log into eDirectory using a valid eDirectory account and credentials. **NOTE:** The Login Script attribute is a stream attribute.
000B03531	Move Tree (Target)	A Move Tree operation has been performed.
000B0354	ACL Changed	The access control list of an eDirectory object has changed. It might have been assigned a more or less restrictive ACL either directly or through inheritance. To trigger this event, select an eDirectory object and modify its trustees list. Either add or remove trustees or modify the rights assigned to a particular trustee.
000B0355	Login Enabled	An eDirectory object's login state has been enabled, allowing a successful login. To trigger this event, clear the Account Disabled attribute on an eDirectory user.
000B0356	Login Disabled	An eDirectory object's login state has been disabled, prohibiting a successful log in. To trigger this event, select the Account Disabled attribute in an eDirectory User object.

EventID	Property Label	Description
000B0357	Intruder Detected	An eDirectory object's attempts to login have failed in a way that a system intrusion is suspected.
000B0358	Login Failed	A user's attempt to log in to eDirectory has failed. This is often caused by using incorrect login credentials.

1Although methods exist to trigger this event, it is not expected to occur in daily operations using current products. This event is more likely to occur with older products or when Novell eDirectory Technical support is used to resolve severe eDirectory issues.

B.2 File System Events

You configure which file system events are logged in the NCP Server object's *Nsure Audit > NetWare* page and in the NetWare Instrumentation. For more information, see "Configuring eDirectory, File System, and NetWare Events" on page 62.

This section reviews the file system events that Nsure Audit can log.

NOTE: The events are listed in the order they appear in the NCP Server object's *Nsure Audit > Filesystem* page.

Table B-2 *File System Events*

EventID	Property Label	Description
000A0001	File Delete	An existing file has been deleted. To generate this event, delete an existing file.
000A0002	File Open	An existing file has been opened for access. This event is very common. To generate this event, edit an existing file.
000A0003	File Create	A new file has been created. This event is not applicable to NSS. To generate this event, attempt to edit a non-existent file or use an editor to save a new file.
000A0004	File Create & Open	A new file has been created and immediately opened for access. To generate this event, attempt to edit a non-existent file or use an editor to save a new file.
000A0005	File Rename	An existing file or directory has been either renamed or moved to another location. To generate this event, rename a file or directory.

EventID	Property Label	Description
000A0006	File Close	A previously open file has since been closed. This event is very common. To generate this event, open or create a file. Close the file or exit the application used to open or create the file.
000A0007	Directory Create	A new directory has been created. To generate this event, create a new directory.
000A0008	Directory Remove	An existing directory has been deleted. To generate this event, delete an existing directory.
000A0009	Directory Modified	A file or directory has been modified. To generate an instance of this event, enable the Read-Only attribute for an existing file or directory.
000A000A	File Salvaged	A previously deleted file has been salvaged (undeleted). To generate this event, use the Salvage feature of the Novell Client™ to find and restore a file.
000A000B	File Purged	A previously-deleted file has been purged (permanently deleted). To generate this event, use the Purge feature of the Novell Client to find and permanently delete a file.
000A000E	DOS Attributes Modified	The generic DOS information for a file system entry has been modified. This event is not applicable to NSS. To generate an instance of this event, use the Novell Filer utility, `filer.exe`, to change the read-only flag on a file.
000A000F	Trustee Added	A trustee has been added to a file or directory. To generate this event, locate an existing file or directory and assign an eDirectory object to be a trustee.
000A0010	Trustee Removed	A trustee has been removed from a file or directory. To generate this event, locate an existing file or directory with at least one assigned trustee. Remove a trustee assignment from that file or directory.
000A0011	Trustee Modified	An existing file or directory trustee has been modified. To generate this event, locate an existing file or directory with at least one assigned trustee. Change the rights granted to one of those trustees.

B.3 NetWare Events

You configure which NetWare events are logged in the NCP Server object's *Nsure Audit* > *NetWare* page and in the NetWare Instrumentation. For more information, see "Configuring eDirectory, File System, and NetWare Events" on page 62.

This section reviews the NetWare events that Nsure Audit can log.

NOTE: The events are listed in the order they appear in the NCP Server object's *Nsure Audit* > *NetWare* page.

Table B-3 *NetWare Events*

EventID	Property Label	Description
000A0101	Volume Mounted	A volume has been successfully mounted. To generate this event, mount a previously unmounted volume.
000A0102	Volume Dismounted	A volume has been successfully dismounted. To generate this event, dismount a previously mounted volume.
000A0103	Server Down	A NetWare server has been instructed to shut down or reboot. To generate this event, down or restart your NetWare server.
000A0104	Module Loaded	A NetWare Loadable Module (NLM) has been loaded. This might occur manually (command line) or via another module. To generate an instance of this event, type `EDIT` at a NetWare console.
000A0105	Module Unloaded	A NetWare Loadable Module (NLM) has been unloaded. This might occur manually (command line) or via another module. To generate this event, type `UNLOAD` *`module_name`* at a NetWare console.
000A0106	Connection Cleared	An object's connection has been cleared. This might occur during log outs, connection inactivity, or server shutdowns. This is a common event. To generate this event, forcefully clear or delete an existing connection from the Connection table in NetWare's MONITOR module.
000A0107	Login (Netware)	A user has obtained a connection to a NetWare server and authenticated that connection. This cites eDirectory authentication to a specific NetWare server. To generate this event, log in to eDirectory using a valid account and credentials. When logging in, specify the server where you want to connect.
000A0108	Protocol Bind	A communications protocol has been successfully bound to an existing NIC adapter. To generate this event use INETCFG to enable or bind a protocol to an existing adapter.
000A0109	Protocol Unbind	A communications protocol has been unbound from an existing NIC adapter. To generate this event, use INETCFG to disable an existing bound protocol.

EventID	Property Label	Description
000A010A	NetWare Alert	NetWare has received an alert message. Time synchronization, memory allocation, and LAN connection problems might invoke this event type. These messages are synonymous with those seen on the NetWare console, such as "*date time* DS-*ver* / Directory Services: Local database is open." To generate this event, unplug the LAN cable from the NIC port on a NetWare server. (Connection state messages should appear.) You can also attempt to unload DS from the console.
000A010B	Logout (Netware)	A user has logged-out from a NetWare server. This cites an eDirectory logout event from a particular NetWare server. To generate this event, log in to eDirectory using a valid account, credentials, and server name. After successfully authenticating, log out from eDirectory.

B.4 Nsure Audit Events

The NAudit Instrumentation logs an event every time the Secure Logging Server loads a Channel, Notification, or Application object. It also logs an event each time a Channel driver fails to load or if there is a bad Heartbeat or Notification configuration. Therefore, by reviewing your system's Audit the Auditor events, you can determine if your logging server is performing the way you expect.

This section reviews the Nsure Audit events that the Nsure Audit Instrumentation can log:

Table B-4 *Nsure Audit Events*

EventID	Property Label	Description
10001	Heartbeat Generated	A Heartbeat event was generated.
10002	License Warning	This event was used in past versions of Nsure^{TM} Audit. It is no longer implemented.
10003	Application Container Used	
10004	Application Allowed	A product instrumentation successfully authenticated with the Secure Logging Server.
10005	Application Failed	A product instrumentation failed to authenticate with the Secure Logging Server. The Secure Logging Server does not log events from any application that fails to authenticate with the server.
10006	Channel Loaded	The designated channel driver successfully loaded on the Secure Logging Server.
10007	Driver Failed	The designated channel driver failed to load on the Secure Logging Server. Therefore, the designated channel is not currently available.

EventID	Property Label	Description
10008	Default Log Channel	Designates the default log channel. This channel functions as the central data store; all system events are logged to this channel.
10009	Log Channel Failed	The designated Channel object failed to load on the Secure Logging Server, so the designated Channel is not currently available.
10010	Notification Loaded	The designated Notification object was successfully loaded on the Secure Logging Server.
10011	Bad Notification	An incorrectly configured notification rule was detected on the designated Notification object.
10012	Heartbeat Loaded	The designated Heartbeat Notification object was successfully loaded on the Secure Logging Server.
10013	Bad Heartbeat	An incorrectly configured heartbeat rule was detected on the designated Heartbeat Notification object.
0001000A	Out of Memory	The central data store is out of space, so the Nsure Audit system has shut down.
0001000B	Server Unload Attempt	An attempt was made to unload the Secure Logging Server.
0001000C	Server Unloaded	The Secure Logging Server was successfully unloaded, so the Nsure Audit system is not currently functioning.
0001000E	Channel Container Used	
0001000F	Notification Container Used	

Using MySQL with Nsure Audit

C

This section provides basic information that helps you use MySQL with Novell® Nsure™ Audit.

Refer to your MySQL manual for information on the following:

- How to install MySQL
- How to create users and grant access rights within MySQL
- How to optimize MySQL
- MySQL Query syntax
- MySQL Miscellaneous Functions
- Optimizing Your Queries
- Repairing a MySQL database

NOTE: All of the current MySQL distributions are available from the Downloads page (http://www.mysql.com/downloads/index.html) at the MySQL Web site. The *MySQL Reference Manual* is available from the Documentation page (http://www.mysql.com/documentation/index.html).

C.1 Channel Requirements

Before you set up a MySQL database for use with Nsure Audit, be aware of the following requirements:

- Nsure Audit supports MySQL 4.0. and 4.1
- MySQL can be on a different server than the Secure Logging Server
- Nsure Audit utilizes its own MySQL library to connect to a MySQL server

C.2 Installing MySQL

The following instructions provide detailed information on installing MySQL on all supported platforms:

- **NetWare 6.5:** Installing MySQL on NetWare® (http://dev.mysql.com/doc/mysql/en/NetWare_installation.html)
- **Windows Server 2000 and 2003:** Installing MySQL on Windows (http://dev.mysql.com/doc/mysql/en/Windows_installation.html)
- **Linux:** Installing MySQL on Linux (http://dev.mysql.com/doc/mysql/en/Linux-RPM.html)
- **Solaris:** Installing MySQL on Other Unix-Like Systems (http://dev.mysql.com/doc/mysql/en/Installing_binary.html)

After you have completed the MySQL installation, continue to Section C.3, "Preparing the MySQL Database," on page 212.

C.3 Preparing the MySQL Database

To prepare the MySQL database, you must create a tablespace and auditusr account on the MySQL server. The tablespace and account enable the Secure Logging Server to log data to MySQL.

Although creating the tablespace is relatively straightforward, determining the hostname in the user account requires some planning. MySQL uses a username@hostname scheme for granting access to databases. If a user tries to log in with a valid user ID but the host name or IP address is incorrect, MySQL denies access to the user. The `%` wildcard grants access to a user logging in from anywhere except the localhost.

To determine the hostname for your user account, consider the following guidelines:

- If Secure Logging Server, iManager, and MySQL are on the same server and Nsure Audit Report is not used from a workstation, consider using `auditusr@'localhost'`.
- If Secure Logging Server and iManager are on the same server, MySQL is on a different server, and Nsure Audit Report is not used at all, consider using `auditusr@'serverIP' AND auditusr@'ServerDNSName'`.
- If Secure Logging Server, MySQL, and iManager are all on different servers, consider using `auditusr@'%'`.
- If you use Nsure Audit Report from a Windows workstation without a static IP address, consider using `auditusr@'%'`.
- In most cases, it is best to create both `auditusr@'%'` and `auditusr@'localhost'`. These username@hostname schemes simplify the authentication to MySQL.

To create the tablespace and user account:

1. On the server hosting the MySQL server software, type the following then specify the password when prompted:

   ```
   mysql -u root -p
   ```

   This loads MySQL Monitor and attempts to log in as user root. The -p switch instructs MySQL Monitor to request a password. When MySQL Monitor loads, a mysql prompt appears.

Depending on the server platform, you can also launch MySQL Monitor through the Mysql folder.

2 Create the Nsure Audit tablespace:

```
CREATE DATABASE naudit;
```

NOTE: You do not need to create the database table; the MySQL driver, `lgdmsql`, automatically creates this table when the logging server first loads the current Channel object configuration in memory. For more information on the table structure, see Section 7.9.1, "MySQL Channel Driver," on page 87.

3 Create the Nsure Audit (auditusr) user account:

```
GRANT all on naudit.* to auditusr@'%'
IDENTIFIED by 'password';
```

The user account has access from any IP address or host name. Be sure to replace `password` with the desired password for the user account.

4 Define which database to query:

```
\u mysql
```

5 Confirm the creation of the auditusr account:

```
select host,user from user;
```

In the User column, you should see `auditusr` next to the appropriate host.

C.4 Creating a User

Users in MySQL are created by granting rights. For example, if you grant a user rights to a database, that user is created.

1 To create a local user, in MySQL Monitor, run the following:

```
GRANT ALL PRIVILEGES ON database.table TO username IDENTIFIED BY
’some_pass’ WITH GRANT OPTION;
```

For example, `grant all privileges on naudit.* to audituser` grants the default audit user access to the default audit database. For additional information on privileges, see the MySQL documentation.

2 To create a remote user (a user who can log in from anywhere), run the following:

```
GRANT ALL PRIVILEGES ON database.table TO username@'%' IDENTIFIED
BY ’some_pass’ WITH GRANT OPTION;
```

For example, `grant all privileges on naudit.* to audituser@'%'` enables audituser to remotely access all tables in the naudit database. Optionally, the `%` wildcard could be replaced with a full or partial DNS name or IP address, for example, `audituser@'%.novell.com` allows access only from a DNS entry resolving to the .novell.com domain.

3 When finished, run `Flush Privileges;` to apply the changes.

C.5 Creating the MySQL Channel Object

1 In iManager, click *Auditing and Logging* > *Logging Server Options*.

2 In the *Secure Logging Server Name* field, browse to and select the Secure Logging Server; then click *OK*.

3 Click the *Channels* tab.

If you are using a Mozilla* browser, select *Channels* from the drop-down menu.

4 Check the *Channels* box, and then click *Channel Actions* > *New*.

5 In the *Channel Type* drop-down list, select *MySQL Channel* from the drop-down list, and specify a name in the *Channel Name* field. Click *OK*.

Do not use spaces, apostrophes, or any special characters in the channel name.

6 Complete the Configuration form.

For information on this form, see Section 7.9.2, “MySQL Channel Object,” on page 87.

7 Click *Apply*.

C.6 Configuring a JDBC Connection in iManager

JDBC database connections are used when performing queries in iManager. These steps assume that the required JDBC driver file (`.jar`) is in the iManager server’s defined classpath.

1 Download the JDBC driver for the MySQL database from the MySQL Development site (http://dev.mysql.com/downloads/).

The `.jar` file must be copied to the `tomcat/common/lib` folder of the server hosting the Nsure Audit iManager plug-in. The `.jar` file enables you to create JDBC connections to MySQL.

NOTE: If your browser renames the `.jar` file as a `.zip` file during the download, you must rename the file with the original `.jar` extension.

2 In iManager, click *Auditing and Logging* > *Query Options*.

3 In the Databases tab, click *New* to add a new database.

4 Follow the instructions in the online help to create the database entry.

C.7 Testing the MySQL Channel Configuration

You can test the MySQL channel configuration as follows:

1 In iManager, click *Auditing and Logging* > *Logging Server Options*.

2 In the *Secure Logging Server Name* field, browse to and select the Secure Logging Server; then click *OK*.

3 In the *General* tab, click *Configuration*.

If you are using a Mozilla browser, select *Configuration* from the drop-down menu.

4 In the *Log Channel* field, browse to and select the MySQL channel object that you just created in the *Log Channel* field. For example, mysql.Channels.Logging Services.

5 Click *Apply*.

6 (Conditional) On NetWare®, test the configuration:

6a At the server console, load `lengine -d .`

If the Nsure Audit console loads, then everything is configured correctly.

6b If it does not load, check the logger screen on the NetWare server for error messages. Use the error messages to troubleshoot the configuration.

7 (Conditional) On Linux or Solaris, test the configuration:

7a As user root from the Linux shell, type `/etc/init.d/novell-naudit restart`.

From the Solaris shell, type `/etc/init.d/naudit` .

7b Wait a few moments and type `ps -AH |grep lengine` .

If `lengine` is listed (which it should be a few times), then the Secure Logging Server is configured correctly. If a problem occurs during the configuration, error messages appear on the server console. Use the error messages to troubleshoot the configuration.

8 (Conditional) On Windows, test the configuration:

8a Click *Start* > *Control Panel* > *Administrative Tools* > *Services*.

8b Right-click the Nsure Audit Manager service, and click *Restart*.

8c Mouse over the Nsure Audit icon in the system tray; if the icon remains loaded and displayed, the Secure Logging Server is configured correctly.

8d If you suspect that the Secure Logging Server might not be configured correctly, open the debug display by pressing Shift and then clicking the *Nsure Audit* icon.

Use the messages to troubleshoot the configuration. When you close the debug display, the Audit Manager service is also shut down. You then need to restart the Audit Manager service to launch the Secure Logging Server.

C.8 Troubleshooting

The following Technical Information Documents contain MySQL troubleshooting information:

- How to troubleshoot the Nsure Audit MySQL channel—TID10088985 (http://support.novell.com/cgi-bin/search/searchtid.cgi?/10088985.htm)
- Is Nsure Audit actually capturing any data to the MySQL channel?—TID10092777 (http://support.novell.com/cgi-bin/search/searchtid.cgi?/10092777.htm)
- Nsure Audit Report Error 145 or 127 querying the MySQL database—TID10091360 (http://support.novell.com/cgi-bin/search/searchtid.cgi?/10091360.htm)
- Error: Could not find the Driver Class: com.mysql.jdbc.Driver—TID10091351 (http://support.novell.com/cgi-bin/search/searchtid.cgi?/10091351.htm)

C.9 SQL Expiration Command Variables

The following table lists the variables that can be used with expiration commands in the MySQL Channel object's SQL Expiration Commands property. For information on the SQL Expiration Commands property, see "MySQL Channel Object" on page 87.

IMPORTANT: When using SQL expiration command variables, remember the following:

- SQL expiration command variables are case sensitive.
- All variable strings must be preceded by a dollar sign ($).

Variable	Description
Nsure Audit Custom Variables	
T	Nsure Audit's default table schema.
l	The table name defined in the MySQL Channel object configuration.
e	The CREATE TABLE Options defined in the MySQL Channel object configuration.
Date Formatting	The date format can be tied together (for example, $Y$M$D).
Y	Year (four digit)
y	Year (two digit)
M	Month (two digit)
D	Day (two digit)
h	Hour (two digit)
m	Minute (two digit)
s	Second (two digit)
n	Now. This value displays as *yyyymmdd*.

C.9.1 Sample SQL Expiration Command Script

In the following example scripts, note that `l` is the letter L as in lion.

Creating and Renaming a Table

```
create table newtable ($T) $e;RENAME TABLE $l TO l$n, newtable TO $l
```

This script does the following:

1. Creates a new table using the CREATE TABLE Options.
2. Renames the current table so it includes the date and time in decimal.
3. Renames the new table with the default table name.

Deleting the Contents of a Table

```
truncate table $l;
```

This script deletes the contents of the table.

Using Oracle with Nsure Audit

D

This section contains the necessary tasks for setting up an Oracle database for use with Novell® Nsure™ Audit:

Refer to your Oracle manual for information on the following:

- How to install Oracle
- How to create users and grant access rights within Oracle
- How to optimize Oracle
- Query syntax
- Repairing an Oracle database

D.1 Channel Requirements

Before you set up an Oracle database for use with Nsure Audit, be aware of the following requirements:

- The Oracle channel driver is used only on platforms where Oracle can run natively, such as Windows, Linux, and Solaris. If you are running the Secure Logging Server on NetWare®, create a JDBC channel to connect to the Oracle server. For more information, see Section 7.7, "JDBC," on page 80.

  IMPORTANT: On Linux and Solaris systems, the 32-bit version of the Oracle client is required. The 64-bit version does not work because Nsure Audit is compiled as a 32-bit application.

- The database server must be using Oracle 8 or above.
- The Oracle 9 Client or greater or the Oracle Instant Client must be installed on the Secure Logging Server.
- The Oracle Transparent Network Substrate (TNS) architecture is used for connections to the Oracle Server.

D.2 Preparing the Oracle Database

To prepare the Oracle database, you must create a tablespace and user account on the Oracle server. The naudit tablespace and auditusr account enable the Secure Logging Server to log data to Oracle.

The auditusr account must have access to the naudit tablespace. The naudit tablespace must have an unlimited quota.This setup is the minimum that is required in order to have the Secure Logging Server connect to the Oracle database.

1 Log in to the Oracle server:

- **1a** On Windows, log in to the server as Administrator.
- **1b** On Linux or Solaris, log in to the system as the Oracle user. If you are logged into X-Windows, open a terminal window, such as konsole, xterm, or gnome-terminal.

2 (Conditional) On Linux or Solaris, execute SQLPlus by completing the following:

- **2a** Enter the following command:

  ```
  $ORACLE_HOME/bin/sqlplus system@servername
  ```

  The username is system, and the *servername* is the database SID.
- **2b** Specify the password when prompted.

3 (Conditional) On Windows, start SQLPlus by completing the following:

- **3a** Click *Start* > *Programs* > *Oracle-Orahome10* > *Application Development* > *SQL PLUS*.

  NOTE: The Oracle home name can vary depending on the Oracle configuration.

  If the Start menu option is not available, execute SQLPlus from the `Oracle_Home/bin` directory. For example, if the *Oracle_Home* directory is `drive:\oracle\ora10`, then type the following at the command prompt:

  ```
  drive:\oracle\ora10\bin\sqlplus system@servername
  ```
- **3b** Specify the username and password for the system account at the login dialog box.
- **3c** Specify the host string, which is usually the database SID or the global *dbname* (the name of the server).

4 At the SQLPlus prompt, create the naudit tablespace by typing the following lines.

```
CREATE TABLESPACE naudit
DATAFILE '/var/opt/oracle/SERVERNAME/naudit.dbf'
SIZE 10M
AUTOEXTEND ON NEXT 10M
MAXSIZE 7500M;
```

Press Enter at the end of each line. The semicolon at the end of the last line notifies SQLPlus that the command is finished.

Note the following about the lines:

- The DATAFILE path depends on the platform being used, the database location, and installation options. On Windows, this is typically `drive:\oracle\oradata\SERVERNAME`. On SuSE® Linux, the DATAPATH is typically `/var/opt/oracle/SERVERNAME`.
- *SERVERNAME* usually matches the database SID set up during installation.

- MAXSIZE is optional. If you omit MAXSIZE, put the semicolon at the end of the AUTOEXTEND line.

NOTE: You do not need to create the database table; the Oracle driver, `lgdora`, automatically creates this table when the logging server first loads the current Channel object configuration in memory. For more information on the table structure, see Section 7.10.1, "Oracle Channel Driver," on page 91.

5 At the SQLPlus prompt, create the auditusr account by typing the following lines:

```
CREATE USER AUDITUSR
IDENTIFIED BY passwd
DEFAULT TABLESPACE naudit
TEMPORARY TABLESPACE TEMP;
```

Press Enter at the end of each line. The semicolon at the end of the last line notifies SQLPlus that the command is finished. Replace *`passwd`* with the appropriate password.

6 At the SQLPlus prompt, use the following commands to grant the auditusr account rights to connect to the database:

Command	Description
`GRANT CREATE SESSION TO auditusr;`	CREATE SESSION allows auditusr to connect to the database. This is required.
`GRANT CREATE TABLE TO auditusr;`	CREATE TABLE is required only to auto-create the table as defined in the log channel. This command is not required if the database administrator wants to create the table in advance. If auditusr is not given the CREATE TABLE right, it is reduced to the role of simply adding and inserting data. If you do not grant the CREATE TABLE right to auditusr, you might need to grant the GRANT SELECT, INSERT ON right as follows so auditusr can add and insert data in the table: `GRANT SELECT, INSERT ON` *`database_name`* `to auditusr`
`ALTER USER auditusr QUOTA unlimited on naudit;`	QUOTA unlimited allows infinite transactions. This is required.

D.3 Setting Up a Remote Connection

This section contains brief instructions on creating an Oracle SID using the Oracle Net Configuration Assistant. See the documentation for your Oracle database or your database administrator for additional help.

1 Start the Oracle Net Configuration Assistant. This is installed as part of the Oracle client tools.

2 Select *Local Net Service Name Configuration*.

3 Select *Add*.

4 Select your Oracle version, then specify the name of the database service. This name should match the SID name displayed in the Oracle Enterprise Manager Console.

5 Select *TCP*.

6 Specify the host and port of your Oracle server, then verify your credentials.

The SID name you just created is provided as the Name parameter when creating an Oracle log channel object.

D.4 Installing and Configuring the Oracle Client

If the Oracle database is not hosted on the Nsure Audit server, you need to install and configure the Oracle client tools on the Nsure Audit server. You can use either the Oracle 9 Client or greater or the Instant Client.

This section includes information on installing and configuring the Instant Client:

- Section D.4.1, "Installing and Configuring the Instant Client for Linux or Solaris," on page 220
- Section D.4.2, "Installing and Configuring the Instant Client for Windows," on page 222

If you are using the Oracle client, use the Oracle documentation for installation information, and then follow the configuration tasks in this section.

D.4.1 Installing and Configuring the Instant Client for Linux or Solaris

1 Download the Instant Client for Linux or Solaris from the Oracle Instant Client web page (http://www.oracle.com/technology/tech/oci/instantclient/instantclient.html).

IMPORTANT: On Linux and Solaris systems, the 32-bit version of the Oracle client is required. The 64-bit version does not work because Nsure Audit is compiled as a 32-bit application.

2 (Conditional) On Linux, as user root install the Oracle Instant Client RPM on the Secure Logging Server:

```
rpm -Uvh oracle-instantclient-basic-10.1.0.3-1.i386.rpm
```

3 (Conditional) On Solaris, as user root extract the Oracle Instant Client zip to the Secure Logging Server in a directory such as `/usr/lib/oracle/10.1.0.3/client/lib`.

4 Change to the Oracle client directory:

```
cd /usr/lib/oracle/10.1.0.3/client/lib
```

5 Create a symbolic link so that Nsure Audit can find its linked library:

```
ln -s libclntsh.so.10.1 libclntsh.so.9.0
```

6 Set up TNS for the database by creating a `tnsnames.ora` file in the `/opt/novell/naudit` directory.

You can also copy the `tnsnames.ora` file from the `$ORACLE_HOME/admin/network` folder on the Oracle server.

7 Add the following lines to the `tnsnames.ora` file:

```
(DESCRIPTION
(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP)(HOST = SERVER_DNS_NAME)(PORT = 1521))
)
(CONNECT_DATA =
(SERVICE_NAME = SERVER_DNS_NAME)
(SID=ORACLE_SID)
```

[Optional. If you use the SID, do not use SERVICE_NAME.]

```
)
)
```

Note the following about the lines:

- For *SERVERNAME*, provide the name of the server or Oracle SID (usually this is in CAPS).
- For *SERVER_DNS_NAME* on the `ADDRESS=` line, provide the actual DNS name or IP address of the server.
- For *SERVER_DNS_NAME* on the `SERVICE_NAME=` line, provide the DNS name of the server or the global database name. The only time this should have to be different from the DNS name is when the DNS name contains dashes. Dashes are not allowed in global database names. If a different global database name is used, provide it here.

  If you are unsure of what to put in the `SERVICE_NAME` line, run `lsnrctl status` from the `Oracle_Home/bin` directory. Find the line that says `Service` followed by a name in quotes. The name in quotes is the name to provide in *SERVER_DNS_NAME*.

  You can use the Oracle SID of the server instead of *SERVICE_NAME*. The SID is usually the server name.

8 Add the LD_LIBRARY_PATH and TNS_ADMIN paths to `/etc/init.d/novell-naudit`.

9 Open `/etc/init.d/novell-naudit` in a text editor, such as vi.

10 In the `novell-naudit init` file, complete the following:

10a Change the LD_LIBRARY_PATH= to the following:

```
export LD_LIBRARY_PATH=/usr/lib:/opt/novell/naudit:/usr/lib/
oracle/10.1.0.3/client/lib:$LD_LIBRARY_PATH
```

10b Add the following below the LD_LIBRARY_PATH line:

```
export TNS_ADMIN=/opt/novell/naudit
```

10c Change the export LD_LIBRARY_PATH LC_ALL line to the following:

```
export LC_ALL
```

11 Save the file and exit the text editor.

Optionally, you can add the *LD_LIBRARY_PATH* and *TNS_ADMIN* paths to the profile `script /etc/profile`.

D.4.2 Installing and Configuring the Instant Client for Windows

1 Download the Instant Client for Windows from the Oracle Instant Client web page (http://www.oracle.com/technology/software/tech/oci/instantclient/index.html). Download both the Basic and ODBC packages.

2 Create a *drive*:\oracle folder; unzip the instantclient-odbc-win32-10.1.0.3-20050113.zip to this folder.

3 Unzip the contents of instantclient-basic-win32-10.1.0.3-20050113.zip to *drive*:\oracle.

Be sure to put oci.dll in the same folder as odbc_install.bat.

4 To install the ODBC driver, double-click odbc_install.bat.

The file might also appear as odbc_install if file extensions are hidden.

5 Add certain environment variables to the Windows system variables:

5a Right-click *My Computer*.

5b Click *Properties* > *Advanced* > *Environment Variables*.

5c In *System Variables*, click *New*.

5d To add a TNS_ADMIN environment variable, enter TNS_ADMIN in the *Variable Name* field and *drive*:\oracle in the Variable Value field. Click *OK*.

5e In *System Variables*, click *New*.

5f To add an ORACLE_HOME environment variable, enter ORACLE_HOME in the *Variable Name* field and *drive*:\oracle in the *Variable Value* field.

6 Create a tnsnames.ora file in the *drive*:\oracle folder.

Verify that Notepad doesn't append a .txt extension to the file.

7 Add the following lines to the tnsnames.ora file:

```
SERVERNAME =
(DESCRIPTION =
(ADDRESS_LIST =
 (ADDRESS = (PROTOCOL = TCP)(HOST = SERVER_DNS_NAME)(PORT = 1521))
)
(CONNECT_DATA =
(SERVICE_NAME = SERVER_DNS_NAME)
)
)
```

Note the following about the lines:

- For *SERVERNAME*, provide the name of the server or Oracle SID (usually this is in CAPS).
- For *SERVER_DNS_NAME* on the ADDRESS= line, provide the actual DNS name or IP address of the server.
- For *SERVER_DNS_NAME* on the SERVICE_NAME= line, provide the DNS name of the server or the global database name. The only time this should have to be different from the DNS name is when the DNS name contains dashes. Dashes are not allowed in global database names. If a different global database name is used, provide it here.

If you are unsure of what to put in the SERVICE_NAME line, run `lsnrctl status` from the `Oracle_Home/bin` directory. Find the line that says `Service` followed by a name in quotes. The name in quotes is the name to provide in *SERVER_DNS_NAME*.

8 Save the file and close Notepad.

9 Reboot the server in order to get the ORACLE_HOME variable working.

D.5 Creating the Oracle Channel Object

1 In iManager, click *Auditing and Logging* > *Logging Server Options*.

2 In the Nsure Audit Secure Logging Server name field, browse to and select the Secure Logging Server; click *OK*.

3 Click the *Channels* tab.

If you are using a Mozilla browser, select *Channels* from the drop-down menu.

4 Select the *Channels* box, then click *Channel Actions* > *New*.

5 In the Channel Type drop-down list, select *Oracle Channel* from the drop-down list, specify a name in the Channel Name field. Click *OK*.

Do not use spaces, apostrophes, or any special characters in the channel name.

6 Provide the appropriate information in the Database page.

For information on this page, see Section 7.10.2, "Oracle Channel Object," on page 91.

7 Click *Apply*.

D.6 Configuring a JDBC Connection in iManager

JDBC database connections are used when performing queries in iManager. These steps assume that the required JDBC driver file (`.jar`) is in the iManager server's defined classpath.

1 Download the JDBC driver for the Oracle database from the Oracle Web site (http://www.oracle.com/technology/tech/oci/instantclient/instantclient.html).

The `.jar` file must be copied to the `tomcat/common/lib` folder of the server hosting the Nsure Audit iManager plug-in. The `.jar` file enables you to create JDBC connections to Oracle.

NOTE: If your browser renames the `.jar` file as a `.zip` file during the download, you must rename the file with the original `.jar` extension.

2 In iManager, click *Auditing and Logging* > *Query Options*.

3 In the Databases tab, click *New* to add a new database.

4 Follow the instructions in the online help to create the database entry.

D.7 Testing the Oracle Channel Configuration

You can test the Oracle channel configuration as follows:

1 In iManager, click *Auditing and Logging* > *Logging Server Options*.

2 In the Nsure Audit Secure Logging Server name field, browse to and select the Secure Logging Server; then click *OK*.

3 In the *General* tab, click *Configuration*.

If you are using a Mozilla browser, select *Configuration* from the drop-down menu.

4 In the *Log Channel* field, browse to and select the Oracle channel object that you just created in the *Log Channel* field. For example, oracle.Channels.Logging Services.

5 Click *Apply*.

6 (Conditional) On NetWare, test the configuration:

6a At the server console, load `lengine -d`.

If the Nsure Audit 2.0 console loads, then everything is configured correctly.

6b If it does not load, check the logger screen on the NetWare server for error messages. Use the error messages to troubleshoot the configuration.

7 (Conditional) On Linux or Solaris, test the configuration:

7a As user root from the Linux or Solaris shell, type `/etc/init.d/novell-naudit restart`.

7b Wait a few moments, then type `ps -AH |grep lengine`.

If `lengine` is listed (which it should be a few times), then the Secure Logging Server is configured correctly. If a problem occurs during the configuration, error messages appear on the server console. Use the error messages to troubleshoot the configuration.

8 (Conditional) On Windows, test the configuration:

8a Click *Start* > *Control Panel* > *Administrative Tools* > *Services*.

8b Right-click the Nsure Audit Manager service, and click *Restart*.

8c Mouse over the *Nsure Audit* icon in the system tray; if the icon remains loaded and displayed, the Secure Logging Server is configured correctly.

8d If you suspect that the Secure Logging Server might not be configured correctly, open the debug display by pressing Shift and then clicking the *Nsure Audit* icon.

Use the messages to troubleshoot the configuration. When you close the debug display, the Audit Manager service is also shut down. You then need to restart the Audit Manager service to launch the Secure Logging Server.

D.8 Establishing an ODBC Connection to Oracle

An ODBC connection to Oracle is required to use Nsure Audit Report (`lreport.exe`), and other windows-based querying tools. See the Oracle ODBC Drivers Web site (http://otn.oracle.com/tech/windows/odbc/index.html) for details on obtaining, installing, and using the Oracle ODBC drivers.

D.9 Creating a View in Oracle

To use reports in the Nsure Audit Reporting Application (`lreport.exe`), you must create a view in Oracle as follows:

1 Log into SQLPLUS as a user with rights to create a view in the Nsure Audit database.

2 Execute the following command:

```
create view NAUDITLOG as * from table_name
```

The default table name defined by the Oracle channel is NAUDITLOG.

3 In the Nsure Audit Reporting Application, define the default table as NAUDITLOG.

For more information on this procedure, see “Defining Your Default Database and Table” on page 136.

Using Microsoft SQL Server with Nsure Audit

E

This section contains basic information on setting up a Microsoft SQL Server database for use with Novell® Nsure™ Audit.

Refer to your SQL Server manual for information on the following:

- How to install SQL Server
- How to create users and grant access rights within SQL Server
- How to optimize SQL Server
- Query syntax
- Repairing an SQL Server database

E.1 Channel Requirements

Before you set up a Microsoft SQL Server database for use with Nsure Audit, be aware of the following requirements:

- Microsoft Windows 2000 or 2003

  The Microsoft SQL Server Channel cannot be used on NetWare®, Linux, or Solaris. Create a JDBC channel to connect to the Microsoft SQL Server. For more information, see Section 7.7, “JDBC,” on page 80.
- Microsoft SQL Server 2000 SP1 or above
- If the SQL Server database is not hosted on the Nsure Audit server, install the SQL Server client tools on the Nsure Audit server. The client tools are included with the Microsoft SQL Server 2000 database installation.

E.2 Preparing the Microsoft SQL Server Database

To prepare the Microsoft SQL Server database, you must create an audit database and user account on the Microsoft SQL Server. The database and account enable the Secure Logging Server to log data to Microsoft SQL.

- Section E.2.1, “Creating an Audit Database,” on page 228
- Section E.2.2, “Creating an Audit Database User Account,” on page 228

E.2.1 Creating an Audit Database

1. In SQL Server Enterprise Manager, click *Tools* > *Wizards*.
2. In the Select Wizard page, expand *Databases* and click *Create Database Wizard.*
3. Complete the Create Database Wizard.

   Provide a relevant database name, such as `auditdb`. Do not use spaces or special characters in the database name.

NOTE: You do not need to create the database table; the Microsoft SQL Server channel driver, `lgdmssql`, automatically creates this table when the logging server first loads the current Channel object configuration in memory. For more information on the table structure, see Section 7.8.1, “Microsoft SQL Server Channel Driver,” on page 84.

E.2.2 Creating an Audit Database User Account

1. In SQL Server Enterprise Manager, click *Tools* > *Wizards*.
2. On the Select Wizard page, expand *Databases* and click *Create Login Wizard*.
3. On the Authentication Mode page, select *SQL Server Authentication*.
4. Specifies a username, such as `auditusr`, and a password.
5. On the Security Roles page, click *Next*.

   You do not need to grant access to any of the roles for the audit user.
6. Select the audit database which the account must access. (This is the database that you just created.)
7. After the wizard completes, expand *Databases* for the server.
8. Select the audit database, then double-click *Users* in the right-hand pane.
9. Double-click the user account.
10. On the Permit in Database Role page, select *DB_Owner*, *DB_Writer,* and *DB_Reader*. Click *OK* when finished.

E.3 Creating the Microsoft SQL Server Channel Object

1. In iManager, click *Auditing and Logging* > *Logging Server Options*.
2. In the *Secure Logging Server Name* field, browse to and select the Secure Logging Server; then click *OK*.
3. Click the *Channels* tab.

   If you are using a Mozilla browser, select *Channels* from the drop-down menu.
4. Select the *Channels* box, then click *Channel Actions* > *New*.
5. In the Channel Type drop-down list, select *MSSQL Channel* from the drop-down list, then specify a name in the Channel Name field. Click *OK*.

   Do not use spaces, apostrophes, or any special characters in the channel name.
6. Provide the appropriate information ins the Configuration page.

   For information on this page, see Section 7.8.2, "Microsoft SQL Server Channel Object," on page 85.
7. Click *Apply*.
8. Restart the Nsure Audit logging server.

E.4 Configuring a JDBC Connection in iManager

JDBC database connections are used when performing queries in iManager. These steps assume the required JDBC `.jar` file is in the iManager server's defined classpath.

1. Download the JDBC driver for your SQL Server database from the Microsoft SQL Server Downloads web page (http://www.microsoft.com/sql/downloads). After you complete the installation, copy the `mssqlserver.jar` file to the `tomcat/common/lib` folder of the server hosting the Nsure Audit iManager plug-in.

   This `.jar` file enables you to create JDBC connections to the SQL Server.
2. In iManager, select *Auditing and Logging* > *Query Options*.
3. Click *New* to add a new database.
4. Follow the instructions in the online help to create the database entry.

E.5 Testing the Microsoft SQL Server Channel Configuration

You can test the Microsoft SQL Server channel configuration as follows:

1. In iManager, click *Auditing and Logging* > *Logging Server Options*.
2. In the *Secure Logging Server Name* field, browse to and select the Secure Logging Server; then click *OK*.
3. In the *General* tab, click *Configuration*.

   If you are using a Mozilla browser, select *Configuration* from the drop-down menu.
4. In the *Log Channel* field, browse to and select the MSSQL Channel object that you just created in the *Log Channel* field. For example, mssql.Channels.Logging Services.

5 Click *Apply*.

6 (Conditional) On NetWare®, test the configuration:

6a At the server console, load `lengine -d .`

If the Nsure Audit console loads, then everything is configured correctly.

6b If it does not load, check the logger screen on the NetWare server for error messages. Use the error messages to troubleshoot the configuration.

7 (Conditional) On Linux or Solaris, test the configuration:

7a As user root from the Linux shell, type `/etc/init.d/novell-naudit restart`.

From the Solaris shell, type `/etc/init.d/naudit .`

7b Wait a few moments and type `ps -AH |grep lengine .`

If `lengine` is listed (which it should be a few times), then the Secure Logging Server is configured correctly. If a problem occurs during the configuration, error messages appear on the server console. Use the error messages to troubleshoot the configuration.

8 (Conditional) On Windows, test the configuration:

8a Click *Start* > *Control Panel* > *Administrative Tools* > *Services*.

8b Right-click the Nsure Audit Manager service, and click *Restart*.

8c Mouse over the Nsure Audit icon in the system tray; if the icon remains loaded and displayed, the Secure Logging Server is configured correctly.

8d If you suspect that the Secure Logging Server might not be configured correctly, open the debug display by pressing Shift and then clicking the *Nsure Audit* icon.

Use the messages to troubleshoot the configuration. When you close the debug display, the Audit Manager service is also shut down. You then need to restart the Audit Manager service to launch the Secure Logging Server.

E.6 Establishing an ODBC Connection to Microsoft SQL Server

An ODBC connection to Microsoft SQL Server is required to use Nsure Audit Report (`lreport`), and other windows-based querying tools. See the Microsoft SQL Server Web site (http://www.microsoft.com/sql/default.asp) for details on obtaining, installing, and using the Microsoft SQL Server ODBC drivers.

Using JDBC Data Stores with Nsure Audit

F

Novell® Nsure™ Audit has the ability to send events to any JDBC-enabled data store. This section contains the necessary tasks for creating a connection to a JDBC-enabled database.

- Section F.1, "Configuration Requirements," on page 231
- Section F.2, "Creating the JDBC Channel Object," on page 232

F.1 Configuration Requirements

The configuration requirements to use the JDBC channel with a JDBC-enabled data store are as follows:

- For performance reasons, we recommend using only the channels discussed in "Data Store" on page 27 as the primary log channel, and use JDBC data stores for notifications.
- Install and configure any JDBC-enabled data store according to the instructions provided by the vendor.
- In the JDBC-enabled data store, create a Nsure Audit database and a database user.
- The server hosting your JDBC data store must have JVM* 1.4.1 or later.
- Obtain the JDBC drivers for your data store.

  The JDBC drivers are available at the following sites:

Table F-1 *JDBC Driver Sites*

Data Store	Driver	Site
MySQL	MySQL Connector/J	http://dev.mysql.com/downloads/
Oracle	Oracle Instant Client	http://www.oracle.com/technology/tech/oci/instantclient/instantclient.html
Microsoft SQL Server	Microsoft SQL Server Driver for JDBC	http://www.microsoft.com/downloads/
QL Server and Sybase JDBC driver	jTDS (S)	http://www.sourceforge.net/

- Copy the JDBC drivers for your data store to the Nsure Audit Java classpath or a subdirectory thereof.

  Nsure Audit defines its Java classpath as follows:

Table F-2 *Nsure Audit Java Classpath*

Platform	Java Classpath
Windows Linux Solaris	`Novell_Audit_install_directory\java\logdriver\`
NetWare	`Novell_Audit_install_directory\`

- If you are going to query a JDBC data store in iManager, copy all required JDBC drivers (`*.jar`) to the following iManager classpaths on your iManager server:
  - **NetWare:** `sys:\tomcat\4\common\lib`
  - **Linux and Solaris:** /var/opt/novell/tomcat4/common/lib
  - **Windows:** `\program files\novell\tomcat\common\lib`
- If you are using the JDBC Channel on a Windows machine, add the `jvm.dll` directory path to the Path system variable. For example, `c:\j2sdk1.4.2_09\jre\bin\server\`. You must reboot the machine for the changes to take effect.
- On Linux/Solaris, the LD_LIBRARY_PATH variable must point to the paths for `libverify.so`, and `libjvm.so`. You must reboot the machine for the changes to take effect.
- On Linux and Solaris platforms, export LD_LIBRARY_PATH to the path of the server JVM. To do this, create `/etc/profile.local` (if it does not exist), then add an export line similar to the following:

  ```
  export LD_LIBRARY_PATH=/usr/lib/java/jre/lib/i386/server:/usr/lib/java/jre/lib/i386/
  ```

  Replace `/usr/lib/java` with the full path to the Java runtime environment, for example, `/usr/lib/SunJava2-1.4.1`.
- When creating the JDBC channel object in iManager, Java classpath entries must be separated by a colon if your JDBC data store is hosted on Linux or Solaris. If your JDBC data store is hosted on NetWare or Windows, Java classpath entries must be separated by a semicolon.

F.2 Creating the JDBC Channel Object

1. If you are creating the JDBC Channel on a Windows machine, you must add the `jvm.dll` directory path to the Path system variable. For example, `c:\j2sdk1.4.2_09\jre\bin\server\`. On Linux/Solaris, the LD_LIBRARY_PATH variable needs to point to the paths for `libverify.so`, and `libjvm.so`. You must reboot the machine for the changes to take effect.
2. In iManager, click *Auditing and Logging* > *Logging Server Options*.
3. In the *Secure Logging Server Name* field, browse to and select the Secure Logging Server; then click *OK*.
4. Click the *Channels* tab.

   If you are using a Mozilla browser, select *Channels* from the drop-down menu.
5. Select the *Channels* box, then click *Channel Actions* > *New*.

6. In the Channel Type drop-down list, select *JDBC Channel* from the drop-down list, then specify a name in the Channel Name field. Click *OK*.

   Do not use spaces, apostrophes, or any special characters in the channel name.
7. Provide the appropriate information ins the Configuration page.

   For information on this page, see Section 7.7.2, "JDBC Channel Object," on page 82.
8. Click *Apply*.
9. Restart the Nsure Audit logging server.

Commands and Utilities

G

This section reviews the startup commands and utilities used with Novell® Nsure™ Audit.

G.1 Platform Agent Startup

The Platform Agent (logevent) is the client portion of the Nsure auditing system. It must be installed on every server or workstation running applications that log events to Novell Nsure Audit.

Logevent is a shared library that is automatically loaded by its local logging applications. Consequently, it does not need to be manually loaded or unloaded.

G.2 Logging Cache Module Startup

The Logging Cache Module (lcache) writes events to the Disconnected Mode Cache if the connection between the Platform Agent and the Secure Logging Server fails. It is installed with logevent on every server or workstation running applications that log events to Novell Nsure Audit.

On NetWare® and Windows, logevent automatically loads lcache. On Linux, the eDirectory instrumentation, nauditds, automatically loads lcache. In some circumstances, on Linux and Solaris systems, lcache must be manually loaded.

To load lcache on Linux systems, enter

```
/opt/novell/naudit/lcache
```

To load lcache on Solaris systems, enter

```
/opt/NOVLnaudit/lcache
```

IMPORTANT: Do not unload lcache. Even if the local logging applications are no longer running, lcache must stay loaded so it can upload cached data to the Secure Logging Server. To prevent the Logging Cache Module from being unloaded, you can set the LogCacheUnload setting to No in the logevent file. For more information, see "Logevent" on page 42.

G.3 Secure Logging Server Startup Commands

The Secure Logging Server (lengine) is the server component in the Nsure auditing system. It is installed on the server you want to manage the flow of information to and from the auditing system.

Lengine automatically loads MDB, the Directory interface. Before starting the logging server, MDB verifies if Novell eDirectory™ is ready. If eDirectory is not ready, the logging server does not load.

NOTE: On Windows systems, the logging server loads, but it automatically falls back to Windows registry configuration.

The startup commands for NetWare, Windows, Linux, and Solaris systems are reviewed in the following sections.

- "Starting and Stopping the Secure Logging Server on NetWare" on page 236
- "Starting and Stopping the Secure Logging Server on Windows" on page 237
- "Starting and Stopping the Secure Logging Server on Linux" on page 237
- "Starting and Stopping the Secure Logging Server on Solaris" on page 237

G.3.1 Starting and Stopping the Secure Logging Server on NetWare

On NetWare, the startup script for the Secure Logging Server is included in the auditsvr.ncf file. Auditsvr.ncf is added to the server's autoexec.ncf file during installation so lengine.nlm loads each time the server restarts.

To manually load the Secure Logging Server on NetWare, enter

```
load lengine
```

or

```
load auditsvr.ncf
```

If you want to prevent the Secure Logging Server from being unloaded by users with access to the server console, you can append the -n switch to the server startup script. (For example, load lengine -n .)

To manually unload the Secure Logging Server on NetWare, enter

```
unload lengine
```

NOTE: Lengine.nlm and auditsvr.ncf are located in the sys:\system directory.

You must individually start or stop each logging server in the tree.

G.3.2 Starting and Stopping the Secure Logging Server on Windows

On Windows, the startup script for the Secure Logging Server is included in the naudit.exe file. Naudit.exe has an Automatic startup type so lengine.exe loads each time the server restarts.

To manually load or unload the Secure Logging Server on Windows, you must start or stop the Novell Nsure Audit Manager service:

1. Click Start > Settings > Control Panel.
2. Open the Services window.
   - On Window NT, select Services.
   - On Windows 2000 and XP, select Administrative Tools > Services.
3. In the list of installed services, right-click Novell Nsure Audit Manager, then select Start or Stop.

You must individually start or stop each logging server in the tree.

G.3.3 Starting and Stopping the Secure Logging Server on Linux

On Linux, the startup script for the Secure Logging Server is /etc/init.d/novell-naudit . This startup script loads lengine each time the server restarts.

To manually start the Secure Logging Server on Linux, enter

```
/etc/init.d/novell-naudit start
```

To stop the Secure Logging Server on Linux, enter

```
/etc/init.d/novell-naudit stop
```

You must individually start or stop each logging server in the tree.

G.3.4 Starting and Stopping the Secure Logging Server on Solaris

On Linux, the startup script for the Secure Logging Server is /etc/init.d/naudit . This startup script loads lengine each time the server restarts.

To manually start the Secure Logging Server on Solaris, enter

```
/etc/init.d/naudit start
```

To stop the Secure Logging Server on Solaris, enter

```
/etc/init.d/naudit stop
```

You must individually start or stop each logging server in the tree.

G.4 NetWare and eDirectory Instrumentation Startup Commands

The NetWare and eDirectory Instrumentations for Novell Nsure Audit (auditNW and nauditDS, respectively) allow Nsure Audit to log NetWare, eDirectory, and file system events.

For information on selecting which events you want Novell Nsure Audit to log, see Chapter 5, "Logging eDirectory, NetWare, and File System Events," on page 61.

To enable NetWare and file system logging, auditNW must be loaded on every server on which you want to log NetWare and file system events. To avoid receiving duplicate entries for eDirectory events, enable the do not sent replicated events option. To enable this, open the Nsure Audit tab of your NCP Server object and check the "Do not send replicated events" checkbox. To log non-replicated events (such as logins), it must be installed on each individual server for which you want to log non-replicated events.

Additionally, the Platform Agent must be installed on every server on which you want to log NetWare, file system, and eDirectory events. AuditNW and nauditDS automatically load the Platform Agent (logevent) to send events to the Secure Logging Server.

Typically, auditNW and nauditDS should be automatically loaded each time the server or workstation restarts. However, you can also manually load or unload the instrumentation files. The following sections review the instrumentation startup commands for NetWare, Windows, Linux, and Solaris systems.

- "Starting and Stopping the NetWare and eDirectory Instrumentations on NetWare" on page 238
- "Starting and Stopping the eDirectory Instrumentation on Windows" on page 239
- "Starting and Stopping the eDirectory Instrumentation on Linux and Solaris" on page 239

G.4.1 Starting and Stopping the NetWare and eDirectory Instrumentations on NetWare

NOTE: At server startup, the NetWare and eDirectory instrumentations should be loaded as soon as possible, but they must be loaded after TCP/IP.

On NetWare, the startup scripts for auditNW and nauditDS are included in the auditagt.ncf file. Auditagt.ncf is added to the server's autoexec.ncf file during installation. Therefore, the NetWare and eDirectory Instrumentations automatically load each time the server restarts.

If you want to prevent auditNW or nauditDS from being unloaded by users with access to the server console, you can append the -n switch to the agent startup scripts. (For example, load auditnw -n .)

To manually start the NetWare or eDirectory Instrumentation on NetWare, enter

```
load auditnw
```

or

```
load nauditds
```

To load both the NetWare and eDirectory Instrumentations, enter

```
load auditagt.ncf
```

To stop the NetWare and eDirectory Instrumentations on NetWare, enter

```
unload auditnw
```

```
unload nauditds
```

NOTE: auditnw.nlm, audit.ds, and auditagt.ncf are located in the sys:\system directory.

You must individually start or stop the instrumentations on each server in the tree.

G.4.2 Starting and Stopping the eDirectory Instrumentation on Windows

On Windows, the eDirectory Instrumentation is managed through the Novell eDirectory Services utility. By default, the eDirectory Instrumentation must be manually loaded on one server per DS Replica.

To manually load or unload the eDirectory Instrumentation on Windows:

1. Load ndscons.exe.

   ndscons.exe is usually in the \novell\nds\ directory.
2. In the list of installed services, select the Novell Nsure Audit Component.
3. Click Start or Stop.

To configure nauditDS.dlm to load each time the server restarts:

1. Load ndscons.exe.

   ndscons.exe is usually in the \novell\nds\ directory.
2. In the list of installed services, select the Novell Nsure Audit Component.
3. Click Startup.
4. Mark the Automatic startup type, then click OK.

G.4.3 Starting and Stopping the eDirectory Instrumentation on Linux and Solaris

On Linux and Solaris systems, the eDirectory Instrumentation must be manually loaded on one server per DS Replica.

To manually start the eDirectory Instrumentation on Linux or Solaris, enter

```
ndstrace -c “load nauditds”
```

To manually stop the eDirectory Instrumentation on Linux or Solaris, enter

```
ndstrace -c “unload nauditds”
```

To automatically load the eDirectory Instrumentation each time the server restarts, add

```
nauditds auto #Nsure Audit Platform Agent
```

to /usr/lib/nds-modules/ndsmodules.conf.

NOTE: On Linux systems, the startup script is /etc/init.d/novell-naudit . On Solaris systems, the startup script is /etc/init.d/naudit .

G.5 AuditExt

The AuditExt utility adds Novell Nsure Audit objects (Logging Services and its associated containers, the Logging Server object, Channel objects, Notification objects, and Application objects) to the eDirectory schema.

Logging applications use AuditExt to create their associated Application objects and to populate the Application objects' log schema attribute.

Novell Nsure Audit stores LSC files as attributes in their respective Application object. English LSC files are stored under the NAuditAppSchemaEn attribute, French LSC files are stored under the NAuditAppSchemaFr attribute, and so forth.

AuditExt is also required to uninstall Novell Nsure Audit. For information on this procedure, see Section 11.4, “Uninstalling Novell Nsure Audit,” on page 173.

G.5.1 Using AuditExt to Extend the Schema

The installation program uses AuditExt to extend the eDirectory schema during the initial installation. Under normal circumstances, the schema should only be extended one time. This is automatically done during the Novell Nsure Audit installation on the first server in the tree.

If, for some reason, the initial schema extension fails, you can run AuditExt to extend the schema again. However, you should not try to extend the schema again until the first schema extension is fully replicated.

NOTE: A common indicator that the Novell Nsure Audit schema extension has failed is when you create Nsure Audit objects, but the objects don't get added to the tree. The tree doesn't recognize the attribute even though you are able to create the objects in iManager.

Another instance when you might need to run the AuditExt utility is to re-create the Logging Services container. If Logging Services is deleted from the tree, it can only be re-created by running AuditExt.

To use AuditExt to extend the eDirectory schema or re-create the Logging Services container:

1. Launch Auditext at the server console.
   - On NetWare, enter `sys:\system\auditext.nlm`.
   - On Windows, enter `\program files\novell\nsure audit\auditext.exe`.
   - On Linux, enter `/opt/novell/naudit/auditext`.
   - On Solaris, enter `/opt/NOVLnaudit/auditext`.
2. Specify your admin username and password.
3. Select Add Schema Extensions, then press Enter.

   AuditExt adds the Nsure Audit objects to the eDirectory schema.

G.5.2 Using AuditExt to Add LSC Files to Application Objects

During their installations, logging applications use the AuditExt utility to automatically create their associated Application objects and to populate the Application objects' log schema attribute. However, if you modify or localize a Log Schema (LSC) file, you can manually add it to the Application object by running the AuditExt utility at the server console.

To add a log schema to an Application object at the server console, enter the following command:

```
auditext -lsc -u:username -p:password "-a:Application_object" -
f:LSC_file -l:language
```

NOTE: If the path to the LSC file contains spaces, enclose the path and the -f flag in quotation marks. For example, "-f:c:/my files/myapp.lsc".

The following is a sample command that adds the English edir.lsc file to the eDirectory Instrumentation Application object:

```
auditext -lsc -u:admin -p:argl "-a:eDirectory Instrumentation" -
f:\temp\edir.lsc -l:en
```

AuditExt requires that the first line of all LSC files is formatted as follows:

```
#^object_name^Application_ID^Application_Identifier^language_identifie
r
```

Each parameter is explained below.

Parameter	Description
object_name	The string that is used as the name of the Application object.
Application_ID	The four-digit hex value assigned to the current application. All Application IDs are assigned through Novell Developer Support and are maintained in the Novell Nsure Audit central registry.
Application_Identifier	The name the logging application uses to identify itself to the logging server. The Application Identifier is stored in the application's certificate.
language_identifier	A two-character code for the current LSC file's language. • EN = English • ES = Spanish • FR = French • DE = German • IT = Italian • PT = Portuguese • RU = Russian

If no path is given, AuditExt looks for the log schema files in the working directory of AuditExt. By default, schema log files are contained following directories:

Operating System	Directory
NetWare	sys:\system\naudit\*.lsc
Windows	\program files\novell\nsure audit\logschema\*.lsc
Linux	/opt/novell/naudit/logschema/*.lsc
Solaris	/opt/NOVLnaudit/logschema/*.lsc

G.6 AudCGen

AudCGen is a command line utility that generates custom x.509 certificates for the Secure Logging Server and logging applications. Novell Nsure Audit uses certificates to authenticate logging applications and sign events. For more information on generating certificates, see Chapter 10, "Security and Non-Repudiation," on page 159.

The AudCGen syntax is as follows:

```
audcgen -cert:filename -pkey:filename [-f] [base:directory] [-
bits:number] [-serial:number]
[-valid:days] {-ss | -appcert:filename -apppkey:filename -
app:Application_Identifier | -v
-out:target_certificate}
```

The following table reviews each of the command parameters.

Parameter	Description
`-cert:filename`	The path and filename for the Secure Logging Certificate. The default path and filename is *base*/cacert.pem .
`-pkey:filename`	The path and filename for the Secure Logging Certificate's private key. The default path and filename is base/capkey.pem .
[-f]	Force overwrite. AudCGen overwrites any existing certificates or private keys of the same name (for example, cacert.pem and capkey.pem or appcert.pem and apppkey.pem) in the output directory. This parameter is optional. If you do not use the -f parameter and there is an existing certificate of the same name, AudCGen aborts.
`[base:directory]`	The default directory for the certificate and private key files. By default, *base* is the root directory. This parameter is optional. If you do not designate a base directory, you can include the directory in the certificate and private key strings.

Parameter	Description
`[-bits:`*`number`*`]`	The number of bits for the certificate.
	The default is 512; however, Novell Nsure Audit can handle certificates up to 1472 bits.
`[-serial:`*`number`*`]`	This parameter assigns a serial number to the current certificate. You can use this option to keep track of your system's certificates.
	This parameter is optional.
`[-valid:`*`days`*`]`	The certificate's expiration in days.
	The current iteration of Novell Nsure Audit does not verify if a certificate is valid.
-ss	Self-sign.
	AudCGen generates a self-signed CA certificate and key.
`-appcert:`*`filename`*	The output path and filename for the Logging Application Certificate.
	The default path and filename is *base*\appcert.pem.
`-apppkey:`*`filename`*	The output path and filename for the Logging Application Certificate.
	The default path and filename is *base*\apppkey.pem.
`-app:`*`Application_Identifier`*	The logging application's Application Identifier.
	This value must match the Application Identifier stored the logging application's Application object.
`-v`	Validate.
	AudCGen validates the certificate designated in the -out parameter.
`-out:`*`target_certificate`*	The path and filename for the certificate you want to validate.
	The default path and filename is *base**.pem

G.7 LETrans

LETrans is a command-line utility that translates raw text log files into human-readable form. It also includes the ability to query an ODBC datasource on your Windows machine, then translate and format the output.

The LEtrans utility takes no parameters; it is configured using the letrans.cfg file. The letrans.cfg file contains a description of each LETrans configuration option.

To Launch LETrans:

1 Open letrans.cfg in a text editor. LETrans and letrans.cfg are located in the following directories:

Operating System	Directory
NetWare	`sys:\system\naudit`
Windows	`\program files\novell\nsure audit`
Linux	`/opt/novell/naudit`
Solaris	`/opt/NOVLnaudit`

2. List the path and name of each untranslated log file in the source files section.
3. Add the path to the log schema file (.lsc) for any additional instrumentations you are using in the schema section.
4. Save letrans.cfg, then execute LETrans from the server.

G.8 NauditPAConfig

The Platform Agent Configuration Tool provides a graphical interface to manage Nsure Audit Platform Agents. This tool operates by making changes to the `logevent.cfg` file, which contains configuration settings for the Platform Agent.

To make configuration changes, you can either open and edit an existing `logevent.cfg` configuration file, or create a new `logevent.cfg` file. When your changes are complete, the updated file must be saved in the correct location for your changes to be applied.

The Platform Agent Configuration tool Java Archive file is located in the following directories:

Table G-1 *NauditPAConfig Directories*

Platform	Directory
NetWare	`sys:\system\naudit\nauditpaconfig.jar`
Windows	`\program files\novell\nsure audit\nauditpaconfig.jar`
Linux	`/opt/novell/naudit/java/nauditpaconfig.jar`
Solaris	`/opt/NOVLnaudit/java/nauditpaconfig.jar`

For a description of each available parameter, see "Logevent" on page 42.

G.8.1 Running the Platform Agent Configuration Tool

To use the Platform Agent Configuration Tool, you must have a Java Runtime Environment (JRE) installed on your workstation. It is also helpful to add the location of your JRE to the path environment variable, though this isn't required if you specify the full path to the Java executable.

To launch the Platform Agent Configuration tool, execute the following command at a console, from the folder where the `nauditpaconfig.jar` file is located:

```
java -jar nauditpaconfig.jar
```

File Descriptions and Locations

H

This section provides a listing of the Novell® Nsure™ Audit program files and their locations on each operating system.

- Section H.1, "Program Files and Directories," on page 245
- Section H.2, "Program Directories," on page 249

H.1 Program Files and Directories

The following table lists the program files and directories, their associated components, and their functions.

See Section H.2, "Program Directories," on page 249 for a breakout of program filenames by operating system.

File	Program Component	Function
`audcgen`		AudCGen is a command line utility that generates SSL certificates for the Secure Logging Server and logging applications. Nsure Audit uses certificates to authenticate logging applications and sign events. For more information, see Section G.6, "AudCGen," on page 242.
`auditagt.ncf`		Auditagt.ncf contains the startup script for the Novell eDirectory™ and NetWare® Instrumentations. It is added to the logging server's autoexec.ncf file during installation.
`auditDS`	eDirectory Instrumentation	The auditDS module hooks into the eDirectory event system. It logs all events configured under *Nsure Audit* > *eDirectory* in the NCP Server object. For more information, see Section B.1, "eDirectory Events," on page 191. **IMPORTANT:** On Windows, `auditDS` must be located in the same directory as the directory database set (DIB). By default, the DIB is located in `\novell\nds\` . On UNIX, `auditDS` must be in `/usr/lib/nds-modules` .
`auditext`		A utility used to add or remove the Nsure Audit objects and attributes from the eDirectory schema. AuditExt is used by the installation program to extend the eDirectory schema during installation. For more information, see Section G.5, "AuditExt," on page 240.

File	Program Component	Function
auditNW	NetWare Instrumentation (for NetWare only)	The auditNW module hooks into the NSS, Traditional File System, and NetWare event systems. It logs all events configured under *Nsure Audit* > *Filesystem* or *Nsure Audit* > *NetWare* in the NCP Server object. For more information, see Section B.3, "NetWare Events," on page 207 and Section B.2, "File System Events," on page 206.
auditsvr.ncf		Auditsvr.ncf contains the startup script for the Secure Logging Server. It is added to the logging server's autoexec.ncf file during installation.
Disconnected Mode Cache Directory	Logging Cache Module for the Platform Agent	The directory where the Logging Cache Module temporarily logs incoming information if the connection to the Secure Logging Server is interrupted.
Java Classpath	Java channel driver JDBC channel driver	The location of the java source code, as well as any other classes or objects that are required by the Nsure Audit Java and JDBC channel drivers.
jdbclogdriver.jar	JDBC channel driver	jdbclogdriver.jar is a Java archive file that contains the JDBC channel driver. For more information, see Section 7.7, "JDBC," on page 80.
lcache	Logging Cache Module for the Platform Agent	The Logging Cache Module enables the Platform Agent to temporarily log incoming information to its Disconnected Mode Cache if the connection to the Secure Logging Server is interrupted. For more information on configuring the disconnected mode cache, see Section 4.1, "Configuring the Platform Agent," on page 41.
lengine	Secure Logging Server	The Secure Logging Server is the server component in the Nsure Auditing system. It manages the flow of information to and from the Nsure Auditing system. It receives incoming events and requests from the Platform Agents; logs information to the data store; monitors designated events; and provides filtering and notification services. For more information, see Section 4.2, "Configuring the Secure Logging Server," on page 44.
letrans	LETrans	A utility designed to convert raw data files logged by the file channel, into a readable, delimited format.

File	Program Component	Function
`lgdcvr`	Critical Value Reset (CVR) channel driver	The CVR channel allows you to flag an attribute in eDirectory with a reset policy. If the value of that specific attribute is changed, the CVR channel driver resets the value as per the policy defined in the CVR Channel object. For more information, see Section 7.4, “CVR,” on page 71.
`lgdfile`	File channel driver	The File channel driver writes events to a flat file in the file system. The driver can be configured to provide a single, comma-separated file (`*.csv`) or a human-readable log file. For more information, see Section 7.5, “File,” on page 74.
`lgdjava`	Java channel driver	The Java channel driver allows the logging server to output filtered events to a Java application. For more information, see Section 7.6, “Java,” on page 79.
`lgdmssql`	Microsoft SQL Server channel driver	The MSSQL channel driver writes events to a Microsoft SQL Server database. For more information, see Section 7.8, “Microsoft SQL Server,” on page 83.
`lgdmsql`	MySQL channel driver	The MySQL channel driver writes events to a MySQL database. For more information, see Section 7.9, “MySQL,” on page 86.
`lgdora`	Oracle channel driver	The Oracle channel enables the logging server to log events to an Oracle database. For more information, see Section 7.10, “Oracle,” on page 90.
`lgdsmtp`	SMTP channel driver	The SMTP channel driver e-mails events to a designated relay host. For more information, see Section 7.11, “SMTP,” on page 93.
`lgdsnmp`	SMNP channel driver	The SNMP channel driver sends events to an SNMP management system. For more information, see Section 7.12, “SNMP,” on page 95.

File	Program Component	Function
`lgdsyslg`	Syslog channel driver	The Syslog channel driver allows the logging server to log events to a specific syslog facility on any syslog host or to a remote syslog daemon. For more information, see Section 7.13, "Syslog," on page 98.
`log`	File channel log file	The filename for the File Channel data store.
`logevent`	Platform Agent	The Platform Agent is the client component in the Nsure auditing system. It receives logging information and system requests from individual applications and transmits the information to the Secure Logging Server. For more information, see Section 4.1, "Configuring the Platform Agent," on page 41.
`logevent.cfg` `logevent.conf`	Platform Agent configuration file	A text-based configuration file that stores the local Platform Agent's configuration settings. Every computer running the Platform Agent has its own `logevent.cfg` file. For more information on the configuration settings and file location, see Section 4.1, "Configuring the Platform Agent," on page 41.
`lreport`	Nsure Audit Report program file	Nsure Audit Report is a Windows-based, ODBC-compliant application that can query Oracle and MySQL data stores (or any other database that has ODBC driver support).
LSC files		Log Schema (LSC) files catalog the events that can be logged for a given application. They also provide event descriptions and field titles, although this is optional.
`mdb`	Interface to Novell eDirectory, the Windows registry, the file system, or other systems	MDB is a simple API that provides all the functionality required to access and manipulate a variety of hierarchical databases (for example, eDirectory, the Windows registry, the file system, or other systems). The MDB API allows Nsure Audit to become independent of the actual database. The underlying concepts of MDB are similar to ODBC, though the functionality differs.
`mdbds`		MDBDS is the MDB driver for eDirectory.
`mdbreg`		MDBREG is the MDB driver for the Windows Registry.
`naudit`		Naudit contains the startup script for the Secure Logging Server on Windows, Linux, and Solaris systems.

File	Program Component	Function
nauditpaconfig.jar		The Platform Agent Configuration Tool provides a graphical interface to manage Nsure Audit Platform Agents. This tool operates by making changes to the logevent.cfg file, which contains configuration settings for the Platform Agent. For more information, see Section G.8, "NauditPAConfig," on page 244.
nproduct.log		Nproduct.log contains any errors that have been logged during startup on Windows, Linux, and Solaris systems.

H.2 Program Directories

The following sections list the Nsure Audit program files and their locations by operating system:

- Section H.2.1, "NetWare Program Directories," on page 249
- Section H.2.2, "Windows Program Directories," on page 250
- Section H.2.3, "Linux Program Directories," on page 252
- Section H.2.4, "Solaris Program Directories," on page 253

H.2.1 NetWare Program Directories

Table H-1 *NetWare Program Directories*

File	Directory
audcgen	sys:\system\audcgen
auditagt.ncf	sys:\system\auditagt.ncf
auditDS	sys:\system\auditDS.nlm
auditext	sys:\system\auditext.nlm
auditNW	sys:\system\auditNW.nlm
auditsvr.ncf	sys:\system\auditsvr.ncf
Disconnected Mode Cache Directory	sys:\etc\logcache\
Java Classpath	sys:\system\java\logdriver\
jdbclogdriver.jar	sys:\system\java\logdriver\jdbclogdriver.jar
lcache	sys:\system\lcache.nlm
lengine	sys:\system\lengine.nlm
letrans	n/a
lgdcvr	sys:\system\lgdcvr.nlm

File	Directory
`lgdfile`	`sys:\system\lgdfile.nlm`
`lgdjava`	`sys:\system\lgdjava.nlm`
`lgdmssql`	n/a
`lgdmsql`	`sys:\system\lgdmsql.nlm`
`lgdora`	n/a
`lgdsmtp`	`sys:\system\lgdsmtp.nlm`
`lgdsnmp`	`sys:\system\lgdsnmp.nlm`
`lgdsyslg`	`sys:\system\lgdsyslg.nlm`
`log`	`sys:\etc\logdir\log`
`logevent`	`sys:\system\logevent.nlm`
`logevent.cfg` `logevent.conf`	`sys:\etc\logevent.cfg`
`lreport`	n/a
LSC files	`sys:\system\naudit\*.lsc`
`mdb`	`sys:\system\mdb.nlm`
`mdbds`	`sys:\system\mdbds.nlm`
`mdbreg`	n/a
`naudit`	n/a
`nauditpaconfig.jar`	`sys:\system\naudit\nauditpaconfig.jar`
`nproduct.log`	n/a

H.2.2 Windows Program Directories

***Table H-2** Windows Program Directories*

File	Directory
`audcgen`	`\program files\novell\nsure audit\audcgen.exe`
`auditagt.ncf`	n/a
`auditDS`	`auditDS.dlm` must be located in the same directory as the directory database set (DIB). By default, the DIB is located in novell\nds\ .
`auditext`	`\program files\novell\nsure audit\auditext.exe`
`auditNW`	n/a
`auditsvr.ncf`	n/a

File	Directory
Disconnected Mode Cache Directory	`\program files\novell\nsure audit\cache\`
Java Classpath	`\program files\novell\nsure audit\java\logdriver\`
`jdbclogdriver.jar`	`\program files\novell\nsure audit\java\logdriver\jdbclogdriver.jar`
`lcache`	`\program files\novell\nsure audit\lcache.exe`
`lengine`	`\program files\novell\nsure audit\lengine.exe`
`letrans`	`\program files\novell\nsure audit\letrans.exe`
`lgdcvr`	`\program files\novell\nsure audit\lgdcvr.dll`
`lgdfile`	`\program files\novell\nsure audit\lgdfile.dll`
`lgdjava`	`\program files\novell\nsure audit\lgdjava.dll`
`lgdmssql`	`\program files\novell\nsure audit\lgdmssql.dll`
`lgdmsql`	`\program files\novell\nsure audit\lgdmsql.dll`
`lgdora`	`\program files\novell\nsure audit\lgdora.dll`
`lgdsmtp`	`\program files\novell\nsure audit\lgdsmtp.dll`
`lgdsnmp`	`\program files\novell\nsure audit\lgdsnmp.dll`
`lgdsyslg`	`\program files\novell\nsure audit\lgdsyslg.dll`
`log`	`\program files\novell\nsure audit\logs\log`
`logevent`	`\program files\novell\nsure audit\logevent.dll`
`logevent.cfg` `logevent.conf`	*windows_directory*`\logevent.cfg`
`lreport`	`\program files\novell\nsure audit\lreport.exe`
LSC files	`\program files\novell\nsure audit\logschema\*.lsc`
`mdb`	`\program files\common files\novell shared\mdb\mdb.dll`
`mdbds`	`\program files\common files\novell shared\mdb\mdbds.dll`
`mdbreg`	`\program files\common files\novell shared\mdb\mdbreg.dll`
`naudit`	`\program files\novell\nsure audit\naudit.exe`
`nauditpaconfig.jar`	`\program files\novell\nsure audit\nauditpaconfig.jar`

File	Directory
nproduct.log	*root_directory*:\ You can override the default location of the nproduct.log file by creating an ERRORFILE environment variable that includes the full path and filename of the log file. For example, c:\documents and settings\all users\nproduct.log c:\logserverlog.txt

H.2.3 Linux Program Directories

Table H-3 *Linux Program Directories*

File	Directory
audcgen	/opt/novell/naudit/audcgen
auditagt.ncf	n/a
auditDS	/usr/lib/nds-modules/libauditDS.so
auditext	/opt/novell/naudit/auditext
auditNW	n/a
auditsvr.ncf	n/a
Disconnected Mode Cache Directory	/var/opt/novell/naudit/cache/
Java Classpath	/var/opt/novell/naudit/java/logdriver/
jdbclogdriver.jar	/opt/novell/naudit/java/logdriver/jdbclogdriver.jar
lcache	/opt/novell/naudit/lcache
lengine	/opt/novell/naudit/lengine
letrans	n/a
lgdcvr	/opt/novell/naudit/lgdcvr.so
lgdfile	/opt/novell/naudit/lgdfile.so
lgdjava	/opt/novell/naudit/lgdjava.so
lgdmssql	n/a
lgdmsql	/opt/novell/naudit/lgdmsql.so
lgdora	/opt/novell/naudit/lgdora.so
lgdsmtp	/opt/novell/naudit/lgdsmtp.so
lgdsnmp	/opt/novell/naudit/lgdsnmp.so
lgdsyslg	/opt/novell/naudit/lgdsyslg.so
log	/var/opt/novell/naudit/logs/log

File	Directory
`logevent`	`/usr/lib/liblogevent.so`
`logevent.cfg` `logevent.conf`	`/etc/logevent.conf`
`lreport`	n/a
LSC files	`/opt/novell/naudit//logschema/*.lsc`
`mdb`	`/usr/lib/libmdb.so`
`mdbds`	`/usr/lib/mdb/mdbds.so`
`mdbreg`	n/a
`naudit`	`/etc/init.d/novell-naudit`
`nauditpaconfig.jar`	`/opt/novell/naudit/java/nauditpaconfig.jar`
`nproduct.log`	`/var/opt/novell/naudit/nproduct.log` You can override the default location of the `nproduct.log` file by creating an errorfile environment variable that includes the full path and filename of the log file. If the default location and the location specified in the ERRORFILE environment variable cannot be used, then `nproduct.log` appears in the home directory of the user executing the file. If a home directory and all path attempts have failed, then the error is written to the console screen.

H.2.4 Solaris Program Directories

Table H-4 *Solaris Program Directories*

File	Directory
`audcgen`	`/opt/NOVLnaudit/audcgen`
`auditagt.ncf`	na/
`auditDS`	`/usr/lib/nds-modules/libauditDS.so`
`auditext`	`/opt/NOVLnaudit/auditext` `/opt/NOVLnaudit/auditext.sh`
`auditNW`	n/a
`auditsvr.ncf`	n/a
Disconnected Mode Cache Directory	`/opt/NOVLnaudit/cache/`
Java Classpath	`/opt/NOVLnaudit/java/logdriver/`
`jdbclogdriver.jar`	`/opt/NOVLnaudit/java/logdriver/jdbclogdriver.jar`
`lcache`	`/opt/NOVLnaudit/lcache`

File	Directory
lengine	/opt/NOVLnaudit/lengine
letrans	n/a
lgdcvr	/opt/NOVLnaudit/lgdcvr.so
lgdfile	/opt/NOVLnaudit/lgdfile.so
lgdjava	/opt/NOVLnaudit/lgdjava.so
lgdmssql	n/a
lgdmsql	/opt/NOVLnaudit/lgdmsql.so
lgdora	/opt/NOVLnaudit/lgdora.so
lgdsmtp	/opt/NOVLnaudit/lgdsmtp.so
lgdsnmp	/opt/NOVLnaudit/lgdsnmp.so
lgdsyslg	/opt/NOVLnaudit/lgdsyslg.so
log	/opt/NOVLnaudit/logs/log
logevent	/usr/lib/liblogevent.so
logevent.cfg logevent.conf	/etc/logevent.conf
lreport	n/a
LSC files	/opt/NOVLnaudit/logschema/*.lsc
mdb	/usr/lib/libmdb.so
mdbds	/usr/lib/mdb/mdbds.so
mdbreg	n/a
naudit	/etc/init.d/naudit
nauditpaconfig.jar	/opt/NOVLnaudit/java/nauditpaconfig.jar
nproduct.log	/opt/NOVLnaudit/nproduct.log You can override the default location of the nproduct.log file by creating an errorfile environment variable that includes the full path and filename of the log file. If the default location and the location specified in the ERRORFILE environment variable cannot be used, then nproduct.log appears in the home directory of the user executing the file. If a home directory and all path attempts have failed, then the error is written to the console screen.

Documentation Updates

I

This section contains information on documentation content changes that have been made in the *Administration Guide* for Novell® Nsure™ Audit since the initial release of Novell Nsure Audit 1.0. This information will help you to keep current on updates to the documentation.

If you have not used Novell Nsure Audit 1.0 or Nsure Audit 1.0.1 you do not need to review this section.

All changes that are noted in this section were also made in the documentation. The documentation is provided on the Web in two formats: HTML and PDF. The HTML and PDF documentation are both kept up-to-date with the documentation changes listed in this section.

The documentation update information is grouped according to the date the changes were published. Within a dated section, the changes are alphabetically listed by the names of the main table of contents sections for Novell Nsure Audit 1.0.3.

If you need to know whether a copy of the PDF documentation you are using is the most recent, the PDF document contains the date it was published on the front title page or in the Legal Notices section immediately following the title page.

The documentation was updated on the following dates:

- Section I.3, "November 10, 2004," on page 257
- Section I.4, "August 2, 2004," on page 258
- Section I.5, "November 26, 2003," on page 259

I.1 July 14, 2006

Updates were made to the following sections. The changes are explained below:

- Chapter 7, "Configuring System Channels," on page 69
- Appendix A, "Event Structure," on page 175
- Appendix B, "Application Events," on page 191
- Appendix C, "Using MySQL with Nsure Audit," on page 211
- Appendix D, "Using Oracle with Nsure Audit," on page 217
- Appendix E, "Using Microsoft SQL Server with Nsure Audit," on page 227
- Appendix F, "Using JDBC Data Stores with Nsure Audit," on page 231
- Appendix H, "File Descriptions and Locations," on page 245

I.1.1 Provided Additional Information About Configuring Java and JDBC Channels

More complete information is provided for the Java and JDBC configuration requirements. This information is available in the following sections:

- Section 7.6, "Java," on page 79

- Section 7.7, “JDBC,” on page 80
- Appendix F, “Using JDBC Data Stores with Nsure Audit,” on page 231

I.1.2 Provided Additional Information on Configuring Oracle, MySQL, and MSSQL Data Stores

The Oracle, MySQL, and MSSQL appendices provide more detailed information on setting up the database data stores. For more information, see the following appendices:

- Appendix C, “Using MySQL with Nsure Audit,” on page 211
- Appendix D, “Using Oracle with Nsure Audit,” on page 217
- Appendix E, “Using Microsoft SQL Server with Nsure Audit,” on page 227

I.1.3 Updated documentation on eDirectory, NetWare, File System, and Nsure Audit Events

Appendix B, “Application Events,” on page 191 provides more complete information on events logged by the eDirectory, NetWare, and Naudit Instrumentations.

I.1.4 Added the Program Files Appendix

Appendix H, “File Descriptions and Locations,” on page 245 provides a complete listing of the Nsure Audit program files and their respective directories on each platform.

I.1.5 Corrections to the Platform Agent Configuration Utility

Filename and directory information was corrected for the Platform Agent configuratino utility (`NauditPAConfig`) in the following sections:

- Section 4.1.3, “Platform Agent Configuration Tool,” on page 44
- Section G.8, “NauditPAConfig,” on page 244

I.2 June 15, 2005

Updates were made to the following sections. The changes are explained below:

- Section I.2.1, “Added Information About Adding LSC Files to Application Objects,” on page 257
- Section I.2.2, “Updated Documentation to Reflect Changes in the Nsure Audit iManager Plug-in,” on page 257
- Section I.2.3, “Provided Information on Additional Event Fields and Event Variables,” on page 257
- Section I.2.4, “Updated Channel Information,” on page 257

I.2.1 Added Information About Adding LSC Files to Application Objects

See Section 6.2, “Creating Application Objects,” on page 65.

I.2.2 Updated Documentation to Reflect Changes in the Nsure Audit iManager Plug-in

Updated screenshots and task names throughout the Administration Guide to reflect changes in the Nsure Audit iManager Plug-in.

The procedures documented in Section 3.4, “Performing Basic Administrative Functions in iManager,” on page 37 changed to reflect the changes in the Nsure Audit iManager Plug-in.

I.2.3 Provided Information on Additional Event Fields and Event Variables

See Section A.1, “Event Structure,” on page 175 and Section A.3, “Managing Event Data,” on page 181.

I.2.4 Updated Channel Information

Information for the File, Java, and JDBC channels was updated. For more information, see Chapter 7, “Configuring System Channels,” on page 69.

I.3 November 10, 2004

Update were made to the following sections. The changes are explained below:

- Section I.3.1, “Clarified JDBC Channel Set Up Instructions,” on page 257
- Section I.3.2, “Updated Troubleshooting Section,” on page 257

I.3.1 Clarified JDBC Channel Set Up Instructions

See Appendix F, “Using JDBC Data Stores with Nsure Audit,” on page 231

I.3.2 Updated Troubleshooting Section

The following topics were added to troubleshooting:

- Section 11.1.9, “Nsure Audit Events Sent During Initialization are Not Logged to the Data Store,” on page 172
- Section 11.1.10, “Nsure Identity Manager 2 DR1 Update required to Use Nsure Audit 1.0.3,” on page 173

I.4 August 2, 2004

Updates were made to the following sections. The changes are explained below.

- Section I.4.1, “Additional Database Setup Instructions,” on page 258
- Section I.4.2, “Updated Event Fields Reference,” on page 258
- Section I.4.3, “Microsoft SQL Server Support,” on page 258
- Section I.4.4, “Updated Platform Agent Configuration Reference,” on page 258
- Section I.4.5, “PAConfig Utility,” on page 258
- Section I.4.6, “WebAdmin References Removed,” on page 259

I.4.1 Additional Database Setup Instructions

Several new appendices were added to provide additional information to set up additional data stores with Nsure Audit. See:

- Appendix C, “Using MySQL with Nsure Audit,” on page 211
- Appendix D, “Using Oracle with Nsure Audit,” on page 217
- Appendix E, “Using Microsoft SQL Server with Nsure Audit,” on page 227
- Appendix F, “Using JDBC Data Stores with Nsure Audit,” on page 231

I.4.2 Updated Event Fields Reference

In Nsure Audit 1.0.2, several additional event fields were added to enhance the querying and reporting features. These updated fields are described in Appendix A, “Event Structure,” on page 175.

I.4.3 Microsoft SQL Server Support

Nsure Audit provides support for using the Microsoft* SQL Server as an event repository. See Section 7.8, “Microsoft SQL Server,” on page 83 for details.

I.4.4 Updated Platform Agent Configuration Reference

The Platform Agent now has additional parameters, contained in logevent.cfg, enabling you to specify the maximum size of the Nsure Audit event cache, and specify the action Nsure Audit takes when this limit is reached (stop logging, drop cache, or generate a warning).

See “Logevent” on page 42 for details.

I.4.5 PAConfig Utility

Details on the new graphical Platform Agent configuration utility were added. See Section G.8, “NauditPAConfig,” on page 244 for details.

I.4.6 WebAdmin References Removed

Support for the WebAdmin interface was removed in Nsure Audit 1.0.2. References to WebAdmin have been removed from this release of the documentation.

I.5 November 26, 2003

Updates were made to the following sections. The changes are explained below.

- Custom Query Macros
- eDirectory Event Instrumentation
- Linux and Solaris Startup Script Location
- Secure Logging Server on Windows XP
- Letrans Utility to Translate Raw Log Files

I.5.1 Custom Query Macros

Several new custom query variables were implemented for use in Nsure Audit Report (LReport. For a complete list, see "Custom Query Macros" on page 150.

I.5.2 eDirectory Event Instrumentation

The eDirectory log schema documentation incorrectly contained an application ID of 0001, instead of 000B. This was corrected, you can view the updated log schema documentation in Section 5.5, "eDirectory Events," on page 63.

I.5.3 Linux and Solaris Startup Script Location

The Linux and Solaris startup scripts are in new locations. See "Starting and Stopping the Secure Logging Server on Linux" on page 237 and "Starting and Stopping the Secure Logging Server on Solaris" on page 237 for details.

I.5.4 Secure Logging Server on Windows XP

Windows* XP is a client release of the Windows operating system and it is not licensed to perform server functions. Therefore, Windows XP was removed as a supported platform for the Secure Logging Server. The Platform Agent is still supported on Windows XP.

I.5.5 Letrans Utility to Translate Raw Log Files

Information about translating raw log files logged by the file channel has been added, see Section 9.3.1, "Using LETrans to Access Data Logged by the File Channel," on page 156 for details.

Glossary

ACL (Access Control List)
A list of the services available on a server. Also listed are the hosts permitted to use each service.

administrative tool
The user interface in which product configuration and management tasks are performed. For Novell® Nsure™ Audit, the administrative tools is iManager.

alert
An audible or visual alarm (such as a phone call, instant message, page, siren, flashing light, e-mail, etc.) intended to inform a system's users and administrators about a policy violation, a change in the operating conditions of a system, or some kind of error condition.

Audit policy
A policy that determines which events should be logged to the data store and how those events should be monitored.

auditing service
A distributed service that aggregates events from many sources and provides monitoring, logging, and reporting to facilitate analysis of the collected data. This is the "engine" of the product.

central data store
The data store that contains every event logged to the system. Novell Nsure Audit logs all events to the central data store before any other action is performed.

channels
The communication paths that Novell Nsure Audit uses to log system events and provide event notification.

channel object
Objects used to store the information the logging server needs to use a certain channel. For example, MySQL Channel objects include the IP address or host name of the MySQL database server, a username and password to connect to the server, the database and table names, and other relevant information. SMTP Channel objects, on the other hand, include the IP address or host name of the SMTP server, a username and password, and message information (the message recipients, sender, subject, and body).

event
Data provided to the Auditing Service to be logged. This includes any significant occurrence in the system or its logging applications such as starting and stopping services, logging users on and off, accessing resources, granting access rights, and so forth.

event collection
The process of gathering events and storing them in the central data store and filtered data stores.

event log
A collection of audit entries that make up an audit trail. Also, the destination file for audit entries or logged events.

event store
A generic reference to any collection of one or more event logs in a directory or database.

forensics
A general term referring to the preservation, identification, extraction, and documentation of computer evidence from relevant logging applications.

instrument
The process of configuring an application's events so they conform to the Novell Nsure Audit standardized event structure.

instrumentation
A logging application that can report events to Novell Nsure Audit. Logging applications must report events using the Novell Nsure Audit program's standardized event structure.

log entry
A recording of a system event in a log file, typically in a standard text or marked-up text format such as TXT, XML, and so forth.

log level
A mandatory component of an event that contains a descriptive severity level of the event. Log levels are 8 bits.

logging
Persistent storage of historical data.

logging application
A generic term used to refer to any application that logs events to the Novell Nsure Audit for the purpose of leveraging its auditing, reporting, monitoring, or notification services.

logging application certificate
The certificate at the logging application. Logging application certificates must be signed by the logging server certificate. This is done using the AudCGen utility.

logging server certificate
The certificate at the logging server.

monitor
A user interface or a collection of user interfaces for viewing the real-time status of one or more aspects of a system or set of systems. A monitor can refer to a single gauge or a cluster of gauges. See “named view” on page 263.

monitoring
The act of viewing data in real time. The data is exposed through a program or set of programs used to oversee computer-based systems and networks for the purpose of tracking usage or identifying, reporting on, and solving problems at the earliest possible stage. Typically, different tools are used to monitor individual system components, although

the individual monitors might feed information to a higher-level monitor in order to encompass an entire computing environment.

named view

A view of a set of one or more monitors that has been named and saved, and can be configured and deployed to a specific user or set of users based on their role in the organization. For example, if an administrator wanted a specific set of users to see how many new users were added to an HR application, the monitor for the HR application could be placed in a view named “New Employees.” Rights to access these named views can then be managed based on roles.

non-repudiation

Unforgeable evidence that a specific action occurred.

This differs slightly from the traditional legal meaning of non-repudiation, which refers to the irrefutable genuineness of a traditional signature. Non-repudiation services typically include non-repudiation of origin, non-repudiation of delivery, non-repudiation of receipt, and non-repudiation of submission. The purpose of non-repudiation, in conformance with ISO/IEC 13888-1, -2 and -3, is to provide verifiable proof or evidence recording of data, based on cryptographic check values generated by using symmetric or asymmetric cryptographic techniques.

Non-repudiation of approval service provides proof of whom is responsible for approval of the content of a message.

Non-repudiation of sending service provides proof of who sent a message.

Non-repudiation of origin service is a combination of approval and sending services.

Non-repudiation of submission service provides proof that a delivery authority has accepted a message for transmission.

Non-repudiation of transport service provides proof for the message originator that a delivery authority has given the message to the intended recipient.

Non-repudiation of receipt service provides proof that the recipient received a message.

Non-repudiation of knowledge service provides proof that the recipient recognized the content of a received message.

Non-repudiation of delivery service is a combination of receipt and knowledge services. It provides proof that the recipient received and recognized the content of a message.

notification

An automatically generated announcement or message intended to inform a system’s users and/or administrators about a specific condition of the system.

payload

Application-specific event data that logging applications can include in their event structure, allowing Novell Nsure Audit to log more specific data.

Platform Agent

The operating system-level agent that handles event transport between logging applications and the Secure Logging server.

policy

An organization's rules governing event logging, the data store, event notifications, reset actions, and so forth, or the implementation of these rules. Policy is usually something that is written down, such as in an operations manual. Policy is often enforced through a defined set of rules.

query results

The events returned from a data query. The information is presented in a data table; rows represent individual records and columns represent fields within those records.

query

A request for specific information from the data store. In Novell Nsure Audit, queries are made using the SQL query language.

report

A Crystal Decisions Report (*.rpt). Reports can graphically represent log data in pie charts, bar charts, and so forth.

resource

A specified right, privilege, or item that can be granted or revoked from a user.

role

A function or position with associated rights and privileges dictating the operations a user is permitted to perform. Multiple roles can be assigned to one user.

rule

Repeatable process steps that are performed in a defined order, and result in the application of a policy.

secure auditing

An auditing service where, the logged data and functioning are defensible in a court of law.

threshold

A specified point that when exceeded begins producing a specified effect or result when it is exceeded.

trigger

An act that sets in motion some course of occurrences. For example, an event that is found to be incongruent with business policy can be an impetus for action such as a notification or value reset.

UTC

Coordinated Universal Time, also know as Universal time. UTC is kept within 0.9 seconds of GMT with leap seconds that are added to or omitted from official timekeeping systems annually to compensate for changes in the rotation of the earth.

The abbreviation UTC is a language-independent international abbreviation which is neither English nor French. It means both “Coordinated Universal Time” and “Temps Universal Coordonné.”