Identity Manager Driver for Linux* and UNIX* 4.8

Implementation Guide

December 18, 2019

NetIQ

Legal Notice

THIS DOCUMENT AND THE SOFTWARE DESCRIBED IN THIS DOCUMENT ARE FURNISHED UNDER AND ARE SUBJECT TO THE TERMS OF A LICENSE AGREEMENT OR A NON-DISCLOSURE AGREEMENT. EXCEPT AS EXPRESSLY SET FORTH IN SUCH LICENSE AGREEMENT OR NON-DISCLOSURE AGREEMENT, NETIQ CORPORATION PROVIDES THIS DOCUMENT AND THE SOFTWARE DESCRIBED IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SOME STATES DO NOT ALLOW DISCLAIMERS OF EXPRESS OR IMPLIED WARRANTIES IN CERTAIN TRANSACTIONS; THEREFORE, THIS STATEMENT MAY NOT APPLY TO YOU.

For purposes of clarity, any module, adapter or other similar material ("Module") is licensed under the terms and conditions of the End User License Agreement for the applicable version of the NetIQ product or software to which it relates or interoperates with, and by accessing, copying or using a Module you agree to be bound by such terms. If you do not agree to the terms of the End User License Agreement you are not authorized to use, access or copy a Module and you must destroy all copies of the Module and contact NetIQ for further instructions.

This document and the software described in this document may not be lent, sold, or given away without the prior written permission of NetIQ Corporation and Omnibond Systems, LLC., except as otherwise permitted by law. Except as expressly set forth in such license agreement or non-disclosure agreement, no part of this document or the software described in this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, or otherwise, without the prior written consent of NetIQ Corporation and Omnibond Systems, LLC.. Some companies, names, and data in this document are used for illustration purposes and may not represent real companies, individuals, or data.

This document could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein. These changes may be incorporated in new editions of this document. NetIQ Corporation and Omnibond Systems, LLC. may make improvements in or changes to the software described in this document at any time.

U.S. Government Restricted Rights: If the software and documentation are being acquired by or on behalf of the U.S. Government or by a U.S. Government prime contractor or subcontractor (at any tier), in accordance with 48 C.F.R. 227.7202-4 (for Department of Defense (DOD) acquisitions) and 48 C.F.R. 2.101 and 12.212 (for non-DOD acquisitions), the government’s rights in the software and documentation, including its rights to use, modify, reproduce, release, perform, display or disclose the software or documentation, will be subject in all respects to the commercial license rights and restrictions provided in the license agreement.

© 2019 Omnibond Systems, LLC. All Rights Reserved. Licensed to NetIQ Corporation. Portions copyright © 2019 NetIQ Corporation. All Rights Reserved.

For information about NetIQ trademarks, see https://www.netiq.com/company/legal/.

NetIQ Trademarks

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

Third-Party Materials

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

Contents

About this Book and the Library

This guide describes implementation of the NetIQ® Identity Manager 4.8 driver for Linux and UNIX.

The driver synchronizes data from a connected Linux or UNIX system with NetIQ Identity Manager 4.8, the comprehensive identity management suite that allows organizations to manage the full user life cycle, from initial hire, through ongoing changes, to ultimate retirement of the user relationship.

Other Information in the Library

The library provides the following information resources:

Identity Manager Setup Guide

Provides overview of Identity Manager and its components. This book also provides detailed planning and installation information for Identity Manager.

Designer Administration Guide

Provides information about designing, testing, documenting, and deploying Identity Manager solutions in a highly productive environment.

User Application: Administration Guide

Describes how to administer the Identity Manager User Application.

User Application: User Guide

Describes the user interface of the Identity Manager User Application and how you can use the features it offers, including identity self-service, the Work Dashboard, role and resource management, and compliance management.

User Application: Design Guide

Describes how to use the Designer to create User Application components, including how to work with the Provisioning view, the directory abstraction layer editor, the provisioning request definition editor, the provisioning team editor, and the role catalog.

Identity Reporting Module Guide

Describes the Identity Reporting Module for Identity Manager and how you can use the features it offers, including the Reporting Module user interface and custom report definitions, as well as providing installation instructions.

Analyzer Administration Guide

Describes how to administer Analyzer for Identity Manager.

Identity Manager Common Driver Administration Guide

Provides information about administration tasks that are common to all Identity Manager drivers.

Identity Manager Driver Guides

Provides implementation information about Identity Manager drivers.

About NetIQ Corporation

We are a global, enterprise software company, with a focus on the three persistent challenges in your environment: Change, complexity and risk—and how we can help you control them.

Our Viewpoint

Adapting to change and managing complexity and risk are nothing new

In fact, of all the challenges you face, these are perhaps the most prominent variables that deny you the control you need to securely measure, monitor, and manage your physical, virtual, and cloud computing environments.

Enabling critical business services, better and faster

We believe that providing as much control as possible to IT organizations is the only way to enable timelier and cost effective delivery of services. Persistent pressures like change and complexity will only continue to increase as organizations continue to change and the technologies needed to manage them become inherently more complex.

Our Philosophy

Selling intelligent solutions, not just software

In order to provide reliable control, we first make sure we understand the real-world scenarios in which IT organizations like yours operate — day in and day out. That's the only way we can develop practical, intelligent IT solutions that successfully yield proven, measurable results. And that's so much more rewarding than simply selling software.

Driving your success is our passion

We place your success at the heart of how we do business. From product inception to deployment, we understand that you need IT solutions that work well and integrate seamlessly with your existing investments; you need ongoing support and training post-deployment; and you need someone that is truly easy to work with — for a change. Ultimately, when you succeed, we all succeed.

Our Solutions

- Identity & Access Governance
- Access Management
- Security Management
- Systems & Application Management
- Workload Management
- Service Management

Contacting Sales Support

For questions about products, pricing, and capabilities, contact your local partner. If you cannot contact your partner, contact our Sales Support team.

Worldwide:	www.netiq.com/about_netiq/officelocations.asp
United States and Canada:	1-888-323-6768
Email:	info@netiq.com
Web Site:	www.netiq.com

Contacting Technical Support

For specific product issues, contact our Technical Support team.

Worldwide:	www.netiq.com/support/contactinfo.asp
North and South America:	1-713-418-5555
Europe, Middle East, and Africa:	+353 (0) 91-782 677
Email:	support@netiq.com
Web Site:	www.netiq.com/support

Contacting Documentation Support

Our goal is to provide documentation that meets your needs. If you have suggestions for improvements, click **Add Comment** at the bottom of any page in the HTML versions of the documentation posted at www.netiq.com/documentation. You can also email Documentation-Feedback@netiq.com. We value your input and look forward to hearing from you.

Contacting the Online User Community

Qmunity, the NetIQ online community, is a collaborative network connecting you to your peers and NetIQ experts. By providing more immediate information, useful links to helpful resources, and access to NetIQ experts, Qmunity helps ensure you are mastering the knowledge you need to realize the full potential of IT investments upon which you rely. For more information, visit http://community.netiq.com.

1 Overview

The Identity Manager 4.8 driver for Linux and UNIX synchronizes data between the Identity Vault and a connected Linux or UNIX system. The driver runs on a target system, such as Linux, Solaris*, AIX*, or HP-UX*. The Identity Vault runs on any platform supported by Identity Manager and communicates with the driver on the connected system over a secure network link.

The driver uses embedded Remote Loader technology to communicate with the Identity Vault, bidirectionally synchronizing changes between the Identity Vault and the connected system. The embedded Remote Loader component, also called the driver shim, runs as a native process on the connected Linux or UNIX system. There is no requirement to install Java* on the connected system.

The driver commits changes to the connected system using customizable shell scripts that issue native system commands. The publication method uses a polling script that scans the system for changes, and a change log to save changes for subsequent publishing. Password changes are sent to the change log using the authentication module framework and are then published to the Identity Vault.

The Linux and UNIX driver uses a scriptable framework, designed so that you can easily add support for existing and future applications.

The Identity Manager 4.8 driver for Linux and UNIX combines the flexibility of the Fan-Out driver for Linux and UNIX systems as well as the bidirectional support and Identity Manager policy options available with the NIS driver. Key features of the driver include:

- Bidirectional synchronization of data without requiring Java or a separate Remote Loader
- Customizable schema to integrate all aspects of Linux and UNIX account administration
- Customizable shell scripts to handle all data to be synchronized
- Low memory and processor requirements on the Metadirectory server
- No LDAP or Fan-Out core driver configuration

The following sections present a basic overview of the Linux and UNIX driver:

- “Driver Architecture” on page 11
- “Configuration Overview” on page 14

Driver Architecture

The Linux and UNIX driver synchronizes information between the Identity Vault and the account management system (files, NIS, or NIS+) on connected Linux and UNIX systems.

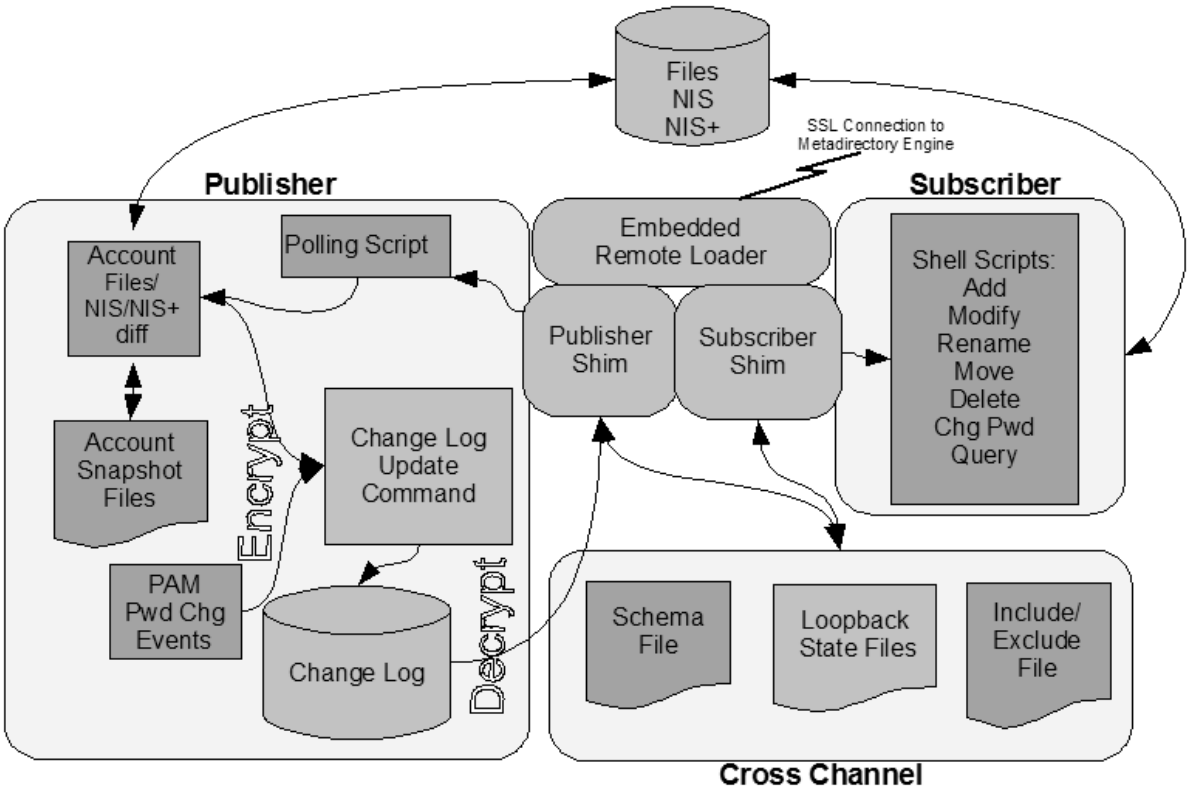
The Identity Manager detects relevant changes to identities in the Identity Vault and notifies the Subscriber component of the driver. After customizable policy processing, events are sent to the Subscriber shim of the embedded Remote Loader process on the connected system. The Subscriber shim uses shared memory to securely pass the information to customizable shell scripts that perform the required actions.

A process on the connected Linux or UNIX system polls the account management system for changes at a configurable interval. If the poll returns identity changes, they are written to the change log. An authentication module on the connected system monitors password changes and submits them to the change log.

The Publisher shim of the embedded Remote Loader process submits the changes from the change log to the Metadirectory engine as events. The Metadirectory engine processes these events using customizable policies and posts relevant changes to the Identity Vault.

The following illustration shows an overview of the architecture.

Figure 1-1 *Linux and UNIX Driver Architecture*



Publisher Channel

The Publisher shim provides identity change information to the Metadirectory engine as XDS event documents. The Metadirectory engine applies policies, takes the appropriate actions, and posts the events to the Identity Vault.

PAM and LAM

Pluggable Authentication Modules (PAM) and AIX Loadable Authentication Modules (LAM) are modules installed on the local system to intercept password changes for participating applications, such as the `passwd` command. These changes are written to the change log and are later presented to the Metadirectory engine by the Publisher shim. For details about the PAM and LAM configurations, see “PAM Configuration Details” on page 94 and “LAM Configuration Details” on page 95.

Change Log

The change log stores identity changes in encrypted form. The polling script uses the change log update command to record identity changes it detects. Password changes are written to the change log by the PAM and LAM modules. Events are removed from the change log by the Publisher shim at configurable intervals and submitted to the Metadirectory engine for processing. If communication with the Metadirectory engine is temporarily lost, events remain in the change log until communication becomes available again.

Change Log Update Command

The change log update command, `nxclh`, encrypts and writes events to the change log. Any process with rights to update the change log can use the change log update command. The change log update command takes command line arguments and standard input, and stores events in encrypted form in the change log for subsequent publishing. The polling script calls the change log update command to record identity changes. For information about using the change log update command, see the NetIQ® Identity Manager Linux and UNIX Driver Developer Kit Web site (https://www.netiq.com/documentation/identity-manager-developer/driver-developer-kit.html).

Polling Script

The polling script, `poll.sh`, is a native shell script that periodically scans the local account management system for modifications that have occurred since the last polling interval. If necessary, the polling script updates the change log by calling the change log update command. You can specify the polling interval during installation and by subsequent configuration of the Driver object.

Account Snapshot Files

The account snapshot files hold information about the state of users and groups. These files are used by the polling script to detect changes made to users and groups in the account management database (files, NIS, or NIS+).

Publisher Shim

The Publisher shim periodically scans the change log for events. Before scanning the change log, the driver calls the polling script to check the local system for changes that might have been made since the previous poll.

When the Publisher shim finds events in the change log, it decrypts, processes, and sends them to the Metadirectory engine in XDS format over a Secure Sockets Layer (SSL) network link.

Subscriber Channel

The Subscriber channel receives XDS command documents from the Metadirectory engine, stores them as name-value variables in shared memory, then calls the appropriate shell script or scripts to handle the command.

The provided shell scripts support adds, modifies, deletes, moves, and renames for User and Group objects, and handle password synchronization. You can extend the shell scripts to support other object types and events. The shell scripts have secure access to the original command data using the shared memory tool (`nxsmh`) that accesses shared memory from the driver shim.

Scriptable Framework

The interface between the account management database (files, NIS or NIS+) and the driver shim uses customizable shell scripts. You can extend the scripts that are provided with the driver to support other applications and databases.

Several utility scripts and helper commands are provided with the driver to facilitate communication with the driver shim and the change log. An extensible connected system schema file allows you to add your own objects and attributes to those already supported by the driver.

For more information about the shell scripts and the scriptable framework, see “The Scriptable Framework” on page 47.

Schema File

The configuration of class and attribute definitions for the connected Linux and UNIX system is specified using the schema file. You can modify and extend this file to include new objects and attributes. For details about configuring the schema file, see “The Connected System Schema File” on page 49.

The schema for the connected system includes two classes: User and Group. These correspond to the passwd and group maps commonly found in `/etc/passwd` and `/etc/group` in the files environment.

By default, the User class contains the attributes loginName, uidNumber, gidNumber, gecos, homeDirectory, and loginShell. These refer to the fields in the passwd map.

```
loginName:x:uidNumber:gidNumber:gecos:homeDirectory:loginShell
```

By default, the Group class contains the attributes groupName, gidNumber, and memberUid. These refer to the fields in the group map.

```
groupName:!:gidNumber:memberUid
```

Include/Exclude File

The include/exclude file allows local system policy to enforce which objects are included or excluded from provisioning, on both the Publisher channel and the Subscriber channel, independently. For details about using the include/exclude file, see “The Connected System Include/Exclude File” on page 50.

Loopback State Files

The loopback state files are used to provide automatic loopback detection for external applications that do not have mechanisms to perform loopback detection. This loopback detection prevents subscribed events from being published back to the Identity Vault.

Configuration Overview

This section discusses driver configuration details specific to the Linux and UNIX driver. For basic configuration information, see the *Identity Manager 4.8 Administration Guide*. For detailed information about configuring the Linux and UNIX driver, see Chapter 5, “Configuring the Linux and UNIX Driver,” on page 37.

Data Flow

Filters and policies control the data flow of users and groups to and from the connected system and the Identity Vault. The Data Flow option, specified during driver import, determines how these filters and policies behave.

- **Bidirectional:** Sets classes and attributes to be synchronized on both the Subscriber and Publisher channels.
- **Application to Identity Vault:** Sets classes and attributes to be synchronized on the Publisher channel only.
- **Identity Vault to Application:** Sets classes and attributes to be synchronized on the Subscriber channel only.

POSIX Information Management

The Linux and UNIX driver uses the RFC 2307 posixAccount and posixGroup attributes. You can use these classes to maintain the Linux and UNIX attributes between corresponding users and groups in the connected system and the Identity Vault.

The POSIX Information Management option, specified during driver import, provides management methods for RFC 2307 posixAccount and posixGroup attributes, such as uidNumber, gidNumber, homeDirectory, loginShell, and memberUid.

- **Manage Local:** The connected system maintains all the RFC 2307 information. RFC 2307 information is not created or stored in the Identity Vault. RFC 2307 schema extensions are not required. This option is useful for maintaining UID and GID information on multiple systems separately.
- **Manage from Identity Vault:** The Identity Vault provides and maintains all RFC 2307 information for users and groups. RFC 2307 information must be present in the Identity Vault before users and groups can be provisioned to the connected system.
- **Manage Bidirectional:** RFC 2307 information can be created and managed by both the Identity Vault and the connected system.

Filter and Schema Mapping

The Metadirectory engine uses filters to control which objects and attributes are shared. The default filter configuration for the Linux and UNIX driver allows objects and attributes to be shared as described in the following table:

Table 1-1 Default Linux and UNIX Driver Filter and Schema Mapping

eDirectory Class	eDirectory Attribute	Linux and UNIX Class	Linux and UNIX Attribute
User	CN	User	loginName
User	gecos	User	gecos
User	gidNumber	User	gidNumber
User	homeDirectory	User	homeDirectory
User	loginShell	User	loginShell
User	uidNumber	User	uidNumber
User	Group Membership	User	gidNumber
Group	CN	Group	groupName
Group	gidNumber	Group	gidNumber
Group	member	Group	memberUid

Policies

The Metadirectory engine uses policies to control the flow of information into and out of the Identity Vault. The following table describes the policy functions for the Linux and UNIX driver in the default configuration:

Table 1-2 Default Linux and UNIX Driver Policy Functions

Policy	Description
Mapping	Maps the Identity Vault User and Group objects and selected attributes to a Linux or UNIX user or group.
Publisher Event	None is provided.
Publisher Matching	Restricts privileged accounts and defines matching criteria for placement in the Identity Vault.
Publisher Create	Defines creation rules for users and groups before provisioning into the Identity Vault.
Publisher Placement	Defines where new users and groups are placed in the Identity Vault.
Publisher Command	Defines password publishing policies.
Subscriber Matching	Defines rules for matching users and groups in the connected system.
Subscriber Create	Defines required creation criteria.
Subscriber Command	Transforms RFC 2307 attributes and defines password subscribing policies.
Subscriber Output	Sends e-mail notifications for password failures and converts information formats from the Identity Vault to the connected system.
Subscriber Event	Restricts events to a specified container.

2 Planning for the Linux and UNIX Driver

This section helps you plan for deployment of the Identity Manager 4.8 driver for Linux and UNIX. Topics include

Deployment Planning

- Review Chapter 3, "Installing the Linux and UNIX Driver," on page 21 and Chapter 5, "Configuring the Linux and UNIX Driver," on page 37.
- Consider where and how you will install each component, and how you will respond to the installation script prompts and other installation decisions.
- Is this a new installation, or are you replacing a NIS driver or Fan-Out driver Platform Services installation? For details about upgrading from the NIS driver or the Fan-Out driver, see Chapter 4, "Upgrading from Another Driver," on page 33.
- How do you plan to prototype, test, and roll out your deployment?
- Do you plan to use the include/exclude file on the connected system to limit your initial deployment to a small number of users and groups?
- If you are using AIX and want to publish password changes, will you use PAM or LAM?

  AIX version 5.3 can use either PAM or LAM, but previous AIX versions must use LAM.

  LAM supports only the files database type. LAM does not support NIS and NIS+. If you have AIX 5.2 and need to support NIS or NIX+, you can do either of the following:

  - Upgrade to AIX 5.3 or newer and use PAM
  - Require users to change their passwords on the Identity Vault.

  If you have AIX 5.3 or newer, `/etc/security/login.cfg` will include a configuration setting for `auth_type`. The valid values for `auth_type` are `STD_AUTH` and `PAM_AUTH`. Within the context of the bidirectional driver, if you choose `STD_AUTH`, then you must use LAM to publish password changes. If you choose `PAM_AUTH`, then you must use PAM to publish password changes.

  NOTE: The setting you choose for `auth_type` may be influenced by reasons outside the scope of the bidirectional driver.

- If any of the systems you connect to Identity Manager are running AIX or HP-UX, you may need to replace the standard `comm` utility included with those operating systems. For more information, see "Replacing comm Utility for AIX and HP-UX" on page 22.
- Do you have NIS or NIS+ clients that you want to publish password changes from?

- What are the host names or IP addresses of all systems that will participate in your configuration?
- Will you use the default TCP port numbers?

Table 2-1 *Default TCP Port Numbers*

Purpose	TCP Port Number
Driver shim connection to Metadirectory engine	8090
Driver shim HTTP services for log viewing and access by remote NIS or NIS+ client PAM modules	8091
Secure LDAP port	636
Non-secure LDAP port	389

Migration Planning

- Where are the objects that you plan to manage with the Linux and UNIX driver currently stored?
- Can you use a Matching policy to select the objects to manage based on criteria, such as department, group membership, or some other attribute?

Customization Planning

- Do you plan to customize the shell scripts provided with the driver?

  For details about the provided scripts, see Table 6-1, “Identity Vault Command Processing Scripts,” on page 47, Table 6-2, “Other Scripts,” on page 48 and the scripts themselves.
- Do you plan to add attributes or classes to the connected system schema file?
- Do you plan to customize policies?

  For details about customizing policies, see the relevant publication(s) on the Identity Manager 4.8 Documentation Web site (https://www.netiq.com/documentation/identity-manager-47/).
- Are the resources needed to perform the customization available within your organization?

Participating Systems

You can install the components of the Identity Manager 4.8 driver for Linux and UNIX to a single system, but the components are typically installed on two systems. The driver is installed on a Metadirectory server. The driver shim is installed on the connected Linux or UNIX system. In addition, you can install the driver PAM module on NIS or NIS+ clients to publish password change information from them.

The connected system runs a lightweight process, called the driver shim or embedded Remote Loader, that communicates with the driver on the Metadirectory server over an encrypted TCP/IP network link.

The Metadirectory server and the connected system can be the same system if the system is running a version of Linux or UNIX supported as a connected system. This can be useful for testing and prototyping. Even if the Metadirectory server and the connected system are the same system, the driver is still run as a Remote Loader driver.

Choosing between the Basic and the Advanced Installation Methods

When you import the driver, you are prompted to choose either the Basic Installation or the Advanced Installation. Select **Advanced Installation** for any of the following:

- You plan to maintain RFC 2307 attribute information, such as uidNumber, gidNumber, homeDirectory, loginShell, and gecos, centrally from the Identity Vault. You can do this with a manual process or by an automated process, such as by using the Linux and UNIX Settings driver. You do not want to publish changes to this information from the Linux or UNIX system.
- You plan to maintain RFC 2307 attribute information locally on the connected Linux or UNIX system. You do not want to subscribe to changes to this information from the Identity Vault.
- You only want to publish information.
- You only want to subscribe to information.
- You want to use Role-Based Entitlements.
- You want to override the defaults and configure specific Linux and UNIX driver options, such as the automatic creation of home directories, the automatic deletion of home directories, or the setting of gecos values.

To view the driver import configuration settings offered by each installation method, see “Creating the Driver in Designer” on page 24.

Establishing a Security-Equivalent User

The driver must run with Security Equivalence to a user with sufficient rights. You can set the driver equivalent to ADMIN or a similar user. For stronger security, you can define a user with only the minimal rights necessary for the operations you want the driver to perform.

The driver user must be a trustee of the containers where synchronized users and groups reside, with the rights shown in Table 2-2. Inheritance must be set for [Entry Rights] and [All Attribute Rights].

Table 2-2 *Base Container Rights Required by the Driver Security-Equivalent User*

Operation	[Entry Rights]	[All Attribute Rights]
Subscriber notification of account changes (recommended minimum)	Browse	Compare and Read
Creating objects in the Identity Vault without group synchronization	Browse and Create	Compare and Read
Creating objects in the Identity Vault with group synchronization	Browse and Create	Compare, Read, and Write
Modifying objects in the Identity Vault	Browse	Compare, Read, and Write
Renaming objects in the Identity Vault	Browse and Rename	Compare and Read
Deleting objects from the Identity Vault	Browse and Erase	Compare, Read, and Write

Operation	[Entry Rights]	[All Attribute Rights]
Retrieving passwords from the Identity Vault	Browse and Supervisor	Compare and Read
Updating passwords in the Identity Vault	Browse and Supervisor	Compare, Read, and Write

If you do not set Supervisor for [Entry Rights], the driver cannot set passwords. If you do not want to set passwords, set the Subscribe setting for the User class nspmDistributionPassword attribute to Ignore in the filter to avoid superfluous error messages. For details about accessing and editing the filter, see the appropriate policy publication on the Identity Manager 4.8 Documentation Web site (https://www.netiq.com/documentation/identity-manager-47/).

For complete information about rights, see the *NetIQ® eDirectory™ Administration Guide*.

3 Installing the Linux and UNIX Driver

This section provides the information you need to install the NetIQ® Identity Manager 4.8 driver for Linux and UNIX.

Topics include

Before You Begin

- Review Chapter 2, “Planning for the Linux and UNIX Driver,” on page 17.
- Ensure that you have the most recent distribution, support pack, and patches for the driver.
- Review the most recent support information for the driver on the NetIQ Support Web site (http://support.netiq.com).

Required Knowledge and Skills

To successfully install, configure, and use the driver, you must have system administration skills and rights for Identity Manager and the target systems. You must be proficient with using iManager to configure Identity Manager drivers. You must be familiar with the facilities of the Linux and UNIX driver, and you must have developed a deployment plan.

For an overview of driver facilities, see Chapter 1, “Overview,” on page 11.

For information about planning for the Linux and UNIX driver, see Chapter 2, “Planning for the Linux and UNIX Driver,” on page 17.

Prerequisites

Software Requirements

For information about supported platforms and operating environments, see the Identity Manager 4.8 Drivers Documentation Web site (https://www.netiq.com/documentation/identity-manager-47-drivers). From this index page, you can select a readme file associated with the platform(s) for which you need support.

Account Management System Requirements

- ❒ Linux or UNIX systems using files (`/etc/passwd`), NIS, or NIS+ are supported.
- ❒ Either Pluggable Authentication Module (PAM), or Loadable Authentication Module (LAM) on AIX must be used if bidirectional password synchronization is desired. The driver uses PAM and LAM to intercept password changes on the connected system.

  Remote NIS and NIS+ client systems that use PAM are also supported.

You can modify the scripts to support other account management systems. Support for modified scripts is provided by the developer community.

Replacing comm Utility for AIX and HP-UX

If you use Identity Manager with a connected system running AIX or HP-UX, you may need to replace the standard `comm` utility (invoked by the `comm` command) included with the operating system. Versions of `comm` that are included with either of these operating systems have been known to fail when used with files that contain long text lines. In general, the problem occurs with text lines longer than 2000 characters.

The Identity Manager driver uses `comm` to get information from `/etc/group`. Therefore, if any of your AIX or HP-UX connected systems has an `/etc/group` file with a line that is longer than 2000 characters, you should use one of the following vendor-approved GNU packages to replace the `comm` utility:

Operating System	Vendor Name and Link to Replacement Utilities
AIX	IBM (ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/coreutils)
HP-UX	HP (http://hpux.connect.org.uk/hppd/hpux/Gnu/coreutils-8.23/)

Getting the Installation Files

1. Obtain the most recent distribution of the Identity Manager 4.8 driver for Linux and UNIX from the NetIQ Downloads Web site (https://dl.netiq.com/index.jsp).

   The -driver is part of the Identity Manager Integration Module 4.80 for Linux and UNIX.
2. Copy the appropriate driver shim installation script file listed in Table 3-1 from the distribution onto your connected system.

Table 3-1 *Linux and UNIX Installation Script Filenames*

Operating System	Architecture	Installation Script File
Linux	Intel* 32-bit	`linux_x86_driver_install.bin`

Operating System	Architecture	Installation Script File
	Intel 64-bit	`linux_x86_64_driver_install.bin`
	z Series s390x 64-bit	`linux_s390x_driver_install.bin`
Solaris	Sparc*	`solaris_sparc_driver_install.bin`
	Intel 32-bit	`solaris_x86_driver_install.bin`
AIX	Power PC*	`aix_driver_install.bin`
HP-UX	PA-RISC* 32-bit	`hpux_driver_install.bin`
	IA64* 32-bit	`hpux_ia64_driver_install.bin`

Running the Installation Script

Several of the installation procedures described in the sections that follow include running the installation script on a Linux or UNIX system.

To run the installation script:

1. Log in to the target server as `root`.
2. Enter one of the following commands as appropriate for your operating system and architecture:

```
sh linux_x86_driver_install.bin
sh linux_x86_64_driver_install.bin
sh linux_s390x_driver_install.bin
sh solaris_sparc_driver_install.bin
sh solaris_x86_driver_install.bin
sh aix_driver_install.bin
sh hpux_ia64_driver_install.bin
```

   These installation commands are self-extracting files, natively executable by the shell.
3. Optionally enter a language choice.
4. Read and accept the license agreement.
5. At the prompt, enter the installation type as directed by the procedure.

```
Select the type of installation:
 1) Install Driver Shim on Linux or UNIX system
 2) Install only PAM Module

Installation Type [1]:
```

6. Respond to the subsequent prompts as appropriate for the selected installation type.

Creating the Driver in Designer

The Linux and Unix Driver supports Designer 4 Package features, which allows you to create a driver by selecting which packages to install. After you create and configure the driver, you need to deploy it to the Identity Vault and start it.

Topics in this section include

Importing the Current Driver Packages

Driver packages can be updated at any time and are stored in the Package Catalog. Packages are initially imported into the Package Catalog when you create a project, import a project, or convert a project. It is important to verify you have the latest packages imported into the Package Catalog before you install the driver.

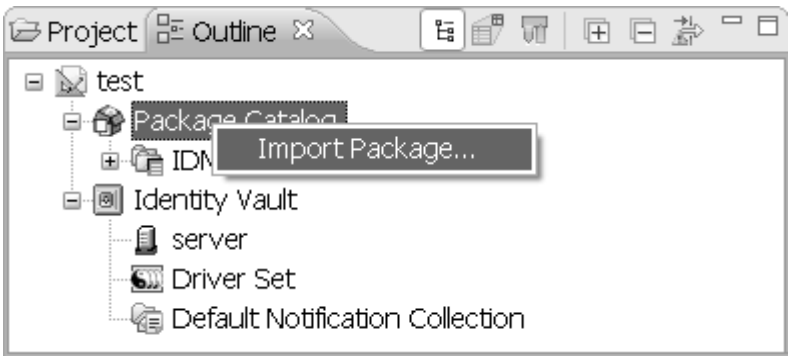
To verify you have the latest packages imported into the Package Catalog:

1. Open Designer.
2. In the toolbar, click **Help** > **Check for Package Updates**.
3. Click **OK** if there are no package updates

   or

   Click **OK** to import the package updates.
4. In the Outline view, right-click the **Package Catalog**.
5. Click **Import Package**.

   Figure 3-1 **Import Package**



6. Select the Linux and Unix Packages

   or

   Click **Select All** to import all of the packages displayed, then click **OK**.

NOTE: By default, only the base packages are displayed. Deselect **Show Base Packages Only** to display all packages.

7 Click **OK** to import the selected packages, then click **OK** in the successfully imported packages message.

8 After the current packages are imported, continue to the next section, “Installing the Driver Packages” on page 25.

Installing the Driver Packages

After you have imported the current driver packages into the Package Catalog, you can install the driver packages to create a new driver.

1 In Designer, open your project.

2 In the Modeler, right-click the driver set where you want to create the driver, then select **New** > **Driver**.

3 Select **Linux and Unix Base** from the list of base packages, then click **Next**.

4 Select the optional features to install for the Linux and Unix driver. The options are:

NOTE: Publications referenced in the following option descriptions can be accessed at the Identity Manager 4.8 Documentation Web site (https://www.netiq.com/documentation/identity-manager-47/).

Default Configuration: This package contains the default configuration information for the Linux and Unix driver. Always leave this option selected.

Entitlements: This package contains configuration information for synchronizing Linux and Unix accounts and policies that enable account creation and auditing for the Linux and Unix driver. To enable account creation and auditing, verify that this option is selected. For more information, see the *Identity Manager 4.8 Entitlements Guide*.

Password Synchronization: This package contains the policies that enable the Linux and Unix driver to synchronize passwords. To synchronize passwords, verify that this option is selected. For more information, see the *Identity Manager 4.8 Password Management Guide*.

Data Collection: This package contains the policies that enable the driver to collect data for reports. If you are using the Identity Reporting Module, verify that this option is selected. For more information, see the *Identity Reporting Module Guide*.

Account Tracking: This package contains the policies that enable you to track accounts for reports. If you are using the Identity Reporting Module, verify that this option is selected. For more information, see the *Identity Reporting Module Guide*.

5 After selecting the optional packages, click **Next**.

6 (Conditional) If the packages you selected to install have package dependencies, you must also install them to install the selected package. Click **OK** to install the listed package dependencies.

7 (Conditional) If more than one type of package dependency must be installed, you are presented with these packages separately. Continue to click **OK** to install any additional package dependencies.

8 (Conditional) The Common Settings page is displayed only if the Common Settings package is installed as a dependency. On the Install Common Settings page, fill in the following fields:

User Container: Select the Identity Vault container where Linux and Unix users will be added if they don’t already exist in the vault. This value becomes the default for all drivers in the driver set.

If you want a unique location for this driver, set the value for all drivers on this page. After the driver is created, change the value on the driver's Global Configuration Values page.

Group Container: Since the Linux and Unix driver does not synchronize Group objects, this setting can be ignored.

9 (Conditional) If not already configured, fill in the following fields on the Common Settings Advanced Edition page, then click **Next**:

User Application Provisioning Services URL: specify the User Application Identity Manager Provisioning URL.

User Application Provisioning Services Administrator: Specify the DN of the User Application Administrator user. This user should have the rights for creating and assigning resources. For more information, see "Setting Up Administrative Accounts" in the *NetIQ Identity Manager 4.8 Common Driver Administration Guide*.

10 On the Install Linux and Unix page, fill in the following field:

Driver Name: Specify a name for the driver that is unique within the driver set.

11 (Conditional) On the Driver Parameters page, review the default Subscriber and Publisher Options. Edit, if necessary, and click **Next**:

12 On the Install Linux and Unix Base page, fill in the following fields to connect to the Remote Loader and click **Next**:

Connect to Remote Loader: By default, the driver is configured to connect using the Remote Loader. You must select **Yes** for this option.

Host Name: Specify the port number where the Remote Loader is installed and is running for this driver. The default port number is 8090.

Port: Specify the Remote Loader's password as defined on the Remote Loader. The Metadirectory server (or Remote Loader shim) requires this password to authenticate to the Remote Loader.

Remote Password: Specify the Remote Loader's password as defined on the Remote Loader. The Metadirectory server (or Remote Loader shim) requires this password to authenticate to the Remote Loader.

Driver Password: Specify the driver object password that is defined in the Remote Loader service. The Remote Loader requires this password to authenticate to the Metadirectory server.

13 (Conditional) On the Entitlements page, review the default values. Edit, if necessary, and click **Next**.

14 (Conditional) On the Account Tracking page, review the default values. Edit, if necessary, and click **Next**:

15 (Conditional) This page is displayed only if you selected to install the Managed System Information packages. On the Install Linux and Unix Managed System Information page, fill in the following fields, then click **Next**:

Classification: Select the classification of the Linux and Unix system. This information is displayed in the reports. Options include:

- **Mission-Critical**
- **Vital**
- **Not-Critical**
- **Other**

If you select **Other**, you must specify a custom classification for the Linux and Unix system.

Environment: Select the type of environment the Linux and Unix system provides. Options include:

- **Development**
- **Test**
- **Staging**
- **Production**
- **Other**

If you select **Other**, you must specify a custom classification for the Linux and Unix system.

NOTE: This page is displayed only if you installed the Managed System package.

---

16 (Conditional) On the System Ownership page, fill in the following fields to define the ownership of the Linux and Unix system, then click **Next**:

Business Owner: Select a user object in the Identity Vault that is the business owner of the Linux and Unix system. This can only be a user object, not a role, group, or container.

Application Owner: Select a user object in the Identity Vault that is the application owner of the Linux and Unix system. This can only be a user object, not a role, group, or container.

17 (Conditional) On the General Information page, fill in the following fields to define your Linux and Unix system, then click **Next**:

Name: Specify a descriptive name for this Linux and Unix system. The name is displayed in reports.

Description: Specify a brief description for this Linux and Unix system. The description is displayed in reports.

Location: Specify the physical location for this Linux and Unix system. The location is displayed in reports.

Vendor: Specify the vendor of Linux and Unix. This information is displayed in reports.

Version: Specify the version of this Linux and Unix system. The version is displayed in reports.

NOTE: This page is displayed only if you installed the Managed System package.

---

18 (Conditional) On the Entitlements Name to CSV File Mappings page, click the **Add Name to File Mapping** icon to populate the page with the entitlement configuration options. Identity Manager uses the CSV file to map Linux and Unix entitlements into corresponding resources in the Identity Manager catalog.

NOTE: This page is displayed only if you installed the Entitlements package.

The information that you specify in this page is used for creating the permissions catalog. Fill in the following fields, then click **Next**:

Entitlement Name: Specify a descriptive name for the entitlement to map it to the CSV file that contains the Linux and Unix entitlement details.

Entitlement Name is the name of the entitlement. This parameter corresponds to the Entitlement Assignment Attribute in Linux and Unix. For example, you could define an entitlement called *ParkingPass*.

Entitlement Assignment Attribute: Specify a descriptive name for the assignment attribute for an entitlement.

Entitlement Assignment Attribute holds the entitlement values in Linux and Unix. For example, you could have an attribute called *Parking*.

You must add this parameter to **Field Names** in the Driver Parameters page or modify it in driver settings after creating the driver.

CSV File: Specify the location of the CSV file. This file must be located on the same server as the driver. This file contains the values for the application entitlements..

Multi-valued?: Set the value of this parameter to **True** if you want to assign resources and entitlements multiple times with different values to the same user. Otherwise, set it to **False**.

19 Review the summary of tasks that will be completed to create the driver, then click **Finish**.

The driver is created. You can modify the configuration settings by continuing with the next section, "Configuring the Driver" on page 28. If you don't need to configure the driver, skip ahead to "Deploying the Driver" on page 28.

Configuring the Driver

There are many settings that can help you customize and optimize the driver. The settings are divided into categories such as Driver Configuration, Engine Control Values, and Global Configuration Values (GCVs). Although it is important for you to understand all of the settings, your first priority should be to review the Driver Parameters located on the Driver Configuration page and the Global Configuration Values. These settings must be configured properly for the driver to start and function correctly.

To access the Driver Properties page:

1 Open your project.

2 In the Modeler, right-click the driver icon or the driver line, then select **Properties**.

3 Modify the driver settings as necessary.

IMPORTANT: In addition to the driver settings, you should review the set of default policies and rules provided by the basic driver configuration. Although these policies and rules are suitable for synchronizing with Linux and Unix*, your synchronization requirements for the driver might differ from the default policies. If this is the case, you need to change them to carry out the policies you want. The default policies and rules are discussed in "Configuration Overview" on page 14.

4 Continue with the next section, "Deploying the Driver" on page 28.

Deploying the Driver

After a driver is created in Designer, it must be deployed into the Identity Vault:

1 In Designer, open your project.

2 In the Modeler, right-click the driver icon or the driver line, then select **Live** > **Deploy**.

3 If you are authenticated to the Identity Vault, skip to Step 5; otherwise, specify the following information:

Host: Specify the IP address or DNS name of the server hosting the Identity Vault.

Username: Specify the DN of the user object used to authenticate to the Identity Vault.

Password: Specify the user's password.

4 Click **OK**.

5 Read through the deployment summary, then click **Deploy**.

6 Read the successful message, then click **OK**.

7 Click **Define Security Equivalence** to assign rights to the driver.

The driver requires rights to objects within the Identity Vault. The Admin user object is most often used to supply these rights. However, you might want to create a DriversUser (for example) and assign security equivalence to that user. Whatever rights that the driver needs to have on the server, the DriversUser object must have the same security rights:

7a Click **Add**, then browse to and select the object with the correct rights.

7b Click **OK** twice.

8 Click **Exclude Administrative Roles** to exclude users that should not be synchronized.

You should exclude any administrative User objects (for example, Admin and DriversUser) from synchronization:

8a Click **Add**, then browse to and select the user object you want to exclude.

8b Click **OK**.

8c Repeat Step 8a and Step 8b for each object you want to exclude.

8d Click **OK**.

9 Click **OK**.

Starting the Driver

When a driver is created, it is stopped by default. To make the driver work, you must start the driver and cause events to occur. Identity Manager is an event-driven system, so after the driver is started, it won't do anything until an event occurs.

To start the driver:

1 In Designer, open your project.

2 In the Modeler, right-click the driver icon or the driver line, then select **Live** > **Start Driver**.

Creating the Driver in iManager

Drivers are created with packages, and iManager does not support packages. In order to create or modify drivers, you must use Designer. See "Creating the Driver in Designer" on page 24.

Installing the Driver Shim on the Connected System

The driver shim and its files are installed into the `/usr/local/nxdrv` directory and other appropriate system locations. For details see "Files and Directories Modified by Installing the Driver Shim" on page 96.

The driver uses an embedded Remote Loader. It is not necessary to install Java on the connected system.

1 Log in to the connected system as `root`, and run the installation script.

For details, see "Running the Installation Script" on page 23.

2 When prompted for the type of installation, enter the option for **Install Driver Shim on Linux or UNIX system**.

3 Respond to additional prompts as appropriate.

3a Provide the Remote Loader and Driver object passwords that you entered when creating the driver in Step 12 on page 26.

3b Specify the Metadirectory server host name or IP address and secure LDAP port number.

These are used to secure the driver shim with SSL.

3c Install the PAM or LAM module if you intend to publish passwords from the connected system. For details, see “Installing the PAM or LAM Module” on page 30.

4 Start the driver shim.

To start the driver shim, run the appropriate command for your operating system as shown in Table 7-1, “Starting the Driver Shim,” on page 57.

Installing the PAM or LAM Module

To synchronize passwords from the connected system, you must install the PAM or LAM module on the connected system.

To synchronize passwords from client systems in a NIS or NIS+ environment, you must install the PAM module on each client system.

To install the Linux and UNIX driver PAM or LAM module:

1 Log in to the target system as `root`, and run the installation script.

For details, see “Running the Installation Script” on page 23.

2 When prompted for the type of installation, enter the option for **Install only PAM Module**.

For AIX systems, the option presented is **Install only PAM and LAM Modules**. AIX version 5.3 can use PAM, but previous AIX versions must use LAM.

3 Respond to additional prompts as appropriate.

If the driver shim is already installed, you can run the `nxdrv-config` command to reconfigure the PAM or LAM Module. For details about using the `nxdrv-config` command, see “Using the nxdrv-config Command” on page 89.

NOTE: The Red Hat* AS 2.1 and 3.0 PAM module `pam_unix.so` does not work with the Linux and UNIX driver PAM module. Edit the PAM configuration file to use `pam_pwdb.so` (located in the `/lib/security` directory) instead. For details about editing the PAM configuration file, see “PAM Configuration Details” on page 94.

Post-Installation Tasks

1 If desired, set **Startup Option** on the Driver Configuration page to **Auto start**. This causes the driver to start when the Metadirectory engine starts.

2 Set the driver shim to start automatically when the connected system starts. For details, see your operating system documentation.

3 Activate the driver.

Identity Manager and Identity Manager drivers must be activated within 90 days of installation or they shut down. At any time during the 90 days, or afterward, you can activate Identity Manager products.

For details about activating NetIQ Identity Manager Products, see the *Identity Manager 4.8 Installation Guide* on the Identity Manager 4.8 Documentation Web site (https://www.netiq.com/documentation/identity-manager-47/).

You can run the `nxdrv-config` command on the connected system at any time to change the driver shim configuration. You can configure the Remote Loader and driver passwords, SSL settings, the PAM or LAM module, and the schema. For details about using `nxdrv-config`, see “Using the nxdrv-config Command” on page 89.

Uninstalling the Driver

1. To remove the driver shim and the PAM or LAM module from the connected system, run `/usr/sbin/nxdrv-uninstall`.
2. To remove the Driver object from eDirectory, click **Delete Driver** on the Identity Manager Overview page in iManager.

4 Upgrading from Another Driver

This section provides the information you need to upgrade the NetIQ® Identity Manager 4.8 driver for Linux and UNIX from earlier versions of the driver, known as the NIS driver. It also provides information for upgrading from the Fan-Out driver.

Topics include

We recommend that you perform the upgrade in a test environment similar to your production environment before upgrading your live production systems.

Before beginning the upgrade process, review Chapter 3, "Installing the Linux and UNIX Driver," on page 21.

To prepare for installing the upgrade:

1. Verify that you have the required knowledge and skills.

   For details, see "Required Knowledge and Skills" on page 21.
2. Ensure that the prerequisites are met.

   For details, see "Prerequisites" on page 21.
3. Prepare the distribution files for installation.

   For details, see "Getting the Installation Files" on page 22.

Upgrading from the Fan-Out Driver

The Identity Manager Fan-Out driver provides one-way synchronization to a heterogeneous mix of systems including Linux and UNIX systems, and IBM* i5/OS* and z/OS* systems. The Fan-Out driver also provides authentication redirection from those systems.

Moving to the Linux and UNIX driver provides two main advantages.

- **Bidirectional Synchronization:** The Linux and UNIX driver allows synchronization from the connected Linux or UNIX system.
- **Standard Identity Manager Policies That Simplify Customization:** The Fan-Out driver makes minimal use of Identity Manager policies.

Consider the following before migrating from the Fan-Out driver to the Linux and UNIX driver.

- **Heterogeneity:** The Fan-Out driver supports operating systems in addition to Linux and UNIX. You can continue to use the Fan-Out driver for those systems while using the Linux and UNIX driver for Linux and UNIX systems.
- **Scalability:** The Fan-Out driver can fan out identities to any number of systems. The Linux and UNIX driver can replicate to only one system. (Although that system might provide account management for many computers using NIS or NIS+.)

  One Linux and UNIX driver is required for each connected system. For best performance, we recommend no more than a total of 60 drivers.

- **Authentication Redirection:** The Fan-Out driver provides authentication redirection from Linux and UNIX using PAM or LAM. The Linux and UNIX driver provides only bidirectional password synchronization.

Preparing for Migration

If necessary, migrate the UID and GID numbers from the appropriate Fan-Out driver Platform Set. You can assign RFC 2307 attributes, such as homeDirectory and loginShell, to objects in the Identity Vault.

To use the Linux and UNIX Settings driver to accomplish this:

1. Install the Linux and UNIX Settings driver on each connected Linux or UNIX system.
2. Set the properties of the Linux and UNIX Settings driver to correspond to the UID/GID ranges that were specified in the Fan-Out driver.
3. Configure the Linux and UNIX Settings driver to populate the desired RFC 2307 attributes.

For details about installing and configuring the Linux and UNIX Settings driver, see the *Linux and UNIX Settings Driver Implementation Guide* on the Identity Manager 4.8 Drivers Documentation Web site (https://www.netiq.com/documentation/identity-manager-47-drivers).

Migrating Fan-Out Driver Platform Services to the Linux and UNIX Driver

Perform the following steps on your target platform system:

1. Stop the following processes:
   - asamrcvr
   - asampsp
2. Remove the Platform Services startup scripts from `/etc/init.d`.
3. Install the driver shim on the connected system.

   For details, see “Installing the Driver Shim on the Connected System” on page 29.
4. Install the Linux and UNIX driver PAM or LAM module.

   For details, see “Installing the PAM or LAM Module” on page 30.

Configuring the Driver

1. Install and set up the Linux and UNIX driver on the Metadirectory server.

   For details, see “Creating the Driver in Designer” on page 24.
2. Make any required policy modifications.

   Create or modify an appropriate policy to use the alternative naming attribute if one was used by the Fan-Out driver. For more information about policy customization, see the policy documentation on the Identity Manager 4.8 Documentation Web site (https://www.netiq.com/documentation/identity-manager-47/).
3. Start the Linux and UNIX driver.

Click the upper right corner of the driver icon, then click **Start driver**.

4 Migrate the users to make new associations. For details, see “Migrating Identities from the Identity Vault to the Connected System” on page 44 and “Migrating Identities from the Connected System to the Identity Vault” on page 45.

Post-Migration Tasks

Perform the steps listed in “Post-Installation Tasks” on page 30.

After the new driver is operating properly, you can remove the Fan-Out driver components.

1 Delete the Platform object from the Fan-Out driver configuration.

2 On the connected system, uninstall Platform Services by removing all startup scripts and deleting the `/usr/local/ASAM` directory.

3 If this is the last platform being served by the Fan-Out driver, you can uninstall the Fan-Out core driver:

3a Remove the `ASAM` directory from the file system.

3b Remove the ASAM System container object and all of its subordinates from the tree.

3c Uninstall the Fan-Out driver plug-ins.

5 Configuring the Linux and UNIX Driver

After you have installed the Identity Manager 4.8 driver for Linux and UNIX, use the information in this section for configuration.

Topics include

Driver Parameters and Global Configuration Values

You can control the operation of the Linux and UNIX driver by modifying the properties described in the following sections.

IMPORTANT: Changing these values requires a restart of the driver.

To change import-only properties, you must re-import the driver configuration file `LinuxUnix-IDM3_5_0-V2.xml` over the existing driver. For details, see “Creating the Driver in Designer” on page 24.

To edit the properties shown on the Driver Configuration page and the Global Configuration Values page:

1. In iManager, select **Identity Manager Overview** from the Identity Manager task list on the left side of the window.
2. Navigate to your Driver Set by searching the tree or by entering its name.
3. Click the driver to open its overview.
4. Click the driver icon.
5. Select **Driver Configuration** or **Global Config Values** as appropriate.
6. Edit the property values as desired, then click **OK**.

Driver Configuration Page

Table 5-1 *Driver Configuration Page*

Property Name	Values or Format
Driver Module	Connect to Remote Loader must be selected.
“Driver Object Password” on page 38	Text Value

Property Name	Values or Format
Authentication ID	Not used by the Linux and UNIX driver.
Authentication Context	Not used by the Linux and UNIX driver.
“Remote Loader Connection Parameters” on page 38	Host name or IP address and port number of the driver shim on the connected system, and the RDN of the object with server certificate
Driver Cache Limit	The recommended value is 0 (zero).
Application Password	Not used by the Linux and UNIX driver.
“Remote Loader Password” on page 39	Text Value
Startup Option	Auto start Manual
“Database Type” on page 39	Files NIS NIS+
“Automatic Loopback Detection” on page 39	Yes No
“Remove Home Directories” on page 39	Yes No
“Create Home Directories” on page 39	Yes No
“Allow Duplicate UIDs” on page 40	Yes No
“Allow Duplicate GIDs” on page 40	Yes No
“Polling Interval” on page 40	Number of seconds
“Heartbeat Interval” on page 40	Number of seconds
“Publisher Disabled” on page 40	Yes No

Driver Object Password

The Driver object password is used by the driver shim (embedded Remote Loader) to authenticate itself to the Metadirectory engine. This must be the same password that is specified as the Driver object password on the connected system driver shim.

Remote Loader Connection Parameters

The Remote Loader Connection Parameters option specifies information that the driver uses for Secure Sockets Layer (SSL) communication with the connected system.

Table 5-2 Remote Loader Connection Parameters

Parameter	Description
`host=hostName`	Connected system host name or IP address.
`port=portNumber`	Connected system TCP port number. The default is 8090.
`kmo=objectRDN`	The RDN of the object with the server certificate signed by the tree's certificate authority. Enclose the RDN in double quotes (") if the name contains spaces.

The following is an example Remote Loader connection parameter string:

```
hostname=192.168.17.41 port=8090 kmo="SSL CertificateIP"
```

Remote Loader Password

The Remote Loader password is used to control access to the driver shim (embedded Remote Loader). This must be the same password that is specified as the Remote Loader password on the connected system driver shim.

Database Type

Database Type specifies the type of account management database that you use for your network-wide information storage.

- **Files:** Local file-based storage (`/etc/passwd`)
- **NIS:** Map-based storage
- **NIS+:** Hierarchical domain-based storage.

Automatic Loopback Detection

Specifies whether the driver shim discards events that would cause loopback conditions. This function supplements the loopback detection provided by the Metadirectory engine.

Remove Home Directories

Specifies whether the driver automatically removes home directories from the file system when users are deleted.

This option has no effect on AIX systems.

Create Home Directories

Specifies whether the driver automatically creates home directories in the file system when users are created.

This option has no effect on AIX systems. On AIX, the `add-user.sh` script uses the native AIX `mkuser` command. By default, this command creates a home directory. This setting is governed by `/usr/lib/security/mkuser.default` and `/etc/security/login.cfg`.

Allow Duplicate UIDs

Specifies whether the driver allows duplicate UIDs on the connected Linux or UNIX system.

AIX does not allow duplicate UIDs. Select **No** for AIX connected systems.

Allow Duplicate GIDs

Specifies whether the driver allows duplicate GIDs on the connected Linux or UNIX system.

AIX does not allow duplicate GIDs. Select **No** for AIX connected systems.

Polling Interval

Specifies the number of seconds that the Publisher shim waits after running the polling script and sending events from the change log to the Metadirectory engine. The default interval is 60 seconds.

Publisher Disabled

Specifies whether the Publisher shim is active.

Select **Yes** if you are using Identity Vault to Application (one-way) data flow. This saves processing time.

Heartbeat Interval

Specifies how often, in seconds, the driver shim contacts the Metadirectory engine to verify connectivity. Specify 0 to disable the heartbeat.

Global Configuration Values Page

Table 5-3 *Global Configuration Values*

Property Name	Values or Format
“Connected System or Driver Name” on page 41	Text Value
“Synchronize Group Membership” on page 41	Yes No
“Exclude Privileged Users and Groups” on page 41	Yes No
“Require POSIX Attributes When Subscribing” on page 42	Yes No
“Use First Name + Last Name for gecos” on page 42	Yes No
“Lower Case CNs” on page 42	Yes No
“The Linux or UNIX Connected System Accepts Passwords from the Identity Vault” on page 42	Yes No

Property Name	Values or Format
“The Identity Vault Accepts Passwords from the Linux or UNIX Connected System” on page 42	Yes No
“The Identity Vault Accepts Administrative Password Resets from the Linux or UNIX Connected System” on page 42	Yes No
“Publish Passwords to NDS Password” on page 42	Yes No
“Publish Passwords to Distribution Password” on page 42	Yes No
“Require Password Policy Validation before Publishing Passwords” on page 43	Yes No
Reset User’s External System Password to the Identity Manager Password on Failure	Yes No
“Notify the User of Password Synchronization Failure via E-Mail” on page 43	Yes No
“User Base Container” on page 43	Identity Vault Container object
“Group Base Container” on page 43	Identity Vault Container object

To view and edit Password Management GCVs, select **Show** for **Show Password Management Policy**.

To view and edit User and Group Placement GCVs, select **Show** for **Show User and Group Placements**.

Connected System or Driver Name

Specifies the name of the driver. This value is used by the e-mail notification templates.

Synchronize Group Membership

This option does not apply if the POSIX Management Mode is set to Manage Local. When it does apply, it has the following effect:

- It specifies whether the driver synchronizes the Group Membership attribute of a corresponding Group object in the Identity Vault (if one exists with that GID).
- The driver always synchronizes a user’s GID number (primary group identification) to the RFC 2307 gidNumber attribute of the corresponding User object in the Identity Vault.

Exclude Privileged Users and Groups

Specifies whether the driver excludes events for users and groups with a uidNumber or gidNumber less than 100.

Require POSIX Attributes When Subscribing

This option does not apply if the POSIX Management Mode is set to Manage Local. When it does apply, it specifies whether the driver requires users and groups from the Identity Vault to have RFC 2307 information, such as uidNumber, gidNumber, and homeDirectory, before it provisions them to the connected Linux or UNIX system.

Use First Name + Last Name for gecos

Specifies whether the driver creates the user gecos field from the First Name and Last Name attributes of the User object in the Identity Vault for subscribed events.

Lower Case CNs

Specifies whether the driver uses lowercase for the CN of User and Group objects it receives in events from the Metadirectory engine.

Linux and UNIX user and group names are usually lowercase.

The Linux or UNIX Connected System Accepts Passwords from the Identity Vault

Specifies whether the driver allows passwords to flow from the Identity Vault to the connected Linux or UNIX system.

The Identity Vault Accepts Passwords from the Linux or UNIX Connected System

Specifies whether the driver allows passwords to flow from the connected Linux or UNIX system to the Identity Vault.

The Identity Vault Accepts Administrative Password Resets from the Linux or UNIX Connected System

Specifies whether the driver allows passwords to be reset from the connected Linux or UNIX system in the Identity Vault. The `root` user can use the `passwd` command to set another user's password.

Publish Passwords to NDS Password

Specifies whether the driver uses passwords from the connected Linux or UNIX system to set non-reversible NDS® passwords in the Identity Vault.

Publish Passwords to Distribution Password

Specifies whether the driver uses passwords from the connected Linux or UNIX system to set NMAS™ Distribution Passwords, which are used for Identity Manager password synchronization.

Require Password Policy Validation before Publishing Passwords

Specifies whether the driver applies NMAS password policies to published passwords. If so, a password is not written to the Identity Vault if it does not conform.

Reset User's External System Password to the Identity Manager Password on Failure

Specifies whether, on a publish Distribution Password failure, the driver attempts to reset the password on the connected Linux or UNIX system using the Distribution Password from the Identity Vault.

Notify the User of Password Synchronization Failure via E-Mail

Specifies whether the driver sends an e-mail to a user if the password cannot be synchronized.

User Base Container

Specifies the base container object in the Identity Vault for user synchronization. This container is used in the Subscriber channel Event Transformation policy to limit the Identity Vault objects being synchronized. This container is used in the Publisher channel Placement policy as the destination for adding objects to the Identity Vault. Use a value similar to the following:

```
users.myorg
```

Group Base Container

Specifies the base container object in the Identity Vault for group synchronization. This container is used in the Subscriber channel Event Transformation policy to limit the Identity Vault objects being synchronized. This container is used in the Publisher channel Placement policy as the destination when adding objects to the Identity Vault. Use a value similar to the following:

```
groups.myorg
```

The Driver Shim Configuration File

The driver shim configuration file `/etc/nxdrv.conf` controls operation of the driver shim. You can specify the configuration options listed in Table 5-4, one per line. You can also specify these options on the driver shim command line. For details about driver shim command line options, see "Driver Shim Command Line Options" on page 93.

Table 5-4 *Driver Shim Configuration File Statements*

Option (Short and Long Forms)	Description

`-conn <connString>`
`-connection <connString>`
	A string with connection options. Enclose the string in double quotes ("). If you specify more than one option, separate the options with spaces.
	`port=<driverShimPort>`
	`ca=<Certificate Authority Key File>`

Option (Short and Long Forms)	Description
`-hp <httpPort>` `-httpport <httpPort>`	Specifies the HTTP services port number. The default HTTP services port number is 8091. You can connect to this port to view log files. For details, see "The Trace File" on page 64 and "The Status Log" on page 65.
`-path <driverPath>`	Specifies the path for driver files. The default path is `/usr/local/nxdrv`.
`-sp <password>` `-setpassword <password>`	Sets the Remote Loader and Driver object passwords.
`-t <traceLevel>` `-trace <traceLevel>`	Sets the level of debug tracing. 0 is no tracing, and 10 is all tracing. For details, see "The Trace File" on page 64. The output file location is specified by the `tracefile` option.
`-tf <fileName>` `-tracefile <fileName>`	Sets the trace file location. The default is `/usr/local/nxdrv/logs/trace.log`.

Example /etc/nxdrv.conf File

```
-tracefile /usr/local/nxdrv/logs/trace.log
-trace 0
-connection "ca=/usr/local/nxdrv/keys/ca.pem port=8090"
-httpport 8091
-path /usr/local/nxdrv/
```

Migrating Identities

When you first run the Linux and UNIX driver, you might have identities in the Identity Vault that you want to provision to the connected system, or vice versa. Identity Manager provides a built-in migration feature to help you accomplish this.

Migrating Identities from the Identity Vault to the Connected System

1. In iManager, open the Identity Manager Driver Overview for the driver.
2. Click **Migrate from Identity Vault**. An empty list of objects to migrate is displayed.
3. Click **Add**. A browse and search dialog box that allows you to select objects is displayed.
4. Select the objects you want to migrate, then click **OK**.

To view the results of the migration, click **View the Driver Status Log**. For details about the log, see "The Status Log" on page 65.

If a user has a Distribution Password, the Distribution Password is migrated to the connected system as the user's password. Otherwise, no password is migrated. For information about Universal Passwords and Distribution Passwords, see the *Password Management Administration Guide* (https://www.netiq.com/documentation/password_management33/).

Migrating Identities from the Connected System to the Identity Vault

1. In iManager, open the Identity Manager Driver Overview for the driver.
2. Click **Migrate into Identity Vault** to display the Migrate Data into the Identity Vault window.
3. Specify your search criteria:
   - **3a** To view the list of eDirectory™ classes and attributes, click **Edit List**.
   - **3b** Select class User or class Group.

     IMPORTANT: Identity Manager imports objects by class in the order specified in the list. Migrate users before you migrate groups so that the users can be added to the newly created groups.

   - **3c** Select the attributes to be used as search criteria for objects of the selected class, then click **OK**.

     The eDirectory attributes map to Linux and UNIX attributes as specified by the driver schema: CN maps to loginName, etc. For the default mappings, see Table 1-1, "Default Linux and UNIX Driver Filter and Schema Mapping," on page 16.

     To see RFC 2307 attributes, click **Show all attributes from all classes** above the attribute list.

   - **3d** Specify values for the selected attributes, then click **OK**.

     The values can include basic regular expressions. For details about basic regular expressions, use the `man grep` command.
4. Click **OK**.

To view the results of the migration, click **View the Driver Status Log**. For details about the log, see "The Status Log" on page 65.

Because local passwords are irreversibly encrypted, they cannot be submitted to the Metadirectory engine until they are changed. Install the PAM or LAM module to capture password changes. For information about installing the PAM or LAM module, see "Installing the PAM or LAM Module" on page 30.

Synchronizing the Driver

To generate events for associated objects that have changed since the driver's last processing, open the Identity Manager Driver Overview page for the driver in iManager, then click **Synchronize**.

6 Customizing the Linux and UNIX Driver

This section provides information about available resources for customizing the Identity Manager 4.8 driver for Linux and UNIX.

Topics include

For details about the filters and policies provided with the Linux and UNIX driver, see “Filter and Schema Mapping” on page 15 and “Policies” on page 16.

The Scriptable Framework

The Linux and UNIX driver provides a comprehensive scriptable framework that you can use to add to the built-in support for files, NIS, and NIS+, and to add support for other applications.

The Linux and UNIX driver scriptable framework includes components that simplify the job of extending the driver to support new applications.

- Embedded Remote Loader
  - Full SSL support, and an installer to easily configure the certificates
  - Web access to debugging information from the embedded Remote Loader
- Encrypted change log that stores changes from the application to the Identity Vault if there is a communication problem
- Loopback detection system to prevent subscribed events from being published back to the Identity Vault
- Shared memory helper programs that provide for securely passing large variables to and from the scripts
- Easily extendable connected system schema file to support any application
- Include/exclude file for simplified testing and deployment by the platform administrator
- Event support, both for applications that have exits or callouts, and for applications that must be polled for changes

The names of objects and attributes in the scripts are the names specified in the connected system schema file.

The following tables describe the major script files.

Table 6-1 *Identity Vault Command Processing Scripts*

Script File	Identity Vault Event
`add-group.sh`	Add Group

Script File	Identity Vault Event
add-group-member.sh	Add Group Member
add-user.sh	Add User
delete-group.sh	Delete Group
delete-user.sh	Delete User
disable-user.sh	Disable User
enable-user.sh	Enable User
modify-group.sh	Modify Group
modify-password.sh	Password Change
modify-user.sh	Modify User
query-read-group.sh	Entry Query for Group
query-read-user.sh	Entry Query for User
query-search-group.sh	Subtree Query for Group
query-search-user.sh	Subtree Query for User
remove-group-member.sh	Remove Group Member
rename-group.sh	Rename Group
rename-user.sh	Rename User

Table 6-2 *Other Scripts*

Script File	Purpose
subscriber.sh	Sets up file path locations. Calls the appropriate shell script based on the type of event and object.
poll.sh	Examines the account management system files to detect changes.
idmlib.sh	Contains a function library to help the scripts access and manipulate Identity Manager data.
heartbeat.sh	Sends a status document to report the health of the application.
globals.sh	Holds configurable options that all shell scripts can use during event processing.
association.sh	Generates an association for a user or group.

The Connected System Schema File

The schema file on the connected system at `/usr/local/nxdrv/schema/schema.def` is used to specify the classes and attributes that are available on the system.

The schema file is read by the driver shim when the Metadirectory engine requests it. This typically happens at driver startup. The schema file is also used by the Policy Editor to map the schema of the Identity Vault to the schema of the external application.

If you change the schema file, you must restart the driver shim and the driver.

The scripts that are provided with the driver depend on the classes and attributes in the schema file that is provided with the driver.

Schema File Syntax

Each line in the schema file represents an element and must begin with the element name: `SCHEMA`, `CLASS`, or `ATTRIBUTE`.

The first element of the schema file is the schema definition. The schema definition is followed by class definitions. Each class definition can contain attribute definitions.

Except for the values of class and attribute names, the contents of the schema file are case insensitive.

Comments

Lines that begin with an octothorpe (#) are comments.

```
# This is a comment.
```

Schema Definition

The first line in the schema file that is not a comment must be the schema definition.

```
SCHEMA [HIERARCHICAL]
```

`HIERARCHICAL` specifies that the target application is not a flat set of users and groups, but is organized by hierarchical components, such as a directory-based container object.

Class Definition

`CLASS` *`className`* `[CONTAINER]`

You must specify a class name. Enclose the class name in double quotes (").

Add the `CONTAINER` keyword if objects of this class can contain other objects.

The class definition is ended by another class definition or by the end of the file.

Attribute Definition

Any number of attribute definitions can follow a class definition. Attribute definitions define attributes for the class whose definition they follow.

`ATTRIBUTE` *`attributeName`* `[`*`TypeAndProperties`*`]`

An attribute name is required. Enclose the attribute name in double quotes (").

If no attribute type is specified, the attribute has the string type. The allowable types are

- `STRING`
- `INTEGER`
- `STATE`
- `DN`

The allowable attribute properties are

- `REQUIRED`
- `NAMING`
- `MULTIVALUED`
- `CASESENSITIVE`
- `READONLY`

Example Schema File

```
SCHEMA HIERARCHICAL
   CLASS "User"
      ATTRIBUTE "cn" NAMING REQUIRED
      ATTRIBUTE "Group Membership" MULTIVALUED DN
   CLASS "Group"
      ATTRIBUTE "cn" NAMING REQUIRED
      ATTRIBUTE "Group Members" MULTIVALUED DN
```

The Connected System Include/Exclude File

You can use an optional include/exclude file on the connected system to control which identities are or are not synchronized between the Identity Vault and the connected system. The include/exclude file is located in `/usr/local/nxdrv/conf/include-exclude.conf`.

The file is read when the driver shim starts. If you make changes to it, you must restart the driver shim.

The include/exclude file can contain include rules and exclude rules. To ensure optimal performance, each include/exclude file should contain no more than 50 entries total.

A default file that excludes many common Linux and UNIX user IDs and groups, such as `root`, is created by the installation process.

You can use the include/exclude file to phase in your deployment of the Linux and UNIX driver, excluding most users and groups at first, and then adding more as you gain confidence and experience.

- “Include/Exclude Processing” on page 51
- “Include/Exclude File Syntax” on page 51
- “Example Include/Exclude Files” on page 54

Include/Exclude Processing

Identity Vault events for identities that match an exclude rule are discarded by the Subscriber shim. Local events for identities that match an exclude rule are not sent to the Metadirectory engine by the Publisher shim.

Included identities are treated normally by the Subscriber and Publisher shims.

Identities that do not match an include rule or an exclude rule in the file are included.

Identities are matched in the following priority:

1. Channel-specific (Publisher or Subscriber) exclude rules
2. Channel-specific include rules
3. General exclude rules
4. General include rules

Within each level of this matching priority, identities are matched against rules in the order that the rules appear in the file. The first rule that matches determines whether the identity is included or excluded.

Include/Exclude File Syntax

Except for class names, attribute names, and the values to match, the contents of the include/exclude file are case insensitive.

The include/exclude file can contain any number of include sections, exclude sections, and single-line rules.

Include sections and exclude sections can contain class matching rules, and class matching rules can contain attribute matching rules. Include sections and exclude sections can also contain association matching rules.

Include and exclude sections can be contained in subscriber and publisher sections to limit their scope to the specified channel.

Class and attribute names used in the include/exclude file must correspond to the names specified in the schema file. For details about the schema file, see “The Connected System Schema File” on page 49.

Comments

Lines that begin with an octothorpe (#) are comments.

```
# This is a comment.
```

Subscriber and Publisher Sections

Subscriber and publisher sections limit the include and exclude sections they contain to the specified channel.

A subscriber section begins with a subscriber line and ends with an endsubscriber line.

```
SUBSCRIBER
  .
  .
  .
ENDSUBSCRIBER
```

A publisher section begins with a publisher line and ends with an endpublisher line.

```
PUBLISHER
  .
  .
  .
ENDPUBLISHER
```

Each subscriber and publisher section can contain include and exclude sections.

Include and Exclude Sections

Include and exclude sections provide rules to specify which objects are to be included or excluded from synchronization.

An include section begins with an include line and ends with an endinclude line.

```
INCLUDE
  .
  .
  .
ENDINCLUDE
```

An exclude section begins with an exclude line and ends with an endexclude line.

```
EXCLUDE
  .
  .
  .
ENDEXCLUDE
```

You can use class matching rules and association matching rules within an include section and an exclude section.

Class Matching Rules

Use a class matching rule within an include section or an exclude section to specify the name of a class of objects to include or exclude.

A class matching rule is defined by a class line that specifies the name of the class and ends with an endclass line.

```
CLASS className
  .
  .
  .
ENDCLASS
```

You can use attribute matching rules within a class matching rule.

Attribute Matching Rules

You can use attribute matching rules within a class matching rule to limit the objects that are included or excluded. If no attribute matching rules are specified for a class, all objects of the specified class are included or excluded.

An attribute matching rule comprises an attribute name, an equals sign (=), and an expression. The expression can be an exact value, or it can use limited regular expressions. For details about limited regular expressions, see “Limited Regular Expressions” on page 54.

attributeName=expression

Multiple attribute matching rules can be specified for a given class.

Attribute matching rules within a class matching rule are logically ANDed together. To logically OR attribute matching rules for a class, specify multiple class matching rules. For example, the following include/exclude file excludes both user01 and user02:

```
# Exclude the User object if its loginName is user01 or user02.
EXCLUDE
CLASS User
    loginName=user01
ENDCLASS
CLASS User
    loginName=user02
ENDCLASS
ENDEXCLUDE
```

Association Matching Rules

You can specify association matching rules in an include or exclude section. Association matching rule expressions can specify an exact association or a limited regular expression. For details about limited regular expressions, see “Limited Regular Expressions” on page 54.

By default, an association is formed by concatenating the object name and the class name. Association formation can be customized in the Subscriber scripts.

For example, to exclude the `root` user, specify

```
EXCLUDE
  rootUser
ENDEXCLUDE
```

Single-Line Rules

`[SUBSCRIBER|PUBLISHER] INCLUDE|EXCLUDE [`*`className`*`] `*`objectSelection`*

Where *objectSelection* can be

{*associationMatch* | *attributeName=expression*}

Single-line rules can specify the Subscriber or Publisher channel at the start of the rule. If a channel is specified, the rule applies only to that channel. Otherwise it applies to both channels.

You must specify whether the rule is to include or exclude the objects it matches.

You can specify a class name to limit matches to only objects of that class.

You must specify either an association or an attribute matching expression. The syntax of the association and attribute matching expression is the same as that of association matching rules and attribute matching rules previously described. For details, see “Association Matching Rules” on page 53 and “Attribute Matching Rules” on page 53.

For example, to ignore events from the ADMIN user in the Identity Vault:

```
# Do not subscribe to events for the ADMIN user.
SUBSCRIBER EXCLUDE adminUser
```

Limited Regular Expressions

A limited regular expression is a pattern used to match a string of characters.

Character matching is case sensitive.

Any literal character matches that character.

A period (`.`) matches any single character.

A bracket expression is a set of characters enclosed by left (`[`) and right (`]`) brackets that matches any listed character. Within a bracket expression, a range expression is a pair of characters separated by a hyphen, and is equivalent to listing all of the characters that sort between the given characters. For example, `[0-9]` matches any single digit.

An asterisk (`*`) indicates that the preceding item is matched zero or more times.

A plus sign (`+`) indicates that the preceding item is matched one or more times.

A question mark (`?`) indicates that the preceding item is matched zero or one times.

You can use parentheses to group multiple expressions into a single item. For example, `(abc)+` matches `abc`, `abcabc`, `abcabcabc`, etc. Nesting of parentheses is not supported.

Example Include/Exclude Files

Example 6-1 *Example 1*

```
# Exclude users whose names start with temp
EXCLUDE
    CLASS User
        loginName=temp.*
    ENDCLASS
ENDEXCLUDE
```

Example 6-2 *Example 2*

```
# Exclude usera and userb
# Because attribute rules are ANDed, these must be in separate
# CLASS sections.
EXCLUDE
    CLASS User
        loginName=usera
    ENDCLASS
    CLASS User
        loginName=userb
    ENDCLASS
ENDEXCLUDE
```

Example 6-3 *Example 3*

```
# Exclude all users except those whose names start with idm
# This works because channel-specific matching takes precedence
# over general matching.
EXCLUDE
    CLASS User
    ENDCLASS
ENDEXCLUDE

SUBSCRIBER INCLUDE User loginName=idm.*
PUBLISHER INCLUDE User loginName=idm.*
```

Managing Additional Attributes

You can add additional attributes to the driver for both the Publisher and Subscriber channels. These attributes can be accessed by the scripts for all event types.

To publish or subscribe to additional attributes, you must add them to the filter and add support for them into the scripts.

Modifying the Filter

1. On the iManager Driver Overview page for the driver, click the **Filter** icon on either the Publisher or Subscriber channel. It is the same object.
2. In the Filter Edit dialog box, click the class containing the attribute to be added.
3. Click **Add Attribute**, then select the attribute from the list.
4. Select the flow of this attribute for the Publisher and Subscriber channels.
   - **Synchronize:** Changes to this object are reported and automatically synchronized.
   - **Ignore:** Changes to this object are not reported and not automatically synchronized.
   - **Notify:** Changes to this object are reported, but not automatically synchronized.
   - **Reset:** Resets the object value to the value specified by the opposite channel. (You can set this value on either the Publisher or Subscriber channel, but not both.)
5. Click **Apply**.

If you want to map this attribute to an existing attribute in the Linux and UNIX schema, modify the Schema Mapping policy for the driver.

For complete details about managing filters and Schema Mapping policies, see the policy documentation on the Identity Manager 4.8 Documentation Web site (https://www.netiq.com/documentation/identity-manager-47/).

Modifying the Scripts for New Attributes

In the Subscriber channel, a specific shell script is called to take the appropriate action for each type of event. If the additional attribute is required for adds and modifies of users, modify `add-user.sh` and `modify-user.sh` to process the additional attribute.

Publishing additional attributes requires that you act on changes made in the Linux or UNIX source application.

7 Using the Linux and UNIX Driver

This section provides information about operational tasks commonly used with the Identity Manager 4.8 driver for Linux and UNIX.

Topics include

Starting and Stopping the Driver

To start the driver:

1. In iManager, navigate to the Driver Overview for the driver.
2. Click the upper right corner of the driver icon.
3. Click **Start driver**.

To stop the driver:

1. In iManager, navigate to the Driver Overview for the driver.
2. Click the upper right corner of the driver icon.
3. Click **Stop driver**.

Starting and Stopping the Driver Shim

To start the driver shim, use the command appropriate for your operating system as shown in the following table:

Table 7-1 *Starting the Driver Shim*

Operating System	Command
AIX	`/etc/rc.d/init.d/nxdrvd start`
HP-UX	`/sbin/init.d/nxdrvd start`
Linux	`/etc/init.d/nxdrvd start`
Solaris	`/etc/init.d/nxdrvd start`

To stop the driver shim, use the command appropriate for your operating system as shown in the following table:

Table 7-2 Stopping the Driver Shim

Operating System	Command
AIX	`/etc/rc.d/init.d/nxdrvd stop`
HP-UX	`/sbin/init.d/nxdrvd stop`
Linux	`/etc/init.d/nxdrvd stop`
Solaris	`/etc/init.d/nxdrvd stop`

Displaying Driver Shim Status

To see status and version information for the driver shim, use the appropriate command for your operating system as shown in the following table:

Table 7-3 Displaying the Status of the Driver Shim

Operating System	Command
AIX	`/etc/rc.d/init.d/nxdrvd status`
HP-UX	`/sbin/init.d/nxdrvd status`
Linux	`/etc/init.d/nxdrvd status`
Solaris	`/etc/init.d/nxdrvd status`

Monitoring Driver Messages

The Linux and UNIX driver writes messages to the system log. Monitor driver activity there in the same way you monitor other key system functions. For details about the messages written by the driver, see Appendix B, “System and Error Messages,” on page 71.

Changing Passwords

To publish password change information, you must change passwords with a method that uses PAM or LAM. The driver obtains password change information through PAM and LAM.

To set a password, use `passwd`, not `yppasswd` or `passwd -r`. `yppasswd` and `passwd -r` bypass the authentication module.

Do not specify a password with `useradd`. This bypasses the authentication module.

For more information about the driver PAM and LAM modules, see “PAM Configuration Details” on page 94 and “LAM Configuration Details” on page 95.

8 Securing the Linux and UNIX Driver

The section describes best practices for securing the Identity Manager 4.8 driver for Linux and UNIX. Topics include

For additional information about Identity Manager security, see the *NetIQ® Identity Manager 4.8 Administration Guide* on the Identity Manager 4.8 Documentation Web site (https://www.netiq.com/documentation/identity-manager-47/).

Using SSL

Enable SSL for communication between the Metadirectory engine and the driver shim on the connected system.

If you don’t enable SSL, you are sending information, including passwords, in the clear.

Physical Security

Keep your servers in a physically secure location with access by authorized personnel only.

Network Security

Require users outside of the corporate firewall to use a VPN to access corporate data.

Auditing

Track changes to sensitive information. Examine audit logs periodically.

For details about using NetIQ Audit to monitor driver operation, see the NetIQ Audit Documentation Web site (http://www.novell.com/documentation/novellaudit20/index.html).

Driver Security Certificates

SSL uses security certificates to control, encrypt, and authenticate communications.

Ensure that the security certificate directory `/usr/local/nxdrv/keys` is appropriately protected. The installation program sets secure file permissions for this directory.

The Driver Shim and the Identity Manager engine communicate through SSL using a certificate created in the Identity Vault and retrieved by the driver shim during the installation process. For more information on this certificate and how to renew or install third-party certificates, refer to the *Identity Manager Administration Guide*.

The Embedded Remote Loader web interface uses a dynamically generated, self-signed certificate for SSL communication. The details of this certificate are as follows:

Table 8-1 *Security Ceritficate Details (Embedded Remote Loader)*

Property Name	Values / Parameters
Subject	SSL Server
Issuer	SSL Server
Validity	1 year
Serial Number	0
Key	1024-bit RSA

Renewal of this certificate automatically occurs every time the driver shim is restarted on the connected platform.

If you have configured your Driver Shim to provide remote NIS or NIS+ clients with password publishing, a certificate is generated during installation for SSL authorization and communication. This certificate is a self-signed certificate authority with the following certificate properties:

Table 8-2 *Security Ceritficate Details (Driver Shim)*

Property Name	Values / Parameters
Subject	soap api certificate authority
Issuer	soap api certificate authority
Validity	10 year
Serial Number	0
Key	4096-bit RSA

These properties can be configured and renewed at any time. For information on how to configure these properties, refer to “The Remote Publisher Configuration File” on page 91.

When remote NIS or NIS+ clients are configured to publish passwords, they retrieve a certificate from the Driver Shim and use this for SSL communication and client authorization. The client certificates contain the following certificate properties:

Table 8-3 Security Ceritficate Details (NIS or NIS+ clients)

Property Name	Values / Parameters
Subject	soap api client
Issuer	soap api certificate authority
Validity	2 year
Serial Number	[starts at 1000]
Key	2048-bit RSA

For more information on how to configure these certificate properties, refer to “The Remote Publisher Configuration File” on page 91.

Driver Shell Scripts

The driver uses shell scripts to perform updates on the connected system, and to collect changes made there.

Ensure that the script directory `/usr/local/nxdrv/scripts` is appropriately protected. The installation program sets secure file permissions for this directory.

The Change Log

The change log file contains information about events on the connected system, including passwords. It is encrypted, but it should be protected against access by unauthorized users.

Ensure that the change log directory `/usr/local/nxdrv/changelog` is appropriately protected. The installation program sets secure file permissions for this directory.

Driver Passwords

Use strong passwords for the Driver object and Remote Loader passwords, and restrict knowledge of them to authorized personnel. These passwords are stored in encrypted form in the security certificate directory `/usr/local/nxdrv/keys`. The installation program sets secure file permissions for this directory.

Driver Code

Ensure that the driver executable directory `/usr/local/nxdrv/bin` and the driver files in `/usr/sbin` are appropriately protected. The installation program sets secure file permissions for this directory and for the driver files added to `/usr/sbin`.

Administrative Users

Ensure that accounts with elevated rights on the Metadirectory system, Identity Vault systems, and the connected systems are appropriately secure. Protect administrative user IDs with strong passwords.

Connected Systems

Ensure that connected systems can be trusted with account information, including passwords, for the portion of the tree that is configured as their base containers.

A Troubleshooting

This section provides information about troubleshooting the Identity Manager 4.8 driver for Linux and UNIX. Topics include

Driver Status and Diagnostic Files

There are several log files that you can view to examine driver operation.

The System Log

The system log is used by the driver shim to record urgent, informational, and debug messages. Examining these should be foremost in your troubleshooting efforts. For detailed message documentation, see Appendix B, “System and Error Messages,” on page 71.

The location for the system log varies from system to system and is generally configured through `/etc/syslog.conf`. The amount of information that is logged by the driver can also be configured through this system log configuration file. The following is a sample fragment from `/etc/syslog.conf`:

```
# sample /etc/syslog.conf
#
*.err;kern.notice;auth.notice                   /dev/sysmsg
*.err;kern.debug;daemon.notice;mail.crit        /var/adm/messages

*.alert;kern.err;daemon.err                     operator
*.alert                                         root
```

The options in the first column determine which messages are logged. The options in the second column specify the destination file or user to send the log output to. For example, specifying `*.err` logs all messages with a priority of err or above. For more information about syslog priorities, view your system documentation using the `man syslog` command.

Messages from the Linux and UNIX driver shim and messages from the scripts are logged with various priorities as shown in Table A-1 on page 64. The information that is recorded depends on your syslog configuration.

Table A-1 Message Priorities

Message Topic	Priority
Script being called	DEBUG
Successful Linux or UNIX command execution	INFO
Publication events	INFO
Failures	ERR

The Trace File

The default trace file exists on the connected Linux and UNIX system at `/usr/local/nxdrv/logs/trace.log`. A large amount of debug information can be written to this file. Use the trace level setting in `/etc/nxdrv.conf` to control what is written to the file. For details about `/etc/nxdrv.conf`, see "The Driver Shim Configuration File" on page 43.

Table A-2 Driver Shim Trace Levels

Trace Level	Description
0	No debugging.
1–3	Identity Manager messages. Higher trace levels provide more detail.
4	Previous level plus Remote Loader, driver, driver shim, and driver connection messages.
5–7	Previous level plus change log and loopback messages. Higher trace levels provide more detail.
8	Previous level plus driver status log, driver parameters, driver command line, driver security, driver Web server, driver schema, driver encryption, driver PAM, driver SOAP API, and driver include/exclude file messages.
9	Previous level plus low-level networking and operating system messages.
10	Previous level plus maximum low-level program details (all options).

The following is an example `/etc/nxdrv.conf` line to set the trace level:

```
-trace 9
```

To view the trace file:

1. Use a Web browser to access the driver shim at `https://`*`driver-address`*`:8091`. Substitute the DNS name or IP address of your driver for *driver-address*.
2. Authenticate by using any user name and the password that you specified as the Remote Loader password.
3. Click **Trace**.

The Script Output File

By default, script output is written to `/usr/local/nxdrv/logs/script-trace.log` on the connected system. This file captures the standard error output from all scripts executed by the driver shim. The location of the script output file is set in the `globals.sh` script.

DSTRACE

You can view Identity Manager information using the DSTRACE facility on the Metadirectory server. Use iManager to set the tracing level. For example, trace level 2 shows Identity Vault events in XML documents, and trace level 5 shows the results of policy execution. Because a high volume of trace output is produced, we recommend that you capture the trace output to a file. For details about using DSTRACE, see the *NetIQ® Identity Manager 4.8 Administration Guide* on the Identity Manager 4.8 Documentation Web site (https://www.netiq.com/documentation/identity-manager-47/).

The Status Log

The status log is a condensed summary of the events that have been recorded on the Subscriber and Publisher channels. This file exists on the connected system at `/usr/local/nxdrv/logs/dirxml.log`. You can also view the status log in iManager on the Driver Overview page. You can change the log level to specify what types of events to log. For details about using the status log, see the *NetIQ Identity Manager 4.8 Administration Guide*.

To view the status log:

1. Use a Web browser to access the driver shim at `https://driver-address:8091`. Substitute the DNS name or IP address of your driver for *driver-address*.
2. Authenticate by using any user name and the password that you specified as the Remote Loader password.
3. Click **Status**.

The PAM Trace File

To log PAM trace messages to `/usr/local/nxdrv/logs/pam_nxdrv.log`, specify the `debug=*` command line option for the driver PAM module in your PAM configuration file. This file is implementation dependent. For details, see your system's PAM documentation. For details about the driver PAM module command line options, see Table C-4, "Linux and UNIX Driver PAM Module Command Line Options," on page 95.

Troubleshooting Common Problems

Driver Shim Installation Failure

- Ensure that you use the correct installation program for your operating system and that you are running on a supported operating system. For details, see Table 3-1, “Linux and UNIX Installation Script Filenames,” on page 22.

  Also, for more information about required systems and software, as well as supported platforms and operating environments, see the Identity Manager 4.8 Drivers Documentation Web site (https://www.netiq.com/documentation/identity-manager-47-drivers). From this index page, you can select a readme file associated with the platform(s) for which you need support.
- Ensure that you run the installation as `root`.
- Ensure that your package management software, such as RPM, is installed and up-to-date.

Schema Update Failure

Examine the log file at `/var/nds/schema.log`.

Ensure that you specify the correct parameters (host name, ADMIN FDN in dotted format, and password).

Ensure that you have network connectivity to the Metadirectory server.

Driver Certificate Setup Failure

To set up certificates, the driver shim communicates with the Metadirectory server using the LDAP secure port (636).

- Ensure that eDirectory™ is running LDAP with SSL enabled. For details about configuring eDirectory, see the *NetIQ eDirectory Administration Guide*.
- Ensure that the connected system has network connectivity to the Metadirectory server.

You can use the command `/usr/local/nxdrv/bin/nxdrv -s` to configure the certificate at any time.

If you cannot configure SSL using LDAP, you can install the certificate manually.

1. In iManager, browse the Security container to locate your tree’s Certificate Authority (typically named *`treeName`* `CA`).
2. Click the Certificate Authority object.
3. Click **Modify Object**.
4. Select the **Certificates** tab.
5. Click **Public Key Certificate**.
6. Click **Export**.
7. Select **No** to export the certificate without the private key, then click **Next**.
8. Select **Base64 format**, then click **Next**.
9. Click **Save the exported certificate to a file**, then specify a location to save the file.
10. Use FTP or another method to store the file on the connected system as `/usr/local/nxdrv/keys/ca.pem`.

Driver Start Failure

- Examine the status log and DSTRACE output.
- The driver must be specified as a Remote Loader driver, even if the Identity Vault and connected system are the same computer. You can set this option in the iManager Driver Edit Properties window.
- You must activate both Identity Manager and the driver within 90 days. The Driver Set Overview page in iManager shows when Identity Manager requires activation. The Driver Overview page shows when the driver requires activation.

  For details about activating NetIQ Identity Manager Products, see the *Identity Manager 4.8 Installation Guide* on the Identity Manager 4.8 Documentation Web site (https://www.netiq.com/documentation/identity-manager-47/).

For more information about troubleshooting Identity Manager engine errors, see the Identity Manager 4.8 Documentation Web site (https://www.netiq.com/documentation/identity-manager-47/).

Driver Shim Startup or Communication Failure

- Examine the trace file.
- Ensure that the connected system's operating system version is supported. For information about required systems and software, as well as supported platforms and operating environments, see the Identity Manager 4.8 D\rivers Documentation Web site (https://www.netiq.com/documentation/idm45/). From this index page, you can select a readme file associated with the platform(s) for which you need support.
- Apply all patches for your operating system.
- Ensure that the Remote Loader and Driver object passwords that you specified while setting up the driver on the Metadirectory server match the passwords stored with the driver shim.

  To update these passwords on the connected system, use the `nxdrv-config` command. The passwords are stored under `/usr/local/nxdrv/keys` in encrypted files `dpwdlf40` (Driver object password) and `lpwdlf40` (Remote Loader password).

  To update these passwords on the Metadirectory server, use iManager to update the driver configuration. For details, see "Driver Configuration Page" on page 37.
- Ensure that the correct host name and port number of the connected system are specified in the Driver Configuration Remote Loader connection parameters. You can change the port number (default 8090) in `/etc/nxdrv.conf`.

Users or Groups Are Not Provisioned to the Connected System

- Examine the status log, DSTRACE output, trace file, and script output file.
- To be provisioned, users and groups must be in the appropriate base container. You can view and change the base containers in iManager on the Global Configuration Values page of the Driver Edit Properties window. For more details, see "Global Configuration Values Page" on page 40.
- To provision identities from the Identity Vault to the connected system, the driver Data Flow property must be set to Bidirectional or Identity Vault to Application. To change this value, re-import the driver rules file over your existing driver.

- If the POSIX Management Mode is Manage from Identity Vault, ensure that the identities to be provisioned have RFC 2307 information. Manage from Identity Vault sets the **Require POSIX Attributes When Subscribing** GCV.
- The user that the driver is security equivalent to must have rights to read information from the base container. For details about the rights required, see Table 2-2, “Base Container Rights Required by the Driver Security-Equivalent User,” on page 19.

Users or Groups Are Not Provisioned to the Identity Vault

- Examine the status log, DSTRACE output, and trace file.
- Examine the **User Base Container** and **Group Base Container** GCV values. For more details, see “Global Configuration Values Page” on page 40.
- To provision identities from the connected system to the Identity Vault, the driver Data Flow property must be set to Bidirectional or Application to Identity Vault. To change this value, re-import the driver rules file over your existing driver.
- The user that the driver is security equivalent to must have rights to update the base container. For details about the rights required, see Table 2-2, “Base Container Rights Required by the Driver Security-Equivalent User,” on page 19.

Identity Vault User Passwords Are Not Provisioned to the Connected System

- Examine the status log, DSTRACE output, and script output file.
- There are several password management properties available in iManager on the Global Configuration Values page of the Driver Edit Properties window. Ensure that the connected system accepts passwords from the Identity Vault. To determine the right settings for your environment, view the help for the options, or see the *NetIQ Identity Manager 4.8 Administration Guide* on the Identity Manager 4.8 Documentation Web site (https://www.netiq.com/documentation/identity-manager-47/).
- Ensure that the user’s container has an assigned Universal Password policy and that the **Synchronize Distribution Password When Setting Universal Password** option is set for this policy.

Connected System User Passwords Are Not Provisioned to the Identity Vault

- Examine the status log, DSTRACE output, and the trace file.
- There are several password management properties available in iManager on the Global Configuration Values page of the Driver Edit Properties window. Ensure that at least one of the following options is set:
  - **The Identity Vault Accepts Passwords from the Linux or UNIX Connected System**
  - **The Identity Vault Accepts Administrative Password Resets from the Linux or UNIX Connected System**

  To determine the right settings for your environment, view the help information for the options, or see the *NetIQ Identity Manager 4.8 Administration Guide* on the Identity Manager 4.8 Documentation Web site (https://www.netiq.com/documentation/identity-manager-47/).
- To set a password, use `passwd`, not `yppasswd` or `passwd -r`, because they bypass the authentication module.

- Do not specify a password with `useradd`. This bypasses the authentication module.
- If the **Require Password Policy Validation before Publishing Passwords** GCV is set, the user's password must satisfy the password rules in the password policy assigned to the user container.
- To capture passwords, PAM or LAM and the driver PAM or LAM module must be installed and enabled. For details about installing the driver PAM or LAM module, see "Installing the PAM or LAM Module" on page 30.

  You can use the `nxdrv-config` command on the connected system to configure the PAM or LAM module. For details, see "Using the nxdrv-config Command" on page 89.
- Ensure that remote NIS or NIS+ clients have the driver PAM module installed, that they have a source of entropy, and that they have network connectivity to the driver shim system.
- If you are using Red Hat AS 2.1 or 3.0, ensure that you are using the `pam_pwdb.so` PAM module. For details, see "Installing the PAM or LAM Module" on page 30.

Users or Groups Are Not Modified, Deleted, Renamed, or Moved

- Examine the status log, DSTRACE output, trace file, and script output file.
- Examine the driver Data Flow setting to verify the authoritative source for identities.
- Identity Vault and connected system identities must be associated before events are synchronized. To view an identity's associations, use Modify User/Group in iManager and click the **Identity Manager** tab. You can migrate identities to establish associations. For details, see "Migrating Identities" on page 44.
- Identity Vault move events can remove the identity from the base container monitored by the driver to a container that is not monitored by the driver. This makes the move appear to be a delete.
- Renaming a user or group is not supported by AIX.

Shared Memory Errors

Shared memory is used by the driver shim to safely and securely communicate with the scripts. If the system shared memory segments become unusable, you must shut down the process and fix the shared memory segments.

Shared memory segments can become unusable on some UNIX systems if the driver shim is improperly terminated without detaching from the segments. For information about how to properly stop the driver shim, see "Starting and Stopping the Driver Shim" on page 57. You can use the `ipcs` system tool to locate these segments and the `ipcrm` tool to manually clear them as shown in the following example:

```
> ipcs -m

------ Shared Memory Segments --------
key          shmid     owner      perms      bytes      nattch     status
0x2a065bbd 1802241  root      600      16384      1

> ipcrm -m 1802241
```

The driver shim generates default segments of 16384 bytes and permissions 600.

B System and Error Messages

Components of the Identity Manager 4.8 driver for Linux and UNIX write messages to the system log to report operational status and problems. For more information about the system log, see “The System Log” on page 63. For detailed troubleshooting information, see Appendix A, “Troubleshooting,” on page 63.

Each message begins with a code of 3-6 characters associated with the driver component that generated the message. Use this code to find message information quickly as follows:

CFG Messages

Messages beginning with CFG are issued by configuration file processing.

CFG001E Could not open configuration file *filename*.

Explanation: Could not open the configuration file.

Possible cause: The file does not exist.

Possible cause: You don’t have permission to read the file.

Action: Ensure that the configuration file exists at the correct location and that you have file system rights to read it.

CFG002E Error parsing configuration file line: <*configline*>.

Explanation: The line is not formatted as a valid configuration statement and cannot be parsed.

Action: Correct the line in the configuration file.

CFG003W Configuration file line was ignored. No matching statement name found: <*configline*>.

Explanation: This line is formatted as a valid configuration file statement, but the statement is not recognized. The line is ignored.

Possible cause: The statement is incorrectly typed or the statement name is used only in a newer version of the software.

Action: Correct the statement.

CFG004E Error parsing configuration file line. No statement name was found: <*configLine*>.

Explanation: Could not find a statement name on the configuration line.

Action: Correct the line in the configuration file to supply the required statement.

CFG005E A required statement *statement_id* is missing from the configuration file.

Explanation: The *statement_id* statement was not specified in the configuration file, but is required for the application to start.

Action: Add the required statement to the configuration file.

CHGLOG Messages

Messages beginning with CHGLOG are issued by change log processing.

CHGLOG000I *nameversion* Copyright 2005 Omnibond Systems, LLC. ID=*code_id_string*.

Explanation: This message identifies the system component version.

Action: No action is required.

DOM Messages

Messages beginning with DOM are issued by driver components as they communicate among themselves.

DOM0001W XML parser error encountered: *errorString*.

Explanation: An error was detected while parsing an XML document.

Possible cause: The XML document was incomplete, or it was not a properly constructed XML document.

Action: See the error string for additional details about the error. Some errors, such as no element found, can occur during normal operation and indicate that an empty XML document was received.

DRVCOM Messages

Messages beginning with DRVCOM are issued by the include/exclude system.

DRVCOM000I *nameversion* Copyright 2005 Omnibond Systems, LLC. ID=*code_id_string*.

Explanation: This message identifies the system component version.

Action: No action is required.

DRVCOM001W Invalid include/exclude CLASS statement.

Explanation: The include/exclude configuration file contains an invalid CLASS statement.

Action: Correct the include/exclude configuration file with proper syntax.

DRVCOM002D An include/exclude Rule was added for class: *class*.

Explanation: The include/exclude configuration supplied a rule for the specified class.

Action: None.

DRVCOM003D An include/exclude Association Rule was added for association *association*.

Explanation: The include/exclude configuration supplied an association rule for the specified association.

Action: None.

HES Messages

Messages beginning with HES are issued by driver components as they use HTTP to communicate.

HES001E Unable to initialize the HTTP client.

Explanation: Communications in the client could not be initialized.

Possible cause: Memory is exhausted.

Action: Increase the amount of memory available to the process.

HES002I Connecting to host *host_name* on port *port_number*.

Explanation: The client is connecting to the specified server.

Action: None.

HES003W SSL communications have an incorrect certificate. rc = *rc*.

Explanation: The security certificate for SSL services could not be verified.

Possible cause: The certificate files might be missing or invalid.

Action: Obtain a new certificate.

LWS Messages

Messages beginning with LWS are issued by the integrated HTTP server.

LWS0001I Server has been initialized.

Explanation: The server has successfully completed its initialization phase.

Action: None. Informational only.

LWS0002I All services are now active.

Explanation: All of the services offered by the server are now active and ready for work.

Action: None. Informational only.

LWS0003I Server shut down successfully.

Explanation: The server processing completed normally. The server ends with a return code of `0`.

Action: No action is required.

LWS0004W Server shut down with warnings.

Explanation: The server processing completed normally with at least one warning. The server ends with a return code of `4`.

Action: See the log for additional messages that describe the warning conditions.

LWS0005E Server shut down with errors.

Explanation: The server processing ended with one or more errors. The server ends with a return code of `8`.

Action: See the log for additional messages that describe the error conditions.

LWS0006I Starting *service*.

Explanation: The server is starting the specified service.

Action: None. Informational only.

LWS0007E Failed to start *service*.

Explanation: The server attempted to start the specified service, but the service could not start. The server terminates processing.

Action: See the log for additional messages that describe the error condition.

LWS0008I Stopping all services.

Explanation: The server was requested to stop. All services are notified and will subsequently end processing.

Action: None. Informational only.

LWS0009I Local host is *host_name* (*IP_address*).

Explanation: This message shows the host name and IP address of the machine that the server is running on.

Action: None. Informational only.

LWS0010I Local host is *IP_address*.

Explanation: This message shows the IP address of the machine that the server is running on.

Action: None. Informational only.

LWS0011I Server is now processing client requests.

Explanation: The server has successfully started all configured services, and it is ready for clients to begin requests.

Action: None. Informational only.

LWS0012I *service* is now active on port *number*.

Explanation: The server *service* is running on the specified TCP port *number*. Clients can begin making requests to the specified service.

Action: None. Informational only.

LWS0013I *service* is now inactive on port *number*.

Explanation: The server *service* is not active on the specified TCP port *number*. Processing continues, but no client requests can be made to the service until it becomes active again.

Action: None. Informational only.

LWS0014E An error was encountered while parsing execution parameters.

Explanation: An error occurred while parsing the execution parameters. The server terminates with a minimum return code of 8.

Action: Collect diagnostic information and contact NetIQ® Technical Support.

LWS0015E *service* failed to start with error *number*.

Explanation: The specified service failed to start. The server terminates with a minimum return code of 8.

Action: Collect diagnostic information and contact NetIQ Technical Support.

LWS0020I Server *version* level: *level*.

Explanation: This message contains information detailing the current service level for the server program being executed. The value of *version* indicates the current release of the server. The value of *level* is a unique sequence of characters that can be used by NetIQ Technical Support to determine the maintenance level of the server being executed.

Action: Normally, no action is required. However, if you report a problem with the server to NetIQ Technical Support, you might be asked to provide the information in the message.

LWS0023I Listen port *number* is already in use.

Explanation: The displayed listen port is already in use by another task running on the local host. The server retries establishing the listen port.

Action: Determine what task is using the required port number and restart the server when the task is finished, or specify a different port in the configuration file. If the port number is changed for the server, the client must also specify the new port number.

LWS0024W Too many retries to obtain port *number*.

Explanation: The server tried multiple attempts to establish a listen socket on the specified port number, but the port was in use. The server terminates with a return code of 4.

Action: Determine what task is using the required port number, and restart the server when the task is finished, or specify a different port in the configuration file. If the port number is changed for the server, the client must also specify the new port number.

LWS0025I Local TCP/IP stack is down.

Explanation: The server detected that the local host TCP/IP service is not active or is unavailable. The server retries every two minutes to reestablish communication with the TCP/IP service.

Action: Ensure that the TCP/IP service is running.

LWS0026E Unrecoverable TCP/IP error *number* returned from *internal_function_name*.

Explanation: An unrecoverable TCP/IP error was detected in the specified internal server function name. The server ends with a minimum return code of 8. The error number reported corresponds to a TCP/IP errno value.

Action: Correct the error based on TCP/IP documentation for the specified errno.

LWS0027W Listen socket was dropped for port *number*.

Explanation: The server connection to the displayed listen port was dropped. The server attempts to reconnect to the listen port so that it can receive new client connections.

Action: Determine why connections are being lost on the local host. Ensure that the host TCP/IP services are running.

LWS0028E Unable to reestablish listen socket on port *number*.

Explanation: The listen socket on the specified port number was dropped. The server tried multiple attempts to reestablish the listen socket, but all attempts failed. The server ends with a return code of 8.

Action: Determine if the host's TCP/IP service is running. If the host's TCP/IP service is running, determine if another task on the local host is using the specified port.

LWS0029I <*id*> Client request started from *ip_address* on port *number*.

Explanation: A new client request identified by *id* has been started from the specified IP address on the displayed port number.

Action: None. Informational only.

LWS0030I <*id*> Client request started from *host* (*ip_address*) on port *number*.

Explanation: A new client request identified by *id* has been started from the specified host and IP address on the displayed port number.

Action: None. Informational only.

LWS0031W Unable to stop task *id*: *reason*.

Explanation: The server attempted to terminate a service task identified by *id*. The server could not stop the task for the specified reason. The server ends with a return code of 4.

Action: See the reason text for more information about why the task could not terminate.

LWS0032I <*id*> Client request has ended.

Explanation: The client requested identified by *id* has ended.

Action: None. Informational only.

LWS0033I <*id*> Client request: *resource*.

Explanation: The client connection identified by *id* issued a request for *resource*.

Action: None. Informational only.

LWS0034W <*id*> Write operation for client data has failed.

Explanation: A write operation failed for the connection identified by *id*. This is normally because the client dropped the connection. The client connection is dropped by the server.

Action: Ensure that the client does not prematurely drop the connection. Retry the client request if necessary.

LWS0035W <*id*> Read operation for client data has timed out.

Explanation: A read operation on the connection identified by *id* has timed out because of inactivity. The client connection is dropped by the server.

Action: Ensure that the client does not prematurely drop the connection. Retry the client request if necessary.

LWS0036W <*id*> Client request error: *error_code* - *error_text*.

Explanation: The server encountered an error while processing the client request. The server terminates the request.

Action: Determine why the request was in error by viewing the error code and error text that was generated.

LWS0037W <*id*> Client request error: *code*.

Explanation: The server encountered an error while processing the client request. The server terminates the request.

Action: Determine why the request was in error by viewing the error code and error text that was generated.

LWS0038I Received command: *command_text*.

Explanation: The server has received the displayed command from the operator. The server processes the command.

Action: None. Informational only.

LWS0043E Task *id* ended abnormally with RC=*retcode*.

Explanation: The server detected a task that ended with a non-zero return code. The server ends with a minimum return code of 8.

Action: View the log for other messages that might have been generated regarding the error.

LWS0045I Idle session time-out is *number* seconds.

Explanation: The message shows the idle time limit for connections. The server automatically terminates sessions that are idle for longer than the specified number of seconds.

Action: None. Informational only.

LWS0046I Maximum concurrent sessions limited to *number*.

Explanation: The message shows the maximum number of concurrent sessions allowed. The server allows only the specified number of concurrent sessions to be active at any given time. All connections that exceed this limit are forced to wait until the total number of connections drops below the specified value.

Action: None. Informational only.

LWS0047W Unable to delete log file *filename*.

Explanation: The log file could not be deleted as specified.

Possible cause: The user service or daemon does not have file system rights to delete old log files.

Action: Verify that the user service or daemon has the appropriate rights.

Action: Examine the current logs for related messages.

LWS0048I Log file *filename* successfully deleted.

Explanation: The log file has been deleted as specified.

Action: None. Informational only.

LWS0049E Error *error* authenticating to the directory as *fdn*.

Explanation: The connection manager could not connect to the directory as user *fdn*. The error was *error*.

Possible cause: The configuration parameters do not contain the correct user or password.

Action: Correct the cause of the error as determined from *error*.

Action: Verify that the User object has the appropriate rights.

Action: Verify that the password given for the User object in the configuration parameters is correct.

LWS0050E Server application initialization failure was detected.

Explanation: During server initialization, an error was detected while initializing the server Application object.

Possible Cause This message is commonly logged when the driver is started and then immediately shut down. This can happen during installation, when the shim is started to generate keys or configure SSL. You can safely ignore this message in those cases.

Action: See the error logs for additional messages that indicate the cause of the error.

LWS0051E Server initialization failure was detected.

Explanation: The server failed to initialize properly because of an initialization error specific to the operating system.

Action: See the log for additional messages that indicate the cause of the error.

LWS0052W This server is terminating because of another instance already running (*details*).

Explanation: The server is shutting down because there is another active instance of this server running on the host.

Possible cause: A previous instance of the server was not stopped before starting a new instance.

Action: Stop or cancel the previous server instance before starting a new one.

LWS0053I The parameter *keyword* is no longer supported.

Explanation: The specified parameter is no longer supported in this release and might be removed in future releases.

Possible cause: An execution parameter was specified that is no longer supported.

Action: Do not specify the unsupported parameter.

LWS0054I The execution parameter *keyword* is in effect.

Explanation: The specified execution parameter is in effect for the server.

Action: Informational only. Processing continues.

LWS0055W Invalid execution parameter detected: *keyword*.

Explanation: An invalid execution parameter was detected.

Action: Do not specify the invalid or unknown execution parameter.

LWS0056I Not accepting new connections because of the MAXCONN limit. There are *number* active connections now for *service*.

Explanation: The specified service has a maximum connection limit that has been reached. The service no longer accepts new connections until at least one of the active connections ends.

Action: If you receive this message frequently, increase the MAXCONN limit for this service or set the MAXCONN to unlimited connections.

LWS0057I New connections are now being accepted for *service*.

Explanation: The service was previously not accepting new connections because of the imposed MAXCONN limit. The service can now accept a new connection because at least one active connection has ended.

Action: None. Informational only.

LWS0058I Listen socket on port *number* has been re-established.

Explanation: The previously dropped listen socket has been re-established. Services using the specified port can now continue. The listen socket previously dropped because of an error or TCP/IP connectivity problems has been re-established. Client connection processing continues.

Action: None. Informational only.

LWS0059W Server is terminating because the required service *serviceName* is ending.

Explanation: The specified required service has ended. The server terminates because it cannot continue running without the required service.

Action: See related log messages to determine why the required service ended. Correct the problem and restart the server.

NET Messages

Messages beginning with NET are issued by driver components during verification of SSL certificates.

NET001W Certificate verification failed. Result is *result*.

Explanation: A valid security certificate could not be obtained from the connection client. Diagnostic information is given by *result*.

Possible cause: A security certificate has not been obtained for the component.

Possible cause: The security certificate has expired.

Possible cause: The component certificate directory has been corrupted.

Action: Respond as indicated by *result*. Obtain a new certificate if appropriate.

NIX Messages

Messages beginning with NIX are issued by the driver shim.

NIX000I *nameversion* Copyright 2005 Omnibond Systems, LLC. ID=*code_id_string*.

Explanation: This message identifies the system component version.

Action: No action is required.

NIX001S An error occurred attempting to attach the shared memory segment to an address space (errno=*errno*).

Explanation: The driver uses shared memory as the mechanism for providing information to the shell scripts. An error occurred attempting to attach the shared memory to a physical address for access.

Possible cause: The calling process has no access permissions for the requested attach type.

Possible cause: An invalid or non-page-aligned address was provided to the system routine.

Possible cause: Memory could not be allocated for the descriptor or for the page tables.

Action: Restart the driver process and ensure that there are adequate memory resources. Verify that the driver process is run as `root` and has permissions to read its configuration files. Contact NetIQ Technical Support for additional instructions if necessary.

NIX002S An error occurred while attempting to allocate a shared memory segment (errno = *errno*).

Explanation: The driver uses shared memory as the mechanism for providing information to the shell scripts. An error occurred attempting to allocate a shared memory segment.

Possible cause: The memory size was too small or too large.

Possible cause: The system shared memory settings might not have adequate values.

Possible cause: The memory segment could not be created because it already exists. This could be caused by an abnormal termination of a previous driver process.

Possible cause: All possible shared memory IDs have been taken.

Possible cause: Allocating a segment of the requested size would cause the system to exceed the system-wide limit on shared memory.

Possible cause: No shared memory segment exists for the given key.

Possible cause: The user or process does not have permission to access the shared memory segment.

Possible cause: No memory could be allocated for segment overhead.

Action: Restart the driver process and ensure that there is sufficient memory.

Action: Verify that the driver process is run as `root` and has permissions to read its configuration files.

Action: If there are other applications on the server that use shared memory, ensure that they are running, healthy, and do not conflict with the requirements for the driver.

Action: Contact NetIQ Technical Support for additional instructions if necessary.

NIX003S An error occurred attempting to create a System V IPC key. The project identifier pathname = *pathname*.

Explanation: The driver uses shared memory as the mechanism for providing information to the shell scripts. An error occurred attempting to create the key used to specify the shared memory segment.

Possible cause: The project pathname is invalid or does not exist.

Action: Restart the driver process.

Action: Ensure that the file pathname is correct and that the process has adequate permissions to read the path.

NIX004S An error occurred while writing data to shared memory (bytes = *bytes*, allocationSize = *allocationSize*).

Explanation: The driver uses shared memory as the mechanism for providing information to the shell scripts. An error occurred while writing data from the driver process into the shared memory segment.

Possible cause: Invalid memory resources or internal error.

Action: Contact NetIQ Technical Support.

NIX005S An error occurred attempting to set an environment variable.

Explanation: The driver uses environment variables for some of the communication between the driver and other processes called from the scripts. An error occurred setting an environment variable.

Possible cause: There was not enough space to allocate the new environment.

Action: Restart the driver and ensure that there are adequate memory resources for the driver process.

NIX006S An error occurred attempting to execute the script [*script*].

Explanation: The driver uses shell scripts to update the system for events from the Identity Vault. An error occurred while attempting to execute one of these scripts.

Possible cause: The script does not exist on the local system.

Possible cause: A memory or environment allocation failure occurred.

Action: Restart the driver and ensure that the script exists on the local system.

NIX007S An error occurred attempting to terminate the script [*script*].

Explanation: The driver uses shell scripts to update the system for events from the Identity Vault. An error occurred while attempting to terminate the script.

Possible cause: The script does not exist on the local system.

Possible cause: A memory or environment allocation failure occurred.

Action: Restart the driver and ensure that the script exists on the local system.

NIX008S The shared memory tool was unable to retrieve a key from the environment.

Explanation: The shared memory tool uses an environment variable to retrieve the key used to unlock the shared memory region and access driver shim data. The tool could not obtain the key from the environment.

Possible cause: The driver shim cannot set environment variables, or the environment has become corrupt during event processing.

Action: Restart the driver shim process and clear any residual shared memory segments.

NXLAM Messages

Messages beginning with NXLAM are issued by the driver LAM module.

NXLAM000I *nameversion* Copyright 2006 Omnibond Systems, LLC. ID=*code_id_string*.

Explanation: This message identifies the system component version.

Action: No action is required.

NXLAM001W Password Change was not submitted for *user*.

Explanation: When a user changes the password using a LAM-enabled application, the LAM module for the driver submits the password change to the change log. An error occurred that prevents the change being submitted to the change log.

Possible cause: If the LAM module is running locally on the same system with the driver shim, certain files or directories could be missing, such as the `/usr/local/nxdrv/keys/lpwd1f40` driver shim key file or the `/usr/local/nxdrv/changelog` change log directory.

Possible cause: If the LAM module is running remotely from the system with the driver shim, the LAM module could not connect to the driver shim. This could be caused by a network problem or a problem with the driver shim.

Possible cause: The LAM module might not be configured properly.

Action: Ensure that the LAM module is installed and configured correctly.

Action: Ensure that the driver shim is running and healthy.

Action: If the LAM module is running remotely, verify connectivity to the driver shim system.

NXPAM Messages

Messages beginning with NXPAM are issued by the driver PAM module.

NXPAM000I *nameversion* Copyright 2006 Omnibond Systems, LLC. ID=*code_id_string*.

Explanation: This message identifies the system component version.

Action: No action is required.

NXPAM001W Password Change was not submitted for *user*.

Explanation: When a user changes the password using a PAM-enabled application, the PAM module for the driver submits the password change to the change log. An error occurred that prevents the change being submitted to the change log.

Possible cause: If the PAM module is running locally on the same system with the driver shim, certain files or directories could be missing, such as the `/usr/local/nxdrv/keys/lpwd1f40` driver shim key file or the `/usr/local/nxdrv/changelog` change log directory.

Possible cause: If the PAM module is running remotely from the system with the driver shim, the PAM module could not connect to the driver shim. This could be caused by a network problem or a problem with the driver shim.

Possible cause: The PAM module might not be configured properly.

Action: Ensure that the PAM module is installed and configured correctly.

Action: Ensure that the driver shim is running and healthy.

Action: If the PAM module is running remotely, verify connectivity to the driver shim system.

OAP Messages

Messages beginning with OAP are issued by driver components while communicating among themselves.

OAP001E Error in SSL configuration. Verify system entropy.

Explanation: Entropy could not be obtained for SSL.

Possible cause: A source of entropy is not configured for the system.

Action: Obtain and configure a source of entropy for the system.

OAP002E Error in SSL connect. Network address does not match certificate.

Explanation: The SSL client could not trust the SSL server it connected to, because the address of the server did not match the DNS name or IP address that was found in the certificate for the server.

Possible cause: The appropriate credentials are missing from the configuration.

Action: If you cannot resolve the error, collect diagnostic information and contact NetIQ Technical Support.

OAP003E Error in SSL connect. Verify address and port.

Explanation: A TCP/IP connection could not be made.

Possible cause: The server is not running.

Possible cause: The configuration information does not specify the correct network address or port number.

Action: Verify that the server is running properly.

Action: Correct the configuration.

OAP004E HTTP Error: *cause*.

Explanation: The user name or password provided failed basic authentication.

Possible cause: The user name or password is incorrect.

Action: Verify that user name is in full context (cn=user,ou=ctx,o=org or user.ctx.org) and that the password was correctly typed.

OAP005E HTTP Error: Internal Server Error.

Explanation: The server experienced an internal error that prevents the request from being processed.

Possible cause: A secure LDAP server is not available.

Action: Ensure that the LDAP server is available.

Action: Ensure that the LDAP host and port are configured correctly.

RDXML Messages

Messages beginning with RDXML are issued by the embedded Remote Loader.

RDXML000I *nameversion* Copyright 2005 Omnibond Systems, LLC. ID=*code_id_string*.

Explanation: This message identifies the system component version.

Action: No action is required.

RDXML001I Client connection established.

Explanation: A client has connected to the driver. This can be the Metadirectory engine connecting to process events to and from the driver, or a Web-based request to view information or publish changes through the SOAP mechanism.

Action: No action required.

RDXML002I Request issued to start Driver Shim.

Explanation: The driver received a command to start the driver shim and begin processing events.

Action: No action required.

RDXML003E An unrecognized command was issued. The driver shim is shutting down.

Explanation: The driver received an unrecognized command from the Metadirectory engine. The driver shim is shutting down to avoid further errors.

Possible cause: Network error.

Possible cause: Invalid data sent to the driver.

Possible cause: The Metadirectory engine version might have been updated with new commands that are unrecognized by this version of the driver.

Possible cause: This message is logged when the driver shim process is shut down from the connected system rather than from a Driver object request. The local system can queue an invalid command to the driver shim to simulate a shutdown request and terminate the running process.

Action: Ensure that the network connection is secured and working properly.

Action: Apply updates for the engine or driver if necessary.

Action: If the driver shim process was shut down from the local system, no action is required.

RDXML004I Client Disconnected.

Explanation: A client has disconnected from the driver. This might be the Metadirectory engine disconnecting after a driver shutdown request or a Web-based request that has ended.

Action: No action required.

RDXML005W Unable to establish client connection.

Explanation: A client attempted to connect to the driver, but was disconnected prematurely.

Possible cause: The client is not running in SSL mode.

Possible cause: Mismatched SSL versions or mismatched certificate authorities.

Possible cause: Problems initializing SSL libraries because of improperly configured system entropy settings.

Action: Ensure that both the Metadirectory engine and the driver are running in the same mode: either clear text mode or SSL mode.

Action: If you are using SSL, ensure that the driver and Metadirectory engine have properly configured certificates, and that the driver system is configured properly for entropy.

RDXML006E Error in Remote Loader Handshake.

Explanation: The Metadirectory engine attempted to connect to the driver, but the authorization process failed. Authorization requires that both supply mutually acceptable passwords. Passwords are configured at installation.

Possible cause: The Remote Loader or Driver object passwords do not match.

Action: Set the Remote Loader and Driver object passwords to the same value for both the driver and the driver shim. Use iManager to modify the driver properties. Re-configure the driver shim on the connected system.

RDXML007I Driver Shim has successfully started and is ready to process events.

Explanation: The Metadirectory engine has requested the driver to start the shim for event processing, and the driver shim has successfully started.

Action: No action required.

RDXML008W Unable to establish client connection from *remoteName*.

Explanation: A client attempted to connect to the driver, but was disconnected prematurely.

Possible cause: The client is not running in SSL mode.

Possible cause: Mismatched SSL versions or mismatched certificate authorities.

Possible cause: Problems initializing SSL libraries because of improperly configured system entropy settings.

Action: Ensure that both the Metadirectory engine and the driver are running in the same mode: either clear text mode or SSL mode.

Action: If you are using SSL, ensure that the driver and Metadirectory engine have properly configured certificates, and that the driver system is configured properly for entropy.

RDXML009I Client connection established from *remoteName*.

Explanation: A client has connected to the driver. This can be the Metadirectory engine connecting to process events to and from the driver, or a Web-based request to view information or publish changes through the SOAP mechanism.

Action: No action required.

C Technical Details

Topics in this section include

Using the nxdrv-config Command

You can use `/usr/sbin/nxdrv-config` to change the driver shim configuration. When you run this command, you are prompted for the function to perform.

```
> nxdrv-config
Which configuration do you want to perform?
1) Set the Remote Loader and Driver object passwords
2) Configure the driver for Secure Sockets Layer (SSL)
3) Configure the driver to allow for remote client publishing,
   such as NIS or NIS+ clients
4) Extend the schema for Identity Manager (must be run on a
   Metadirectory server)
5) Configure PAM for publishing password changes
6) Configure LAM for publishing password changes
Select one configuration option [q/?]:
```

Enter the number of the function you want to configure, then respond to the prompts as discussed in the following toopic:

Setting the Remote Loader and Driver Object Passwords

The `nxdrv-config` command prompts you to enter and confirm the Remote Loader password and the Driver object password.

```
Enter Remote Loader password:
Confirm Remote Loader password:
Enter Driver object password:
Confirm Driver object password:
```

The Remote Loader password is used by the Metadirectory engine to authenticate itself to the driver shim (embedded Remote Loader). The Driver object password is used by the driver shim to authenticate itself to the Metadirectory engine.

The Remote Loader and Driver object passwords set by `nxdrv-config` are stored on the connected system. The Remote Loader and Driver object passwords set for the driver using iManager are stored in the Identity Vault. Each password on the connected system must exactly match its counterpart in the Identity vault.

To change the passwords after driver installation:

1. In iManager, navigate to the Driver Overview for the driver.
2. Click the driver icon.
3. Specify the Driver object password.
4. Specify the Remote Loader password.

   The Remote Loader password is below the Authentication heading.
5. Click **Apply**.
6. Restart the driver.

Configuring the Driver for SSL

The `nxdrv-config` command prompts you to enter the LDAP server host address and port, then displays the Certificate Authority for that server and asks you if you accept it.

```
You are about to connect to the eDirectory LDAP server to retrieve
the eDirectory Tree Trusted Root public certificate.

Enter the LDAP Server Host Address [localhost]: sr.digitalairlines.com
Enter the LDAP Server Port [636]:

Certificate Authority:
   Subject:       ou=Organizational CA,o=TREENAME
   Not Before:    20050321144845Z
   Not After:     20150321144845Z
Do you accept the Certificate Authority? (Y/N) y
```

Enter the host name or IP address and TCP port number of an LDAP server for your Identity Vault. The LDAP server must be configured for SSL, and it must be listening on the SSL port. The default SSL port is 636.

The driver shim connects to the specified server and displays information about the Certificate Authority. If you accept the Certificate Authority, the driver shim saves it to the local file system.

If you do not have LDAP configured for SSL, you can use a manual process to configure the driver for SSL. For details, see “Driver Certificate Setup Failure” on page 66.

Configuring Remote Client Publishing

The `nxdrv-config` command generates a new certificate and key, used to authenticate remote publishing clients, such as NIS and NIS+ clients.

```
New certificate authority keys were generated:

  Subject:       /CN=soap api certificate authority
  Serial Number: 0
  Valid From:    20060411002823Z
  Valid To:      20160409002823Z
```

The keys are 2048-bit, Base64-encoded, RSA public/private key pairs. They are written to `/usr/local/nxdrv/keys/soap-ca-cert.pem` (public certificate) and `/usr/local/nxdrv/keys/soap-ca-key.pem` (private key). These keys are used to issue and sign certificates for remote publishing when you configure PAM on a remote client. The default time duration for the certificate authority is 10 years. You can change the time duration and other remote publisher parameters in the configuration file `/usr/local/nxdrv/conf/remote-publisher.conf`. For details about the configuration file, see "The Remote Publisher Configuration File" on page 91.

Configuring PAM

The `nxdrv-config` command asks you if you are configuring PAM on a remote client.

If you are configuring PAM on a remote client, the `nxdrv-config` command does the following:

1. Prompts you for the host name or IP address and port number of the Linux or UNIX connected system.
2. Calls the command to mint a security certificate for the remote client. This command requires you to enter the Remote Loader password.
3. Sets up the PAM configuration file.

If you are configuring PAM on the connected system, the `nxdrv-config` command sets up the PAM configuration file.

```
Are you configuring PAM from a remote NIS client? (Y/N) [N]
Configuring PAM...
Using PAM configuration file: [/etc/pam.conf]
Inserting line [/usr/lib/security/pam_nxdrv.so.1 mechanism=api]
original PAM file backed up to /etc/pam.conf.nxdrv.04152006151641
```

The `nxdrv-config` command locates the PAM configuration file, makes a backup copy, and inserts a line for the Linux and UNIX driver PAM module.

Configuring LAM

The `nxdrv-config` command makes a backup copy of the `/usr/lib/security/methods.cfg` file, then appends the stanza for the Linux and UNIX driver to the `methods.cfg` file.

```
original methods.cfg backed up to
/usr/lib/security/methods.cfg.nxdrv.04152006154047
```

The Remote Publisher Configuration File

The `/usr/local/nxdrv/conf/remote-publisher.conf` file on the connected Linux or UNIX system controls the issuing of security certificates to remote publishing clients. It is used when a remote client is configured.

Enter configuration statements, one per line.

Comments

Lines that begin with an octothorpe (#) are comments.

Example

```
# This is a comment line.
```

CA-DELAY Statement

The `CA-DELAY` statement specifies the number of days that the Certificate Authority remains valid.

Syntax

`CA-DELAY=`*`days`*

Example

```
CA-DELAY=3650
```

CLIENT-DELAY Statement

The `CLIENT-DELAY` statement specifies the number of days that the client certificate remains valid.

Syntax

`CLIENT-DELAY=`*`days`*

Example

```
CLIENT-DELAY=1025
```

VERIFY-SERIAL-NUMBERS Statement

The `VERIFY-SERIAL-NUMBERS` statement specifies whether the driver shim verifies that the certificate serial number of a connecting client matches the serial number specified for it in a `CLIENT` statement.

Syntax

`VERIFY-SERIAL-NUMBERS={true|false}`

Example

```
VERIFY-SERIAL-NUMBERS=true
```

NEXT-SERIAL-NUMBER Statement

The `NEXT-SERIAL-NUMBER` statement specifies the next unused client certificate serial number.

Syntax

`NEXT-SERIAL-NUMBER=`*`number`*

Example

```
NEXT-SERIAL-NUMBER=1000
```

CLIENT Statements

`CLIENT` statements are written by the driver shim when a remote client is configured, and are used by the driver shim to verify a client when it connects to publish a password.

Syntax

`CLIENT ADDRESS=`*`address1,address2, . . .`* `SERIAL=`*`serialNumber`*

Example

```
CLIENT ADDRESS=192.168.17.41,192.168.17.42,192.168.17.46 SERIAL=1952
```

Driver Shim Command Line Options

The following options can be specified on the driver shim (`/usr/local/nxdrv/bin/nxdrv`) command line. You can also specify driver shim configuration file statements as command line options. For details about the driver shim configuration file, see “The Driver Shim Configuration File” on page 43.

Options Used to Set Up Driver Shim SSL Certificates

The following command line options are used to set up the driver shim SSL certificates:

Table C-1 *Driver Shim Command Line Options for Setting Up SSL Certificates*

Option (Short and Long Forms)	Description
`-s` `-secure`	Secures the driver by creating SSL certificates, then exits.
`-p` `-password`	Specifies the Remote Loader password.

Other Options

Table C-2 *Other Driver Shim Command Line Options*

Option (Short and Long Forms)	Description
`-c <congFile>` `-config <configFile>`	Instructs the driver shim to read options from the specified configuration file. Options are read from `/etc/nxdrv.conf` by default.
`-?` `-help`	Displays the command line options, then exits.
`-v` `-version`	Displays the driver shim version and build date, then exits.

PAM Configuration Details

The PAM module can publish password information on the system running the driver shim or from a remote system such as a NIS or NIS+ client. The only task of the driver PAM module is to obtain the password during normal password change operations that use PAM-enabled tools, such as the `passwd` command.

You can install and optionally configure the PAM module at any time using the installation program. For details, see “Installing the PAM or LAM Module” on page 30.

After it is installed, you can configure the PAM module with the `nxdrv-config` command. For details, see “Using the nxdrv-config Command” on page 89.

The installation script installs the PAM module as appropriate for the server operating system as shown in the following table:

Table C-3 *PAM Modules*

Operating System	PAM Module
AIX	`/usr/lib/security/pam_nxdrv`
HP-UX	`/usr/lib/security/libpam_nxdrv.1`
Linux	`/lib/security/pam_nxdrv.so`
Solaris	`/usr/lib/security/pam_nxdrv.so.1`

If you respond to the prompt to configure the PAM module, the installation script places an entry for the PAM module in the appropriate PAM configuration file for the password facility. The `nxdrv-config` command also does this.

You can edit your PAM configuration file manually. The PAM module requires a command line option as shown in Table C-4. For the location and syntax of your PAM configuration file, see your system’s PAM documentation. If you choose to edit your own PAM configuration files, you must place the PAM module entry below the module that obtains the new password during a password change.

Table C-4 *Linux and UNIX Driver PAM Module Command Line Options*

Option	Description
`debug=*`	Logs PAM module activity to the `/usr/local/nxdrv/logs/pam_nxdrv.log` file.
`host=`*`hostName`*	Required for SOAP. Specifies the host name or IP address of the driver shim system.
`mechanism=api`	The PAM module uses the API to send password change information to the driver shim. This method is used when the PAM module is running on the same system as the driver shim.
`mechanism=soap`	The PAM module uses Simple Object Access Protocol (SOAP) to send password change information to the driver shim. This method is used when the PAM module is running on a different system from the driver shim, such as with NIS or NIS+ clients.
`port=`*`portNumber`*	Required for SOAP. Specifies the TCP port number of the driver shim system. The default port is 8091.
`LC_ALL=`*`locale`*	Specifies the standard character set (*`locale`*) in use by the connected system. This ensures accurate text data conversion between the connected system and Identity Manager, which uses the UTF-8 character set. For example, the option `LC_ALL=iso8559-1` would enable the PAM module to convert passwords and user IDs in the iso8559-1 (Latin-1) character set to UTF-8.

The Linux and UNIX driver PAM module is contained in the pam-password part of the PAM stack below the other PAM modules on the system. When the other PAM modules participate in a dialog with a user who is changing the password, the driver PAM module uses pam_get_item to get the new password from the PAM framework.

When the Linux and UNIX driver PAM module obtains a new password on the system running the driver shim, it writes the new password to the change log so it can be published into the Identity Vault.

When the PAM module is used from a host other than the one where the driver shim is running (such as NIS or NIS+ clients), it uses a secure TCP/IP channel to communicate with the driver shim. If the password change event cannot be sent to the driver shim, a message is written to the system log.

LAM Configuration Details

PAM is supported by AIX beginning with AIX 5.3, but earlier versions use the IBM Loadable Authentication Module (LAM) technology instead of PAM. The Linux and UNIX driver LAM module implements password publishing in the LAM environment for files mode only. The LAM module is not supported for NIS or NIS+ on AIX.

You can install and optionally configure the LAM module at any time using the installation program. For details, see “Installing the PAM or LAM Module” on page 30.

After it is installed, you can configure the LAM module with the `nxdrv-config` command. For details, see “Using the nxdrv-config Command” on page 89.

The installation script installs the LAM module `NXDRV` into the `/usr/lib/security` directory of the connected AIX system. If you respond to the prompt to configure the LAM module, the installation script adds an NXDRV stanza to `/usr/lib/security/methods.cfg`. The `nxdrv-config` command also adds this stanza.

You can edit your `/usr/lib/security/methods.cfg` file manually. The following example shows the driver LAM stanza:

```
NXDRV:
   program = /usr/lib/security/NXDRV
   options = db=BUILTIN
```

If the LAM module is installed, the default AIX files-mode scripts cause AIX users to be associated with the LAM module via individual user stanzas in `/etc/security/user`. Alternatively, you can change the global stanza in `/etc/security/user` to use the LAM module by default, and change the scripts so that they don't assign NXDRV SYSTEM and registry attributes to files-mode users. More fine-tuned configurations are also possible and are referenced in the `add-user.sh` script file.

Publisher Channel Limitations

The Publisher channel generates events based on modifications that are discovered by polling. Because events are interpreted after they have occurred, some assumptions must be made. This can lead to unexpected results under certain circumstances.

For example, a user might be renamed on the local Linux or UNIX system. If the user's UID is not changed, the polling script can determine that the event is a rename, not a delete followed by an add. However, if a user is renamed and its UID is changed, the polling script must assume that this is a delete followed by an add.

You can modify the polling script to provide a more accurate approach using additional contextual clues that are specific to your particular environment. For example, you might modify the polling script behavior to additionally look at the password hash or a gecos field component to decide whether a user has been deleted or simply renamed. Preserving the user's identity might be essential to preserving the appropriate rights and resources to another connected system.

Files and Directories Modified by Installing the Driver Shim

Topics in this section include

Main Driver Shim Files

Main driver ship files include the following:

Driver Shim Directory

When you install the driver, the `/usr/local/nxdrv` directory is created and populated with driver-related files and subdirectories.

/usr/sbin Files

The following commands are added to `/usr/sbin`:

Table C-5 *Driver Commands Placed in /usr/sbin*

Command	Function
`nxdrv-uninstall`	Uninstalls the Linux and UNIX driver
`nxdrv-config`	Updates the configuration

init.d Files

Commands to start, stop, and display the status of the driver are added to the appropriate file for the connected system operating system.

Table C-6 *Commands for Starting, Stopping, and Displaying the Status of the Driver Shim*

Operating System	Command
AIX	`/etc/rc.d/init.d/nxdrvd`
HP-UX	`/sbin/init.d/nxdrvd`
Linux	`/etc/init.d/nxdrvd`
Solaris	`/etc/init.d/nxdrvd`

Man Pages

The installation process adds man pages for the driver shim, change log update command, and shared memory tool to `/usr/man`.

Driver Shim Configuration File

The installation program places a default driver shim configuration file at `/etc/nxdrv.conf`.

Driver PAM Files

The driver installation script adds the driver PAM module to the appropriate library, and adds a line to the PAM configuration file for the pam-password function. The location of these depends on the operating system used by the connected system. For details, see Table C-3, “PAM Modules,” on page 94 and your operating system’s PAM documentation.

Driver LAM Files

The installation script installs the LAM module `NXDRV` into the `/usr/lib/security` directory of the connected AIX system, and adds an NXDRV stanza to `/usr/lib/security/methods.cfg`.