

Administration Guide

October 22, 2008

Novell® Identity Manager Fan-Out Driver for Midrange

3.6

www.novell.com



Legal Notices

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

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

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

Copyright © 2004, 2007-2008 Omnibond Systems, LLC. All Rights Reserved. Licensed to Novell, Inc. Portions Copyright © 2004, 2007-2008 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

Novell, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed on the [Novell Legal Patents Web page \(http://www.novell.com/company/legal/patents/\)](http://www.novell.com/company/legal/patents/) and one or more additional patents or pending patent applications in the U.S. and in other countries.

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

Online Documentation: To access the online documentation for this and other Novell products, and to get updates, see [the Novell Documentation Web page \(http://www.novell.com/documentation\)](http://www.novell.com/documentation).

Novell Trademarks

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

Third-Party Materials

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

Contents

About This Guide	11
Part I Concepts and Facilities	13
1 Introduction	15
1.1 Driver Highlights	15
1.2 Driver Organization	16
2 Structure and Function	17
2.1 Core Driver	18
2.1.1 Core Driver Component Details	19
2.2 Platform Services	21
2.2.1 User and Group Management	22
2.2.2 User Authentication	23
2.2.3 Platform Configuration File	25
2.3 Directory Objects	25
2.3.1 The ASAM Master User Object	26
2.3.2 Configuration-Oriented Objects	26
2.3.3 Census Container	26
2.3.4 Platform Objects	28
2.3.5 Platform Set Objects	28
2.4 Migration	29
3 Examples	31
3.1 Password Check for Login	32
3.2 User Added to eDirectory	32
3.3 Census Trawl	33
3.4 User Deleted from eDirectory	34
3.5 Group Deleted from eDirectory	35
3.6 User Added to a Group	35
Part II Core Driver Administration	37
4 Core Driver Planning	39
4.1 Configuration Planning	39
4.2 Configuration and Performance Guidelines	41
4.2.1 eDirectory	42
4.2.2 Object Services and the Event Subsystem	42
4.2.3 Event Journal Services	43
4.2.4 Authentication Services	43
4.2.5 Platform Systems	43
4.2.6 Platform Services / Authentication Services Relationship	44
4.3 Requirements	44
4.3.1 User Rights Requirements	44
4.3.2 Password Replication Requirements	44

4.3.3	Core Driver Requirements	45
4.3.4	Requirements for Workstations Used for Installation and Administration	46
4.3.5	Platform Services Requirements	46
4.4	Installation Planning Worksheet	46
4.4.1	Installation Planning Worksheet	46
4.4.2	Items on the Installation Planning Worksheet	48
5	Installing the Core Driver	51
5.1	Preparing for Core Driver Installation	51
5.1.1	Essentials	51
5.1.2	Other Advance Considerations	51
5.1.3	General Installation Sequence	54
5.2	Step-By-Step Installation Instructions	55
5.2.1	Installing the Driver Shim on Linux or Solaris	56
5.2.2	Installing the Driver Shim on Windows Systems	60
5.2.3	Setting Up the Core Driver in iManager	64
5.2.4	Other Tasks Following Installation	67
5.3	Activating the Driver After Evaluation	69
5.4	Performance Tuning	69
5.4.1	Secondary Drivers	70
5.4.2	Platform Operation Modes	70
6	Configuring and Administering the Core Driver	71
6.1	Configuration Overview	71
6.1.1	Core Driver Configuration	71
6.1.2	Platform Services Configuration	71
6.2	Driver System Security Overview	71
6.2.1	Connection Security	72
6.2.2	ASAM Master User Security	72
6.3	Administration Overview	73
6.3.1	Monitoring Core Drivers	74
6.3.2	Monitoring Platform Services	74
6.3.3	Maintaining the Census	74
6.4	Applications For Configuration	75
6.4.1	Using iManager With the Fan-Out Driver Plug-In	76
6.4.2	Using Designer With the Fan-Out Driver Plug-In	77
6.5	Management Tasks	78
6.5.1	Configuring the Census	79
6.5.2	Configuring Core Drivers	82
6.5.3	Configuring the iManager Plug-In	87
6.5.4	Configuring Logs	87
6.5.5	Configuring Platform Sets	87
6.5.6	Configuring Platforms	89
6.5.7	Configuring Provisioning	91
6.5.8	Configuring Search Objects	92
6.5.9	Configuring Linux/UNIX UID/GID Sets	93
6.5.10	Displaying Component Status	94
6.5.11	Viewing Driver Documentation	94
6.5.12	Viewing Logs	94
6.5.13	Displaying Provisioning Details	96
6.5.14	Reviewing Naming Exceptions	96
6.5.15	Reviewing Platform Errors	97
6.5.16	Managing Trawls	97

7	Troubleshooting the Core Driver	99
7.1	Obtaining Debugging Output	99
7.1.1	Debugging the Core Driver	99
7.2	Troubleshooting Core Driver Configuration Issues	99
7.2.1	Rights Issues	100
7.2.2	Platform Services Process / Authentication Services Issues	100
7.2.3	Platform Receiver / Event Journal Services Issues	100
7.2.4	Census Issues	100
7.2.5	LDAP Issues	101
7.3	Troubleshooting Network Issues	101
7.4	Troubleshooting eDirectory Issues	101
Part III	Platform Services Planning	103
8	Platform Services Overview	105
8.1	Platform Services Component Summary	105
8.2	Authentication Services	106
8.3	Identity Provisioning	107
8.4	Account Redirection	107
8.5	The Platform Services Process	108
8.6	The Platform Services Cache Daemon	108
8.7	The System Intercept	108
8.8	The Platform Receiver	109
8.8.1	Modes of Operation	109
8.8.2	Selecting a Mode of Operation	111
8.9	Receiver Scripts	111
8.10	Standard Exclude List	112
9	Planning for Platform Services	115
9.1	Basic Considerations	115
9.2	Security Planning Considerations	115
9.2.1	Users, Passwords, and Groups	115
9.2.2	Connection Security	116
9.2.3	Administrative Password Resets	116
9.2.4	Securing the AS Client API	116
9.3	Planning Considerations for Authentication Services	116
9.4	Planning Considerations for Identity Provisioning	117
9.5	Planning Considerations for Password Replication Platforms	118
9.6	Planning Considerations for Account Redirection Platforms	118
10	The Platform Configuration File	119
10.1	Platform Configuration File Location	119
10.1.1	z/OS	119
10.1.2	i5/OS	119
10.1.3	Linux and UNIX	120
10.2	Platform Configuration File Syntax	120
10.3	Configuration Statements	120
10.3.1	ACF2.DISABLE Statement	121
10.3.2	ACF2.EXPIREWARN Statement	121
10.3.3	ADMINPASSWORD Statement	122

10.3.4	ADMINUSER Statement	122
10.3.5	AM.GROUP.INCLUDE Statement / AM.GROUP.EXCLUDE Statement	122
10.3.6	AM.USER.INCLUDE Statement / AM.USER.EXCLUDE Statement	123
10.3.7	AS.USER.INCLUDE Statement / AS.USER.EXCLUDE Statement	124
10.3.8	AS.USER.NONNDS Statement	124
10.3.9	ASAMDIR Statement	125
10.3.10	AUTHENTICATION Statement	125
10.3.11	CODEPAGE Statement	126
10.3.12	DEBUGLOGFILE Statement	126
10.3.13	DEBUGTOSTDOUT Statement	126
10.3.14	DIRECTTOAUTHENTICATION Statement	127
10.3.15	ENTROPY Statement	127
10.3.16	HONORMVSDISABLE Statement	127
10.3.17	IGNORESTANDARDEXCLUDES Statement	128
10.3.18	KEY Statement	128
10.3.19	LOCALE Statement	129
10.3.20	PASSWORDPROMPT Statement	129
10.3.21	PASSWORDPROMPTCURRENT Statement	129
10.3.22	PASSWORDPROMPTCHANGE Statement	129
10.3.23	PASSWORDPROMPTCHANGEAGAIN Statement	130
10.3.24	PLATFORMNAME Statement	130
10.3.25	PASSWORDSOURCE Statement	130
10.3.26	POLLINT Statement	131
10.3.27	POLLRAND Statement	131
10.3.28	PROVISIONING Statement	131
10.3.29	RUNMODE Statement	132
10.3.30	SECURITY Statement	132
10.3.31	SMF Statement	133
10.3.32	SYSLOGFACILITY Statement	133
10.3.33	TRACEFILE Statement	133
10.3.34	TRACETOSTDOUT Statement	134
10.3.35	UPDATEPASSWORD Statement	134
10.3.36	UPDATESAMBA Statement	134
10.3.37	USEFILEIPC Statement	135
10.4	Using Include and Exclude Configuration Statements	135
10.4.1	Mask Characters and Examples	136
10.4.2	Rules by Which Masks Are Matched Against User IDs and Groups	136

Part IV Platform Services Administration 139

11 Installing Platform Services 141

11.1	About Platform Services for i5/OS	141
11.1.1	The System Intercept	141
11.1.2	The Platform Receiver	141
11.1.3	Receiver Scripts	142
11.1.4	Authentication Services	142
11.2	Platform Services Installation Procedure	143
11.3	Uninstalling Platform Services	143

12 Configuring and Administering Platform Services 145

12.1	Platform Certificate Management	145
12.2	Administering Platform Services	145
12.2.1	Starting and Stopping the Platform Receiver	145
12.2.2	Platform Receiver Command Line Parameters	146
12.2.3	Maintaining Files Used by the Platform Receiver	146

13 Troubleshooting Platform Services	149
13.1 Obtaining Debugging Output	149
13.2 Troubleshooting Identity Provisioning	149
13.3 Troubleshooting Network Issues	150
Part V API Development	151
14 About the API	153
14.1 Using the API in the z/OS Environment	153
14.2 Using the API in the OS/400 Environment	154
14.3 Using the API in the Linux/UNIX Environment	154
14.4 API Function List	154
15 C Language API Reference	157
ASC_ADMINRSTPASSWD	158
ASC_CHGPASSWD	161
ASC_CHKPASSWD	163
ASC_DAYS	166
ASC_GETCONTEXT	167
ASC_GRPMEM	170
ASC_INIT	173
ASC_INIT_EXT	175
ASC_LISTSEQV	177
ASC_READATTR	180
ASC_RIGHTS	183
ASC_SECEQUAL	186
ASC_STRERROR	188
ASC_TERM	189
ASC_USER_INCLUDE_EXCLUDE	191
16 Java Language API Reference	193
Class com.novell.asam.JAscAuth.JAscAuth	194
Classes Used by checkPassword	202
Exception Classes in com.novell.asam.JAscAuth	204
17 API Examples	207
17.1 Adding API Support to the Apache Web Server	207
17.2 Adding API Support to the QUALCOMM POP Server	207
17.3 Adding API Support to SASL	208
17.4 Adding API Support to SSH Secure Shell	208
17.5 Adding API Support to TACACS+	208
Part VI Appendixes	209
A OS/400 Provisioning Tips	211
A.1 Provisioning Attributes	211

A.2	Synchronizing Password Expiration	211
B	Password Change Validation Exit	213
C	Troubleshooting the API	215
D	Messages	219
D.1	Message Format	220
D.2	Message Destination	221
D.2.1	i5/OS	222
D.2.2	z/OS	222
D.2.3	Linux and UNIX	222
D.2.4	Windows	222
D.3	AGT Messages	222
D.4	ASC Messages	226
D.5	AUDA Messages	230
D.6	AUDG Messages	233
D.7	AUDR Messages	235
D.8	AXML Messages	241
D.9	CFG Messages	243
D.10	CFGGA Messages	244
D.11	CFGP Messages	245
D.12	CRT Messages	245
D.13	DIR Messages	247
D.14	DOM Messages	258
D.15	DRVCOM Messages	259
D.16	EJS Messages	259
D.17	HES Messages	269
D.18	LWS Messages	269
D.19	NET Messages	275
D.20	OAP Messages	275
D.21	OBJ Messages	276
D.22	PLS Messages	293
D.23	PRCV Messages	293
D.24	RDXML Messages	298
D.25	W3LM Messages	300
Glossary		303

About This Guide

This guide provides you with the information you need to administrate the Novell® Identity Manager Fan-Out driver for Midrange.

The Fan-Out driver supports multi-platform implementation of Novell Identity Manager 3.6, 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.

The guide includes the following parts:

- ♦ Part I, “Concepts and Facilities,” on page 13
- ♦ Part II, “Core Driver Administration,” on page 37
- ♦ Part III, “Platform Services Planning,” on page 103
- ♦ Part IV, “Platform Services Administration,” on page 139
- ♦ Part V, “API Development,” on page 151
- ♦ Part VI, “Appendixes,” on page 209
- ♦ “Glossary” on page 303

Audience

This guide is for system administrators and others who plan for, install, configure, use, and troubleshoot the Identity Manager Fan-Out driver. It assumes that you are familiar with Identity Manager, Novell eDirectory™, and the administration of systems and platforms you connect to Identity Manager.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation, or go to [the Documentation Feedback site \(http://www.novell.com/documentation/feedback.html\)](http://www.novell.com/documentation/feedback.html) and enter your comments there.

Documentation Updates

For the most recent version of this guide, visit [the Identity Manager 3.6 Drivers Documentation Web site \(http://www.novell.com/documentation/idm36drivers\)](http://www.novell.com/documentation/idm36drivers).

Additional Documentation

For additional documentation about Identity Manager drivers, see [the Identity Manager 3.6 Drivers Documentation Web site \(http://www.novell.com/documentation/idm36drivers\)](http://www.novell.com/documentation/idm36drivers).

For documentation about Identity Manager, see [the Identity Manager 3.6 Documentation Web site \(http://www.novell.com/documentation/idm36\)](http://www.novell.com/documentation/idm36).

For documentation about other related Novell products, such as eDirectory and iManager, see [the Documentation Web site’s product index \(http://www.novell.com/documentation\)](http://www.novell.com/documentation).

Documentation Conventions

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

A trademark symbol (® , ™ , etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark.

When a single pathname can be written with a backslash for some platforms or a forward slash for other platforms, the pathname is presented with a backslash. Users of platforms that require a forward slash, such as Linux or UNIX, should use forward slashes as required by your software.

Concepts and Facilities

Part I describes the concepts and facilities of the Novell® Identity Manager Fan-Out Driver. It includes the following chapters:

- ♦ Chapter 1, “Introduction,” on page 15
- ♦ Chapter 2, “Structure and Function,” on page 17
- ♦ Chapter 3, “Examples,” on page 31

Introduction

1

The Novell® Identity Manager 3.6 Fan-Out Driver is an identity provisioning solution, based on Novell eDirectory™, Identity Manager, and related technology.

With Identity Manager, you can manage the full user life cycle, delivering first-day access to essential resources, providing single login, and modifying or revoking access rights. Identity Manager also provides self-service features that enable users to maintain their own passwords and profile information.

By adding the Fan-Out Driver, you can use Identity Manager to *fan out* identity provisioning to hundreds of systems with minimal effort. You can centrally manage user accounts and have them automatically created, configured, maintained, and removed when appropriate. It uses extensible scripts to manage account rights, home directories, and other resources as well as the user definitions themselves. Meanwhile, system administrators for the individual platforms in your enterprise can retain control over their areas.

New Features In This Release

The Fan-Out Driver includes these new features:

- ◆ An Embedded Remote Loader to provide eDirectory, Identity Manager, and the Fan-Out Core Driver with greater individual resources and fault tolerance while maintaining a simple configuration.
- ◆ A configurable administrative proxy user to allow the Fan-Out Driver with a better controlled LDAP identity.
- ◆ Support for 64-bit Linux for Core Driver systems.
- ◆ New central logging for platform responses.
- ◆ Support for new connecte platform systems, including HP-UX* Itanium*, z/Series* Linux and Tru64*.
- ◆ Support for non-interactive platform installations.
- ◆ New installers and configuration wizards, supporting native UNIX package managers.
- ◆ Additional improvements in:
 - ◆ Scalability
 - ◆ Memory efficiency
 - ◆ Caching

1.1 Driver Highlights

The Fan-Out Driver differs from other Identity Manager drivers in the following ways:

- ◆ **Fan-Out Scalability** A single instance of the Fan-Out Driver can provision centrally managed account information to hundreds of dissimilar platform systems throughout your enterprise.
- ◆ **Scripts** The Fan-Out Driver scripts process data change events on all supported platforms. This enables platform system administrators to automatically manage local resources associated with accounts as well as the definitions of those accounts.

- ◆ **Authentication Redirection** It supports the Novell Universal Password feature, which employs access to a central repository for login and password rules. This includes support of bidirectional password synchronization.
- ◆ **Application Programming Interface (API)** An easy-to-use API enables programmers to extend applications from individual platforms to fully leverage eDirectory.

1.2 Driver Organization

The Fan-Out Driver has two functional divisions.

- ◆ **Authentication Services** Provides real-time Identity Vault (eDirectory) access for user authentication and related purposes.
- ◆ **Identity Provisioning** Provides user and group management.

The Fan-Out Driver has two structural divisions.

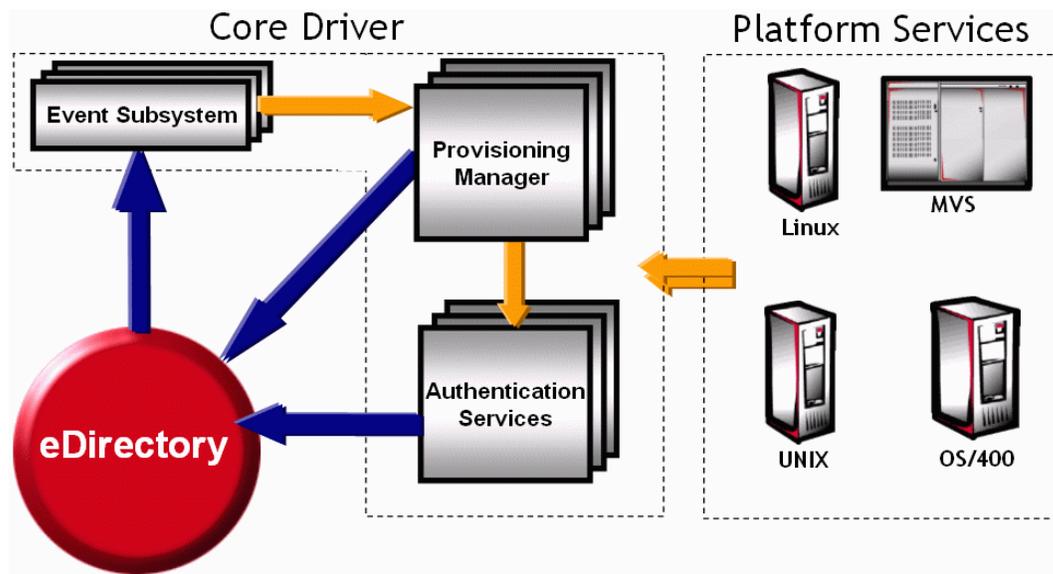
- ◆ **The Core Driver** Interfaces with eDirectory to provide Authentication Services (such as password verification) and provisioning events (such as Add User or Remove Group).
- ◆ **Platform Services** Uses the Core Driver to bring common authentication and account life cycle management to a broad selection of supported platforms.

Structure and Function

2

There are two structural divisions of the Novell® Identity Manager Fan-Out Driver: the Core Driver and Platform Services.

Figure 2-1 Fan-Out Driver Components



The Core Driver provides Authentication Services and information about changes to users and groups to target platforms that have been configured to run Platform Services.

The driver obtains and stores the information it uses in an eDirectory™-based “Identity Vault.” To access the Identity Vault, the Core Driver uses LDAP Services for eDirectory.

For ease of management, target platforms that share the same user and group population are grouped together into Platform Sets.

Communication between the driver components occurs through TCP/IP and is encrypted.

The driver includes a secure Web interface that works as an iManager plug-in for administration and monitoring.

The Core Driver records significant occurrences in an Audit Log, and each component writes an Operational Log. Each Core Driver component maintains performance statistics, which can be viewed in the Web interface.

The Fan-Out Driver includes an application programming interface (API). This allows programmers to extend applications to use Authentication Services, which allows them to take advantage of your existing eDirectory constructs.

Binary files, configuration files, and other files used by the driver components are stored in the ASAM directory in the file system of the host server.

Additional Resources

Details about configuring and administering the Core Driver and Platform Services are provided in later sections of this guide. Other sections provide information about API development and driver system messages. Also be aware that this is one of three available administration guides for the Fan-Out Driver, each tailored to the range of platforms with which it can work:

- ♦ *Identity Manager Fan-Out Driver for Linux and UNIX Administration Guide*
- ♦ *Identity Manager Fan-Out Driver for Mainframes Administration Guide (z/OS)*
- ♦ *Identity Manager Fan-Out Driver for Midrange Administration Guide (OS/400, i5/OS)*

For information about eDirectory, see the *Novell eDirectory Administration Guide*.

Section Topics

The topics in this section describe the structure and function of the Identity Manager Fan-Out Driver.

- ♦ [Section 2.1, “Core Driver,” on page 18](#)
 - ♦ [“Object Services” on page 20](#)
 - ♦ [“Event Journal Services” on page 20](#)
 - ♦ [“Audit Services” on page 20](#)
 - ♦ [“Certificate Services” on page 20](#)
 - ♦ [“Web Services” on page 20](#)
 - ♦ [“Authentication Services” on page 20](#)
 - ♦ [“Event Subsystem” on page 21](#)
 - ♦ [“Embedded Remote Loader” on page 21](#)
- ♦ [Section 2.2, “Platform Services,” on page 21](#)
- ♦ [Section 2.3, “Directory Objects,” on page 25](#)
- ♦ [Section 2.4, “Migration,” on page 29](#)

2.1 Core Driver

The Core Driver can be further broken down into two main parts:

- ♦ The Driver Shim
- ♦ The objects that represent Core Driver properties and functionality in Identity Vault (eDirectory)

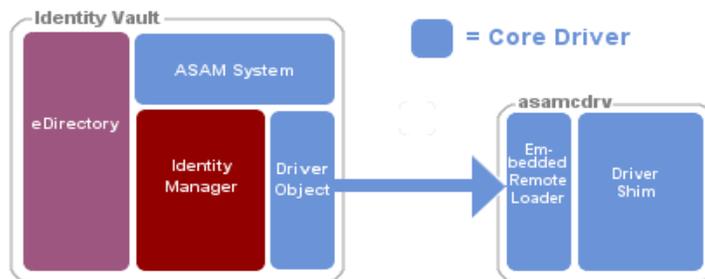
The Shim is the installed driver software that provides authentication services, such as password verification, to target platforms. It provides identity provisioning events, such as add, modify, and delete, for users and groups, to target platforms. It also uses its Event Subsystem to retrieve events from Identity Manager. Finally, the Core Driver includes an imbedded remote loader that replaces the functionality of the standard remote loader used by Identity Manager.

The objects used in the Identity Vault to store Core Driver properties and functionality include:

- ◆ The Driver object, which stores configuration information about the Core Driver.
- ◆ The ASAM System container object, which stores configuration and user management information for users connecting to other systems via the Fan-Out Driver.

A writable replica of the partition holding the ASAM System container must reside on the LDAP host server used by a Core Driver. A User object, configured during installation, is used by the driver to perform an LDAP Bind for access to eDirectory.

Figure 2-2 Core Driver



In summary, the Core Driver provides these functions:

- ◆ eDirectory access to platforms for Authentication Services, such as password verification.
- ◆ Provisioning events to Platform Services for the maintenance of local user accounts and groups.
- ◆ The Web interface that you use to configure and manage the driver
- ◆ Management of the objects inside the ASAM System container
- ◆ An audit trail of significant occurrences

You can run multiple Core Drivers to provide redundancy for Authentication Services and Identity Provisioning functions.

One Core Driver is designated as the primary Core Driver. Other Core Drivers are secondary Core Drivers. Only the primary Core Driver listens for events from eDirectory. The primary Core Driver also serves the Web interface and provides environmental information during the installation process for other Core Drivers.

2.1.1 Core Driver Component Details

The software architecture of the Core Driver includes eight main components. Five of the components are collectively referred to as the Provisioning Manager:

Provisioning Manager Components

Descriptions of each component in Provisioning Manager follows.

Object Services

Object Services maintains the objects within the ASAM System container. Some of these objects store configuration information for the various driver components. Others represent users and groups of users that can be defined on target platforms. The object that contains these users and groups is called the Census.

Object Services on the primary Core Driver is notified by the Event Subsystem of events, such as add, modify, or delete, pertaining to users and groups of users in eDirectory. These events are used to maintain the Census.

To initially build and periodically ensure the integrity of the Census, Object Services examines specified portions of eDirectory for users and groups. This process is called a Trawl. You can use the Web interface to set the Trawl schedule. Only the primary Core Driver performs Trawls.

Census Search objects that you define using the Web interface describe which objects in eDirectory are included in the Census. Platform Set Search objects that you define using the Web interface describe which users and groups are managed for a given set of platforms.

For more information about Object Services and the Census, see [“Census Container” on page 26](#). For more information about associating users and groups with sets of platforms, see [“Platform Set Objects” on page 28](#).

Event Journal Services

Event Journal Services receives provisioning events from Object Services and makes them available to sets of platforms according to the rules you specify. Event Journal Services ensures that provisioning events for a platform are delivered, even if the platform is not always available.

Platforms can periodically connect to Event Journal Services to receive provisioning events, or they can maintain a persistent connection and receive events as they occur.

By defining multiple Core Drivers to provide events to platforms, you can provide for improved availability.

Audit Services

Audit Services maintains the Audit Log and Operational Logs for a Core Driver.

Certificate Services

Certificate Services mints the certificates used by Secure Sockets Layer (SSL) to authenticate and secure connections between the components.

Web Services

Web Services provides the secure Web interface for monitoring and administering the Identity Manager Fan-Out Driver. The Web interface is provided through an iManager plug-in.

Authentication Services

Authentication Services provides Platform Services with the time-critical interface to eDirectory. This interface is used for such functions as checking the passwords of users logging in to the platform. This interface is also used by the AS Client API.

By defining multiple Core Drivers to provide Authentication Services to platforms, you can provide for improved performance and availability.

Authentication Services supports platform communications using SSL and DES encryption.

Event Subsystem

The Event Subsystem uses Identity Manager to subscribe to eDirectory events and provides them to Object Services. Objects of interest must be replicated on the Core Driver server.

Embedded Remote Loader

Identity Manager includes a software component known as the Remote Loader. It is used to interface with drivers on the various systems that can be connected to Identity Manager.

The Core Driver bypasses this component, using its own Embedded Remote Loader. The resulting tighter integration provides eDirectory, Identity Manager, and the Fan-Out Core Driver with greater individual resources and fault tolerance while maintaining a simple configuration

2.2 Platform Services

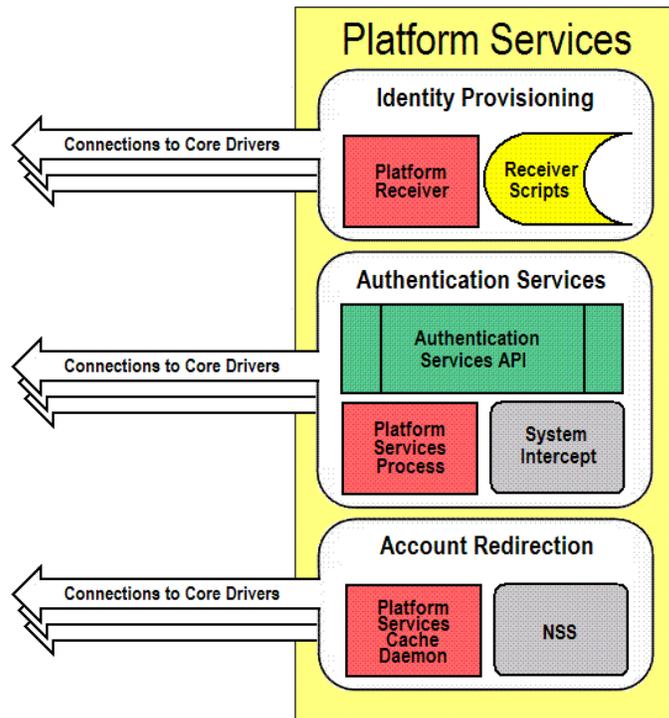
Platform Services enables a system to utilize the Core Driver functions. A platform can use Authentication Services for some or all users, and can use Identity Provisioning in maintaining some or all local user accounts and groups. For a complete account redirection solution, a platform can use the Name Service Switch and Platform Services Cache Daemon for some or all users.

Some types of platforms communicate with Authentication Services using SSL, and others use DES encryption. All platform communication with Event Journal Services uses SSL.

A platform that uses SSL-based communication must have a valid certificate to communicate with the Core Driver for most functions. A platform that uses DES encryption must use the same DES key as defined for it in the Core Driver configuration.

The Identity Manager Fan-Out Driver does not support authentication or password changes for eDirectory users who have a null password.

Figure 2-3 Platform Services



2.2.1 User and Group Management

Management of users and groups on the platform is carried out by Receiver scripts, which are called by the Platform Receiver based on provisioning events obtained from the Core Driver.

Platform Receiver

The Platform Receiver connects to the Event Journal Services component of the Core Driver, requests provisioning events, and runs a script to carry out the appropriate platform-specific processing for the given type of event. The Platform Receiver provides failover support for connections to Event Journal Services if more than one Core Driver is available.

Receiver Scripts

Receiver scripts are run by the Platform Receiver to process provisioning events.

The Identity Manager Fan-Out Driver provides a set of fully functional base scripts in the customary scripting language for each supported platform. You can extend these base scripts as appropriate for your needs.

The Receiver script functions are

- ◆ Add User
- ◆ Modify User
- ◆ Delete User

- ◆ Delete User Pending
- ◆ Enable User
- ◆ Disable User
- ◆ Rename User
- ◆ Add User to Group
- ◆ Remove User from Group
- ◆ Add Group
- ◆ Modify Group
- ◆ Delete Group
- ◆ Delete Group Pending
- ◆ Rename Group

2.2.2 User Authentication

Authentication redirection is handled by the Platform Services Process, which is called by the System Intercept. The Platform Services Process is also called by applications using the AS Client API.

Platforms that use password replication receive notification of password changes in eDirectory through the Platform Receiver and send notification of local password changes detected by the password change intercept to the Core Driver using the Platform Services Process.

Account redirection is handled by the Platform Services Cache Daemon, which is called by the Name Service Switch. Platforms that are configured for account redirection use a local memory cache pool for account records and retrieve all account and password information from this cache.

Platform Services Process

The Platform Services Process establishes and maintains connections to Core Drivers for Authentication Services, and provides load balancing and failover among them. These connections are used to provide Authentication Services to the platform.

Platform Services Cache Daemon

The Platform Services Cache Daemon establishes and maintains a connection to a Core Driver and receives event data from Event Journal Services. This data is stored away in memory cache and used to supply account information to the Name Service Switch.

AS Client API

The AS Client API provides a programming interface to Authentication Services. It is furnished as routines that can be called from C and Java*. The AS Client API includes functions to

- ◆ Validate a user ID/password combination
- ◆ Change a user's password, given the current password
- ◆ Perform an administrative password reset
- ◆ Obtain the fully distinguished name for a user ID
- ◆ Determine if a user has Security Equal To a given object

- ◆ Determine if an object has the specified effective rights to the specified attribute of a given object
- ◆ Obtain a list of members of a group
- ◆ Obtain a list of security equivalences for a user
- ◆ Obtain the eDirectory Home Directory attribute value for a user
- ◆ Determine if a given user is in the Authentication Services Include/Exclude list

For details about using the AS Client API, see [Part V, “API Development,” on page 151](#).

System Intercept

The System Intercept is called by the native security system for password verification and password change. Because passwords are checked using eDirectory or, on supported platforms, replicated from eDirectory, a user has the same password throughout the enterprise, regardless of the platform used.

System Intercepts are implemented using standard, vendor-provided mechanisms.

Authentication Services Methods

There are two methods for providing users with the same password across the platforms in your enterprise.

Password Redirection: Requests to check passwords are intercepted at the platform and redirected to objects in eDirectory. The end result is that the user has the same password on all systems.

Password Replication: Changes to passwords are intercepted and replicated between eDirectory and participating platforms. As with password redirection, the end result is that the user has the same password on all systems.

The following table shows the Authentication Services methods available for each platform OS type:

Table 2-1 *Authentication Services by Operating System*

Platform OS Type	Authentication Services Method
z/OS	Password redirection, Password replication (optional)
OS/400	Password replication
UNIX	Password redirection, Password replication (optional)

Password Redirection

Platforms that use password redirection employ a System Intercept to gain control when a password is to be verified. The System Intercept passes the request to Authentication Services, through the Platform Services Process. Authentication Services uses the Census to identify the User or Alias object in eDirectory that corresponds to the request. Then Authentication Services verifies the password using that object and returns the result to the platform.

The System Intercepts for z/OS* and UNIX systems store the password in the local security system upon a successful authentication or password change. For logins, if Authentication Services cannot be reached, the user's password is verified using the local security system.

Password Replication

Platforms that use password replication receive notification of password changes through the Platform Receiver.

The Core Driver is notified of changes to passwords as follows:

- ♦ By ensuring that your eDirectory is configured to fully support Universal Password, the driver is notified of password changes in eDirectory.
- ♦ The Password Validation Program Exit is installed on an OS/400* system and captures password change information.

When Authentication Services receives notification of a password change, it verifies the authenticity of the notification and then stores the encrypted password. This is detected by the Event Subsystem, which generates the appropriate provisioning event to notify those platforms that are authorized to receive password information.

By default, passwords are converted to lowercase before they are sent to a platform.

Account Redirection: Requests for Posix user and group information are intercepted at the platform Name Service Switch and redirected to objects in eDirectory. This information includes loginName, uidNumber, gidNumber, gecos, homeDirectory, loginShell, groupName, memberUid and passwords.

2.2.3 Platform Configuration File

You use the platform configuration file to specify Platform Services configuration information, such as

- ♦ Which users are authenticated using Authentication Services and which users are authenticated using the local security system
- ♦ Which user accounts and groups are managed using Identity Provisioning and which are managed locally
- ♦ Information used to locate the Core Driver servers.

2.3 Directory Objects

The Identity Manager Fan-Out Driver maintains objects in eDirectory with configuration information for the Core Driver and platforms as well as users and groups of users available to the platforms. These are stored in the ASAM System container.

You maintain configuration information by using the Web interface. Do not use any other method of changing objects in the ASAM System container unless advised by support personnel.

A writable replica of the partition holding the ASAM System container must reside on the LDAP host server used by a Core Driver.

2.3.1 The ASAM Master User Object

The Core Driver processes perform an LDAP Bind as the ASAM Master User to gain access to eDirectory. The ASAM Master User object can be specified during installation.

The ASAM Master User must have Supervisor rights to the container in eDirectory that holds the users and groups that can be added to the Census. This is known as the User and Group Subtree. These rights are granted during installation.

To use the AS Client API to access objects outside of the User and Group Subtree, you must grant additional rights to the ASAM Master User.

- ◆ You must grant the ASAM Master User Browse object rights and Compare property rights to any object that is accessed through the AS Client API.
- ◆ You must grant the ASAM Master User Read property rights to any object whose Security Equals list or Group Membership list, or other attribute value is accessed through the AS Client API.

2.3.2 Configuration-Oriented Objects

Configuration information for Identity Manager Fan-Out Driver components is stored in objects that correspond to them.

- ◆ Audit Services object
- ◆ Certificate Services object
- ◆ Event Journal Services object
- ◆ Object Services object
- ◆ Web Services object
- ◆ Event Subsystem objects
- ◆ Authentication Services objects
- ◆ UID/GID Set objects
- ◆ Platform Set objects
- ◆ Platform objects

Identity Manager Fan-Out Driver program component configuration objects list each of their host server network addresses. Before accepting a communication connection from another component, driver components verify that the connection originates from a network address listed in the corresponding configuration object.

2.3.3 Census Container

Based on your specifications, Object Services maintains a Census of users and groups of users for use with target platforms. Users in the Census are represented by Enterprise User (eUser) objects. Groups of users in the Census are represented by Enterprise Group (eGroup) objects.

Object Services uses events from the Event Subsystem to maintain the Census. Object services of the primary Core Driver also periodically trawls eDirectory for information to ensure the validity of the Census.

Authentication Services uses eUser objects in the Census to locate the corresponding User objects in eDirectory for password verification and other functions. For information about associating eUsers and eGroups from the Census with sets of platforms for provisioning purposes, see [“Platform Set Objects” on page 28](#).

You use the Web interface to specify Census Search objects that identify the users and groups that are to be included in the Census.

Search objects can get Enterprise Users from

- ◆ Specifically identified User objects
- ◆ Group object membership
- ◆ Organizational Role object occupant lists
- ◆ Objects in containers (and subcontainers, to whatever depth you set)

Search objects can get Enterprise Groups from

- ◆ Specifically identified Group objects
- ◆ Group objects in containers (and subcontainers, to whatever depth you set)
- ◆ Identity Manager entitlements

Dynamic Groups as Search Objects

Dynamic groups use an LDAP search filter to define a set of rules that, when matched by eDirectory User objects, define the members of the group. Membership in the group is evaluated dynamically by eDirectory. There is no actual list of members for a dynamic group like there is for a static group.

Events involving users who are already in the Census that affect their membership in a dynamic group that is a Search object are seen by the Event Subsystem as they happen. This is because the Core Driver interrogates Search objects to discover if an event involving a given User object is of interest.

Events involving users not already in the Census and events involving the LDAP search filter of a dynamic group that is a Search object are not seen by the Event Subsystem. This is because there is nothing to drive the LDAP search. Such changes are not detected until the next Trawl is run.

Naming Exceptions

Because Enterprise User objects and Enterprise Group objects share the same name space, their names must be unique. If a duplicate name is found based on your Census Search objects, the resulting Enterprise User or Enterprise Group object is placed in the Exceptions container rather than being made available in the Census. You can use the Web interface to review naming exceptions.

Enterprise User Objects

Enterprise User (eUser) objects reside in the Census container. An eUser object represents a single User object, or an Alias object that references a User object, in eDirectory.

An eUser object includes a reference to the User object or Alias object that it represents in eDirectory. The User object referenced by an Alias object is provisioned to platforms, not the Alias object itself.

Enterprise Group Objects

Enterprise Group (eGroup) objects reside in the Census container. An eGroup object represents a group of users, and is based on a Group object, or an Alias object that references a Group object, in eDirectory. Enterprise Group objects must be based on static Group objects. Dynamic Group objects are not provisioned.

An eGroup object includes a reference to the Group object or Alias object that it represents in eDirectory. The Group object referenced by an Alias object is provisioned to platforms, not the Alias object itself.

Enterprise Group objects in the Census contain a list of the eUser objects that are represented in the corresponding Group object in eDirectory (but not any users that are not present in the Census).

Inactive Users and Groups

You can choose to have Enterprise User and Enterprise Group objects whose corresponding User or Group object is deleted from eDirectory or is no longer covered by a Census Search object remain in the Census in an inactive state. Because Enterprise User objects relate to User objects in eDirectory through a globally unique identifier, this prevents another person from receiving access to resources as an unintended result of the reuse of the user name. Inactive users cannot authenticate through Authentication Services.

Delete Pending Duration

You can use the Web interface to specify a Delete Pending Duration. During this interval, eUser and eGroup objects whose corresponding User and Group objects have either been deleted from eDirectory or are no longer covered by a Search object are not deleted from target platforms. The results of a Delete User or Delete Group Receiver script can be difficult to reverse. Delete Pending Duration provides a grace period to allow recovery from a disastrous mistake affecting many users.

The Delete User Pending or Delete Group Pending Receiver script is called when a delete event becomes pending for a user or group. The Delete User or Delete Group script is not called until the Delete Pending Duration expires.

2.3.4 Platform Objects

A Platform object represents a specific target platform that runs Platform Services.

2.3.5 Platform Set Objects

Platform Set objects provide the relationship between Search objects and Platform objects. You can use Platform Sets to group together multiple platforms that share the same user and group population.

The following example illustrates how you can fan out your user and group population to platforms that are grouped into Platform Sets.

Table 2-2 Platform Set Example

Containers with Users	OU=Students Henri Markus Rie	OU=Faculty Carmen Eleu Mario	OU=Staff Isabel Claire Kenji
------------------------------	-------------------------------------	-------------------------------------	-------------------------------------

Search Objects	OU: Students Include Users: Yes	OU: Faculty Include Users: Yes	OU: Staff Include Users: Yes
Platform Sets	Academic Search Objects: Students, Faculty Platforms: StudentDataServer, LabWorkstation1, LabWorkstation2, LabWorkstation3	Employee Search Objects: Faculty, Staff Platforms: BenefitsServer, EmployeeDataServer	Everyone Search Objects: Students, Faculty, Staff Platforms: MailServer, LibraryServer
Platforms	<p>StudentDataServer</p> <p>Platform Set: Academic Users: Henri, Markus, Rie, Carmen, Eleu, Mario</p> <p>LabWorkstation1</p> <p>Platform Set: Academic Users: Henri, Markus, Rie, Carmen, Eleu, Mario</p> <p>LabWorkstation2</p> <p>Platform Set: Academic Users: Henri, Markus, Rie, Carmen, Eleu, Mario</p> <p>LabWorkstation3</p> <p>Platform Set: Academic Users: Henri, Markus, Rie, Carmen, Eleu, Mario</p>	<p>BenefitsServer</p> <p>Platform Set: Employee Users: Carmen, Eleu, Mario, Isabel, Claire, Kenji</p> <p>EmployeeDataServer</p> <p>Platform Set: Employee Users: Carmen, Eleu, Mario, Isabel, Claire, Kenji</p>	<p>MailServer Platform Set: Everyone Users: Henri, Markus, Rie, Carmen, Eleu, Mario, Isabel, Claire, Kenji</p> <p>LibraryServer Platform Set: Everyone Users: Henri, Markus, Rie, Carmen, Eleu, Mario, Isabel, Claire, Kenji</p>

2.4 Migration

In some cases, a system other than eDirectory might contain the users that you want to participate with the driver. There are tools, such as LDIF, that you can use to import these users into eDirectory.

If you cannot extract the passwords for the affected user accounts, you can use the driver Password Migration component. This component can help you accomplish a smooth transition to basing your user accounts in eDirectory. The Password Migration component is available only on z/OS platforms. For details about the Password Migration component, see the *Identity Manager Fan-Out Driver for Mainframes Administration Guide*.

Examples

3

This section presents some examples of processing to illustrate how the various components of the Novell®-Identity Manager Fan-Out Driver work together. These examples do not exhaustively describe each detail involved in the processing, but give a representative account of the steps involved.

- ♦ Section 3.1, “Password Check for Login,” on page 32
- ♦ Section 3.2, “User Added to eDirectory,” on page 32
- ♦ Section 3.3, “Census Trawl,” on page 33
- ♦ Section 3.4, “User Deleted from eDirectory,” on page 34
- ♦ Section 3.5, “Group Deleted from eDirectory,” on page 35
- ♦ Section 3.6, “User Added to a Group,” on page 35

Use [Figure 3-1](#) and [Figure 3-2](#) for reference as you study the examples.

Figure 3-1 Driver Components

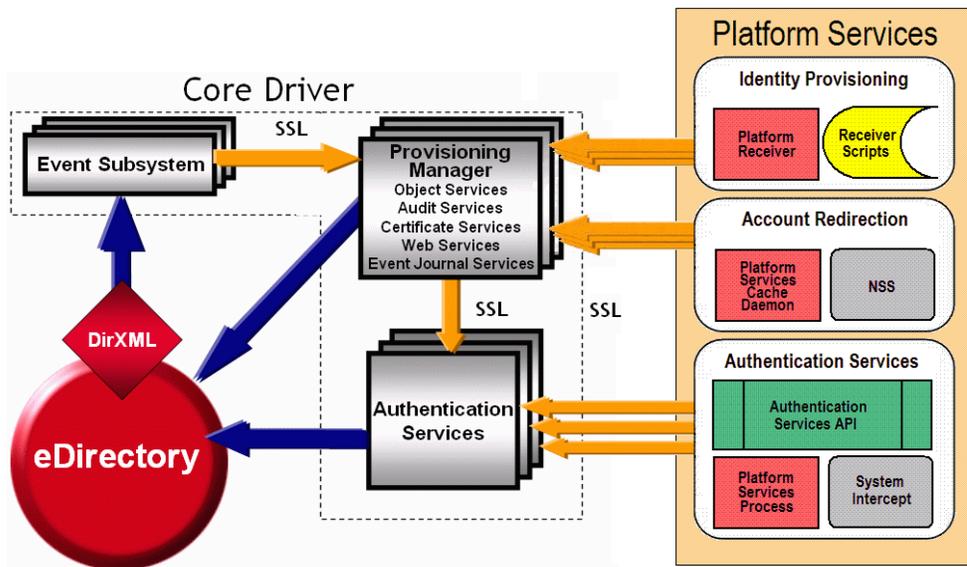
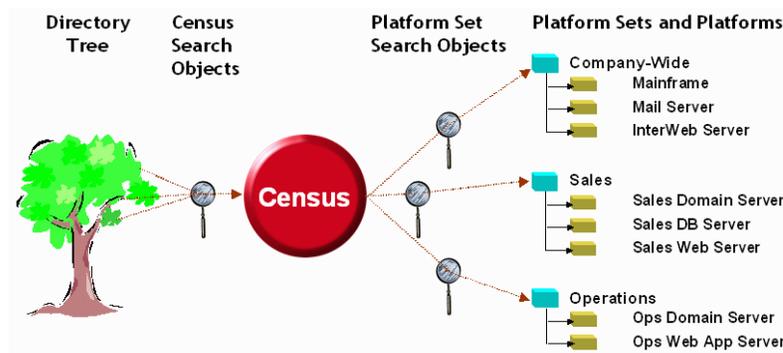


Figure 3-2 User and Group Population Information Flow



3.1 Password Check for Login

A user logs in to a platform.

1. The user enters user ID and password information in response to a login prompt from the operating system, and the System Intercept receives control.
2. The System Intercept calls the Check Password API function (unless the user is excluded from processing based on the specifications in the platform configuration file).
3. The Platform Services Process uses its load-balancing algorithm to select a Core Driver for Authentication Services. (Platform Services establishes a connection with each Core Driver for Authentication Services upon startup.)
4. The Platform Services Process makes a Check Password request to Authentication Services.
5. Authentication Services obtains from the Census the eUser object whose name matches the user ID. Authentication Services gets from that eUser object the distinguished name of the corresponding User object (or Alias object) in eDirectory™.
6. Authentication Services checks the password against the object in eDirectory that corresponds to the eUser.
7. If the password is not already present in the eUser object, Authentication Services stores the password there for the Provisioning Manager to use for password replication.
8. Authentication Services returns the result of the Check Password request to the Platform Services Process.
9. Authentication Services notifies Audit Services, which records the action in the Audit Log.
10. The Platform Services Process returns the result to the System Intercept.
11. The System Intercept returns the result to the local security system.

3.2 User Added to eDirectory

An administrator adds a new user to eDirectory. The user is covered by a Census Search object.

1. An administrator adds the new user to eDirectory.
2. The Event Subsystem receives the change and notifies Object Services.
3. If the user is covered by a Census Search object, Object Services of the primary Core Driver creates an eUser object for the user in the Census container, and associates the user with the Platform Set container objects whose Platform Set Search objects cover the user.

If the common name of the new user is the same as a name that already exists in the Census container, its eUser object is instead created in the Exceptions container, and the exception must be resolved by an administrator. For guidance in avoiding and resolving exceptions, see the [Part II, “Core Driver Administration,” on page 37](#).

4. Object Services notifies Event Journal Services.
5. When each Platform Receiver of the associated Platform Sets requests an event and this event is the next one for that platform, Event Journal Services obtains detailed information about the new user by reading its object from eDirectory and passes the provisioning event to the Platform Receiver.

If Event Journal Services cannot obtain the new user information yet because directory synchronization is not complete, the next event for the platform is processed and this one is tried again later.

6. Each Platform Receiver that receives the provisioning event checks to see if a user by that name already exists (unless the user is excluded from processing based on specifications in the platform configuration file).

If the user already exists, the Platform Receiver notifies Event Journal Services.

If the user does not exist, the Platform Receiver calls the Add User Receiver script, which adds the new user to the local security system and prepares it for use. The Platform Receiver then notifies Event Journal Services of the script outcome.

7. Event Journal Services notifies Audit Services, which records the action in the Audit Log.

3.3 Census Trawl

Object Services of the primary Core Driver periodically performs a Trawl to verify the contents of the Census. A Trawl is also run to initially build the Census, or a part of it, whenever you use the Web interface to define a new Census Search object.

The following steps are performed for each Census Search object:

1. Object Services scans the Census Search object for users and groups.
2. For any user or group that does not have a corresponding eUser or eGroup in the Census container:
 - a. Object Services creates an eUser or eGroup object in the Census container, and associates the user or group with the Platform Set container objects whose Platform Set Search objects cover the user or group.

If the common name of the new user or new group is the same as a user or group that already exists in the Census container, the eUser or eGroup object is instead created in the Exceptions container, and the exception must be resolved by an administrator. For guidance in avoiding and resolving exceptions, see [Part II, “Core Driver Administration,” on page 37](#).

- b. Object Services notifies Event Journal Services.
- c. When each Platform Receiver of the associated Platform Sets requests an event and this event is the next one for that Platform, Event Journal Services obtains detailed information about the new user or group by reading its object from eDirectory and passes the provisioning event to the Platform Receiver.

If Event Journal Services cannot obtain the information yet because directory synchronization is not complete, the next event for the platform is processed and this one is tried again later.

- d. Each Platform Receiver that receives the provisioning event checks to see if a user or group by that name already exists (unless the user or group is excluded from processing based on specifications in the platform configuration file).

If the user or group already exists, the Platform Receiver notifies Event Journal Services.

If the user or group does not exist, the Platform Receiver calls the Add User or Add Group Receiver script, which adds the new user or group to the local security system and prepares it for use. The Platform Receiver then notifies Event Journal Services of the script outcome.

- e. Event Journal Services notifies Audit Services, which records the action in the Audit Log.

The following steps are performed for each user and group in the Census.

1. Object Services verifies that the user or group is still covered by a Search object.

If it does not, the same steps are followed as for [Section 3.4, “User Deleted from eDirectory,” on page 34](#) or [Section 3.5, “Group Deleted from eDirectory,” on page 35](#).

2. Object Services verifies that the User object or Group object that corresponds to the user or group still exists in eDirectory.

If it does not, the same steps are followed as for [Section 3.4, “User Deleted from eDirectory,” on page 34](#) or [Section 3.5, “Group Deleted from eDirectory,” on page 35](#).

3.4 User Deleted from eDirectory

A User object that is covered by a Census Search object is deleted from eDirectory.

1. An administrator deletes a user from eDirectory.
2. The Event Subsystem receives the deletion and notifies Object Services.
3. If the user is covered by a Census Search object, then Object Services takes one of the following actions based on configuration information that you have specified:

Object Services marks the corresponding eUser object in the Census as inactive. (Inactive users cannot authenticate through Authentication Services.)

or

Object Services marks the eUser object for deletion after the event has been processed by all associated platforms.

4. Object Services notifies Event Journal Services.
5. When each Platform Receiver of the associated Platform Sets requests an event and this event is the next one for that Platform, Event Journal Services passes the provisioning event to the Platform Receiver. When the last Platform Receiver of a Platform Set has received the event, the next Trawl removes the Platform Set association for the eUser (if you have defined your configuration to remove deleted users rather than mark them inactive).

6. Each Platform Receiver that receives the provisioning event calls its Disable/Delete User Receiver script to disable the user in the local security system or to delete it and clean up its resources (unless the user is excluded from processing based on specifications in the platform configuration file).
7. Event Journal Services notifies Audit Services, which records the action in the Audit Log.

If you have specified a Delete Pending Duration, Event Journal Services indicates to the Platform Receiver that a delete is pending for the user. When the Delete Pending Duration has expired, Event Journal Services delivers the delete event.

3.5 Group Deleted from eDirectory

A Group object that is covered by a Census Search object is deleted from eDirectory.

1. An administrator deletes a group from eDirectory.
2. The Event Subsystem receives the deletion and notifies Object Services.
3. If the group is covered by a Census Search object, then Object Services takes one of the following actions based on configuration information that you have specified:
Object Services marks the corresponding eGroup object in the Census inactive.
or
Object Services marks the eGroup object for deletion after the event has been processed by all associated platforms.
4. Object Services notifies Event Journal Services.
5. When each Platform Receiver of the associated Platform Sets requests an event and this event is the next one for that platform, Event Journal Services passes the provisioning event to the Platform Receiver. When the last Platform Receiver of a Platform Set has received the event, the next Trawl removes the Platform Set association for the eGroup.
6. Each Platform Receiver that receives the provisioning event calls its Delete Group Receiver script to delete the group from the local security system and clean up its resources.
7. Event Journal Services notifies Audit Services, which records the action in the Audit Log.

If you have specified a Delete Pending Duration, Event Journal Services indicates to the Platform Receiver that a delete is pending for the group. When the Delete Pending Duration has expired, Event Journal Services delivers the delete event.

3.6 User Added to a Group

A user for which there is an eUser object in the Census is added to the member list of a Group object in eDirectory.

1. An administrator adds the user to the member list of the Group object.
2. The Event Subsystem receives the change and notifies Object Services.
3. Object Services notifies Event Journal Services.
4. When each Platform Receiver of the Platform Sets associated with both the eUser and the eGroup requests an event and this event is the next one for that platform, Event Journal Services obtains detailed information about the user by reading its object from eDirectory, and passes the provisioning event to the Platform Receiver.

If Event Journal Services cannot yet obtain updated user information due to incomplete directory synchronization, the next event for the platform is processed and this one is tried again later.

5. Each Platform Receiver that receives the provisioning event calls its Add User to Group Receiver script, which adds the user to the group in the local security system.
6. Event Journal Services notifies Audit Services, which records the action in the Audit Log.

Core Driver Administration



Part II provides you the information you need to administer the Core Driver component of the Novell® Identity Manager Fan-Out Driver. It includes the following chapters:

- ♦ Chapter 4, “Core Driver Planning,” on page 39
- ♦ Chapter 5, “Installing the Core Driver,” on page 51
- ♦ Chapter 6, “Configuring and Administering the Core Driver,” on page 71
- ♦ Chapter 7, “Troubleshooting the Core Driver,” on page 99

Core Driver Planning

4

This section helps you plan for your deployment of the Novell® Identity Manager Fan-Out Driver. If the Fan-Out Driver is new to you, read the information presented earlier in [Part I, “Concepts and Facilities,” on page 13](#) before proceeding.

Major topics in this section include

- ◆ [Section 4.1, “Configuration Planning,” on page 39](#)
- ◆ [Section 4.2, “Configuration and Performance Guidelines,” on page 41](#)
- ◆ [Section 4.3, “Requirements,” on page 44](#)
- ◆ [Section 4.4, “Installation Planning Worksheet,” on page 46](#)

4.1 Configuration Planning

There are a number of issues to resolve in planning for your deployment of the Identity Manager Fan-Out Driver. Considering these issues now will make your installation go more smoothly.

- ◆ Decide how you will deploy the use of the driver throughout your enterprise.

Do you want to start with a small subset of your platform systems? Do you want to start with a small subset of your user community?

Platforms can operate with Include/Exclude lists to control which users the driver handles for authentication, and which users the driver defers to the native authentication mechanism. Platforms can also operate with Include/Exclude lists to control which user accounts the driver manages using provisioning events and which user accounts are managed locally. For more information, see [Part III, “Platform Services Planning,” on page 103](#).

- ◆ Decide who will administer your driver configuration.

The Web interface is used to monitor and administer the driver. Make it available to these persons and ensure that they have the necessary rights to use it. For details about the rights needed for administrative functions, see [Section , “Rights Required for Web Application Use,” on page 76](#)

- ◆ Decide which eDirectory™ servers in your network will run Core Drivers.

A writable replica of the partition holding the ASAM System container must reside on the LDAP host server used by a Core Driver.

Each object that is covered by a Census Search object must be present in a replica (full or filtered) on the system that hosts the primary Core Driver.

- ◆ Will you install additional Core Drivers to provide redundancy for Authentication Services?

The Platform Services Process includes load balancing and failover support to provide for continued processing should a Core Driver become unavailable.

- ◆ Will you install additional Core Drivers to provide redundancy for Identity Provisioning?

The Platform Receiver includes failover support to provide for continued processing if the Core Driver it normally uses for Identity Provisioning becomes unavailable.

- ♦ Decide where in your eDirectory tree the ASAM System container and ASAM Master User objects should go. Creating a special container for them is a good practice.
Make sure you set password policies appropriate for the ASAM Master User object.
For more information about requirements for the ASAM Master User, see [Section 6.2.2, “ASAM Master User Security,” on page 72.](#)
- ♦ Decide upon your Census parameters.
Which objects in your eDirectory tree will be used as the source of Enterprise Users and Enterprise Groups? This depends on how you place User and Group objects in your directory.
When do you want to run a Census Trawl? Because the Census is maintained in real time using provisioning events, Trawls are used primarily to verify the consistency of the Census. Once a day is reasonable for most cases.
Do you want Enterprise Users whose User objects have been deleted from eDirectory to be automatically removed from the Census? They can be removed after remaining inactive for a specified number of days, or you can choose to manage inactive users manually.
Do you want to delay user password expiration until the end of the day of expiration? This can result in smoother operation for users on platforms with third-party systems that cache and reuse passwords during the day.
Do you want to immediately delete users and groups from platforms when they are deleted from eDirectory or are no longer covered by a Search object, or do you want to provide a grace period to recover from accidental changes?
For information about setting these parameters, see [“Configuring the Census” on page 79.](#)
- ♦ How will you resolve naming exceptions?
For more information, see [Section 6.5.14, “Reviewing Naming Exceptions,” on page 96.](#)
- ♦ Decide which systems in your network will run Platform Services.
User names, passwords, and group names must conform to the character set and length restrictions imposed by the platform operating system in order to participate in Authentication Services and Identity Provisioning on that platform. Determine how you will handle those that do not meet the restrictions.
- ♦ Decide how you will organize your Platform Sets. Each platform belongs to exactly one Platform Set. A Platform Set provides the relationship between a group of platforms and the Search objects that define their user and group population.
- ♦ Users and Groups have the same UID number and GID number on each Linux/UNIX platform in a Platform Set.
What UID and GID numbers do you want to reserve for local administrator use on Linux/UNIX platforms?
Will you use the RFC2307 posixAccount and posixGroup auxiliary object classes for enterprise-wide UID and GID assignments?
- ♦ Will any of your platforms use password replication? If so, you must ensure that the driver is notified of changes to passwords.
If your eDirectory is configured to fully support Universal Password, the driver is notified of password changes in eDirectory.
If you do not use Universal Password, you must install and configure the appropriate password intercepts.
For more information, see [“Password Replication Requirements” on page 44.](#)

- ◆ Platforms configured to use password replication do not normally receive provisioning events for user accounts until the passwords for these accounts are known to the driver.

The driver uses system intercepts or Universal Password to collect password information. Users must either change their passwords where these are installed and configured, or authenticate on a driver platform before they can be populated onto a platform that uses password replication (Permit Password Replication specified as Yes for the Platform object in the Web interface).

By planning a staged deployment of the Fan-Out Driver to the platforms in your enterprise so that most users have authenticated using other platforms first, you ensure the availability of these users to password replication platforms when you are ready to deploy the driver on them.

- ◆ You can use Identity Manager to provision accounts to a different tree from the eDirectory tree you install the driver in. You can use the Novell Client™ Password Intercept to provide password change information back to the driver for password replication purposes from this tree. For more information, see [“External Password Sources” on page 84](#).
- ◆ Consider how your own applications could benefit from the use of the Authentications Services (AS) Client API.

Using the AS Client API is simple and straightforward. For more information about using the Client API, see [Part V, “API Development,” on page 151](#).

- ◆ Will any of your Platform Receiver scripts need attributes other than those configured in the driver by default?

For a list of the attributes configured by default, see [Section 4.3.3, “Core Driver Requirements,” on page 45](#).

To configure additional attributes, you must add them to the Event Subsystem Subscriber filter. For details about adding attributes to the Subscriber filter, see the *Identity Manager Administration Guide*.

The attribute names that you use in the Subscriber filter must be the eDirectory names.

If you are using attributes on an OS/400 platform, you might need to map attribute names to their System Distribution Directory field names. This is done using the Attribute Name Mapping file. For more information about the Attribute Name Mapping file on OS/400, see the *Identity Manager Fan-Out Driver for Midrange*.

4.2 Configuration and Performance Guidelines

Many factors affect the performance of the Identity Manager Fan-Out Driver. Performance is most critical for Authentication Services, such as Check Password and Get Context.

There are many relationships within the driver, and one or more of the factors described in the following sections can affect all of these relationships. Use the following as guidelines in planning and troubleshooting your Fan-Out Driver installation.

Acceptable Authentication Services performance is achievable using two or three low-end servers for Core Drivers. However, if your present network experiences problems, such as slow logins related to eDirectory, Fan-Out Driver operations will experience similar response problems.

For fault tolerance, your configuration should include Core Drivers running on several servers.

For fault tolerance, each Core Driver should use a different LDAP host server.

For optimal performance, each Core Driver and its LDAP host server should run on the same server.

Topics in this section include

- ◆ [Section 4.2.1, “eDirectory,” on page 42](#)
- ◆ [Section 4.2.2, “Object Services and the Event Subsystem,” on page 42](#)
- ◆ [Section 4.2.3, “Event Journal Services,” on page 43](#)
- ◆ [Section 4.2.4, “Authentication Services,” on page 43](#)
- ◆ [Section 4.2.5, “Platform Systems,” on page 43](#)
- ◆ [Section 4.2.6, “Platform Services / Authentication Services Relationship,” on page 44](#)

4.2.1 eDirectory

Tuning eDirectory on your network is beyond the scope of this document. Much documentation on this subject is available elsewhere, including Novell Technical Information Documents (TIDs), which are available at the [Novell Support Web site \(http://support.novell.com\)](http://support.novell.com). The health and performance of eDirectory is critical to the ability of the driver to respond to Authentication Services requests and to deliver provisioning events in a timely manner. Therefore, the health and performance of eDirectory should be your starting point in doing any performance planning and troubleshooting with the driver.

Factors in driver performance relative to eDirectory include

- ◆ The size of the eDirectory tree
- ◆ Communication links between the LDAP host servers used by Core Drivers and servers holding replicas of the ASAM System container and other objects referenced by the driver
- ◆ LAN traffic
- ◆ Size of partitions containing relevant objects
- ◆ Performance of CPU and disks in servers holding relevant replicas
- ◆ Amount of memory in servers holding relevant replicas

The driver interfaces with eDirectory through LDAP. For LDAP tuning guidance, see the *Novell eDirectory Administration Guide*.

4.2.2 Object Services and the Event Subsystem

The Object Services component of the Core Driver is primarily responsible for maintaining the Census and other objects in the ASAM System container. Object Services receives provisioning events from the Event Subsystem, updates the Census as required, and passes the provisioning events to Event Journal Services. It is important for Identity Provisioning that the Core Driver be running at all times, but it is mostly a background process that does not require a great deal of processing power and is, for the most part, not a time-critical process.

Object Services performs Trawls to initially build and to verify the Census by performing a series of requests based on the Census Search objects defined in your configuration. For each Organizational Unit represented in the configuration, Object Services issues a single request to eDirectory to return all the objects contained in the given Organizational Unit.

Focus your Search objects to the specific directory locations of your users and groups rather than specifying a top level container object. This provides better feedback information during a Trawl and reduces the likelihood of an LDAP time-out because of slow servers or slow network links.

The Event Subsystem uses the Identity Manager to provide events to Object Services. The Event Subsystem requires minimal processing power, but it does require replicas for all objects that are monitored. Network connectivity and eDirectory synchronization are the primary performance factors for the Event Subsystem.

For optimal performance, a writable partition of all replicas containing objects contained in the Census should reside on the same server as the LDAP host server used by a Core Driver. However, be aware that operations that lock the directory on the local server, such as running NDSRepair, sometimes delay requests or cause them to fail.

4.2.3 Event Journal Services

Event Journal Services waits for Platform Receivers to connect, then provides pending events and a snapshot of User and Group objects for processing. Network connectivity to the platforms, and proximity of the Core Driver to servers holding replicas of managed User and Group objects are the primary performance factors for Event Journal Services.

Platforms with very large numbers of managed users and groups should be connected to Event Journal Services with connections of adequate bandwidth to ensure that Full Sync Mode and Check Mode processing will complete within an acceptable time.

To reduce the number of concurrent connections that must be serviced by a Core Driver host, avoid using Persistent Mode on Platform Receivers.

4.2.4 Authentication Services

Authentication Services is responsible for processing requests made by Platform Services.

For optimal performance, LDAP host servers used by Core Drivers should hold a writable replica that contains the User objects represented in the Census, and other objects that might be referenced often by Authentication Services.

4.2.5 Platform Systems

Platform Services sends requests to the Core Driver. The systems on which Platform Services reside can be anything from a desktop workstation to a high-end mainframe system. The inherent performance of these systems is based on a number of factors, including

- ◆ System load
- ◆ The power of the system
- ◆ Network traffic
- ◆ Connectivity and bandwidth to the Core Drivers
- ◆ The number of Core Drivers defined in the configuration

Consider each of these as you configure each platform and as you select the location of the Core Drivers.

4.2.6 Platform Services / Authentication Services Relationship

The performance of the Platform Services / Authentication Services transaction is the most important performance relationship in the driver. The communication relies on the TCP/IP stack of the platform and Authentication Services server. TCP/IP configuration on the platform, the Authentication Services server, and the routers in between is the most important factor in the performance of servicing Authentication Services requests. Guidelines for configuring TCP/IP are beyond the scope of this section. Refer to appropriate Novell and platform operating system documentation and TIDs for further information.

The Maximum Transmission Unit (MTU) TCP/IP parameter has been shown in testing to have an appreciable effect on performance in a situation in which the Authentication Services server is low on power. See Novell TID 2911035 (on the [Support Knowledgebase Web page \(http://support.novell.com\)](http://support.novell.com)) for more information on this topic.

Platform system planners should be aware of a mandatory three-second delay in reporting a bad password on a password check request. This delay is in eDirectory itself. It cannot be configured by the driver.

4.3 Requirements

The system requirements for driver components are described in the following sections. Identity Manager Fan-Out Driver components do not require the systems they run on to be dedicated solely to them.

Topics in this section include

- ♦ [Section 4.3.1, “User Rights Requirements,” on page 44](#)
- ♦ [Section 4.3.2, “Password Replication Requirements,” on page 44](#)
- ♦ [Section 4.3.3, “Core Driver Requirements,” on page 45](#)
- ♦ [Section 4.3.4, “Requirements for Workstations Used for Installation and Administration,” on page 46](#)
- ♦ [Section 4.3.5, “Platform Services Requirements,” on page 46](#)

4.3.1 User Rights Requirements

The installation and configuration of the driver require a user with full administrative rights and privileges in eDirectory and on the target systems. You can grant more limited rights to other users to use the Fan-Out Driver Web interface for administrative functions. For details of rights needed for administrative functions, see [Section , “Rights Required for Web Application Use,” on page 76](#).

4.3.2 Password Replication Requirements

If you use password replication, you must ensure that the driver is notified of changes to passwords.

- ♦ If your eDirectory is configured to fully support Universal Password, the driver is notified of password changes in eDirectory.

- ♦ The OS/400 Password Validation Program Exit provides password change information from OS/400 platforms.
- ♦ When configuring the policy for Universal Password, be sure to select the option that allows administrative users to retrieve the Universal Password.

For information about installing and configuring the password intercepts, see [Part IV, “Platform Services Administration,”](#) on page 139.

4.3.3 Core Driver Requirements

- Novell Identity Manager.
- Novell eDirectory versions supported by the Identity Manager version in use.
- Novell iManager versions supported by Identity Manager version in use.
- One of the following OS platforms, in a version supported by the Identity Manager and eDirectory version in use:
 - ♦ Windows
 - ♦ Linux
 - ♦ Solaris
- TCP/IP network connectivity.
- A writable replica of the partition that will hold the ASAM System container must reside on the LDAP host server used by the Core Driver.
- Replicas (full or filtered) of objects that will be covered by a Census Search object (primary Core Driver only).

The Fan-Out Driver will be configured for the attributes in the following lists. If you use filtered replicas, include the attributes shown in the following lists. If you add other attributes to the Subscriber filter, you must ensure that they are also available in your filtered replicas.

Alias Attributes

- ♦ Aliased Object Name
- ♦ CN
- ♦ GUID

User Attributes

- ♦ CN
- ♦ Group Membership
- ♦ GUID
- ♦ Login Disabled
- ♦ Surname

ASAM-enterpriseUser Attributes

- ♦ ASAM-addTime
- ♦ GUID

Group Attributes

- ♦ CN

- ◆ GUID
- ◆ Member

Organizational Role Attributes

- ◆ CN
- ◆ GUID
- ◆ Role Occupant

TIP: iManager provides a wizard for setting up filtered replicas.

4.3.4 Requirements for Workstations Used for Installation and Administration

The workstations used to install, configure, and administer the driver must meet the following requirements.

- TCP/IP network connectivity.
- The ability to run iManager.
- Connectivity to the Identity Vault (eDirectory) tree to be managed by the driver.
- If the installation computer runs UNIX, `gzip` and `tar` utilities.
- Connectivity to the file system of the computer that is to receive components being installed. If the installation computer is not the same as the target host, a drive must be mapped to the target host.

4.3.5 Platform Services Requirements

For information about required systems and software, as well as supported platforms and operating environments, see the [Identity Manager 3.6 Drivers Documentation Web site \(http://www.novell.com/documentation/idm36drivers\)](http://www.novell.com/documentation/idm36drivers). From this index page, you can select a Readme file associated with the platform(s) for which you need Fan-Out Driver support.

4.4 Installation Planning Worksheet

You must provide certain information during the installation process. The Installation Planning Worksheet provides you with a convenient place to concisely record that information ahead of time for reference during the installation process.

For an overview of the installation process, see [Section 5.1, “Preparing for Core Driver Installation,” on page 51](#).

4.4.1 Installation Planning Worksheet

For a detailed explanation of the items on the Installation Planning Worksheet, see [“Items on the Installation Planning Worksheet” on page 48](#).

Census Search Objects

Object Name	Include Users	Include Groups	Expand	Include Aliases	Depth
-------------	---------------	----------------	--------	-----------------	-------

Administrative Users

User Object	Rights
-------------	--------

UID/GID Sets

UID/GID Set Name	Reserved Range Lower Bound	Reserved Range Upper Bound
------------------	----------------------------	----------------------------

Platform Sets

Platform Set Name	UID/GID Set	Search Objects
-------------------	-------------	----------------

Platforms

Platform Name	Platform Set	Permit PW Replication	Network Address
---------------	--------------	-----------------------	-----------------

4.4.2 Items on the Installation Planning Worksheet

This section describes each of the items on the Installation Planning Worksheet

Census Search Objects

Record Census Search object names and attributes. These describe how the Census is to be populated.

Object Name

The fully distinguished name of each Census Search object.

A Census Search object can be a User object, a Group object, an Organizational Role object, an Organization container object, or an Organizational Unit container object.

Include Users

Whether or not User objects covered by this Census Search object are added to the Census.

Include Groups

Whether or not Group objects covered by this Census Search object are added to the Census.

Expand

Whether or not users who are members of Group objects or occupants of Organizational Role objects covered by this Census Search object are added to the Census.

Include Aliases

Whether or not Alias objects covered by this Census Search object are added to the Census. The user or group object that is represented by an alias object is provisioned to platforms.

Depth

For container Census Search objects only: Indicate how many steps down the directory tree hierarchy beyond the Census Search object the driver searches for users and groups to add to the Census. Zero causes the search to stop at the container Search object.

Administrative Users

The users to grant rights to administer the driver and the types of functions you will allow them to perform. For details, see [Section , “Rights Required for Web Application Use,” on page 76.](#)

UID/GID Sets

Establish UID/GID set names and reserved ranges.

UID/GID Set Name

The common name to give each UID/GID set in the ASAM System container.

Reserved Range Lower Bound and Reserved Range Upper Bound

The low and high bounds of the range to reserve for the system administrator to use locally on the platform. The driver does not assign UID/GID numbers in the reserved range.

Platform Sets

A Platform Set provides the relationship between the Search objects that describe a collection of users and groups, and one or more platform systems that share the same users and groups.

Platform Set Name

The name to give each Platform Set configuration object in the ASAM System container.

UID/GID Set

The name of a previously defined UID/GID Set that is to be used to assign UID numbers and GID numbers for Linux/UNIX users and groups associated with the Platform Set.

Leave this option empty to use the PosixAccount and posixGroup uidNumber and gidNumber attributes.

Search Objects

Search objects identify the users and groups to be used to populate the platforms that you add to this Platform Set.

A Platform Set Search object can be a User object, a Group object, an Organizational Role object, an Organization container object, an Organizational Unit container object, or a Dynamic Group object.

Platforms

Include for each individual platform.

Platform Name

The common name to give the Platform Configuration object in the ASAM System container.

Platform Set

The name of the Platform Set this platform is to belong to.

Password Replication

Whether or not requests for password replication information from eDirectory are honored for this platform.

Network Address

The IP address or DNS name of the platform to be used for TCP/IP communications with other driver components. If your platform uses more than one network interface, list all of the addresses.

Installing the Core Driver

5

Earlier sections of this *Administration Guide* identify two major parts of the Novell® Identity Manager Fan-Out Driver, which are the Core Driver and Platform Services. The Core Driver must be installed first.

Before beginning the Core Driver installation, you should complete the planning process described in [Chapter 4, “Core Driver Planning,”](#) on page 39, and you should be familiar with the topics presented in [Part I, “Concepts and Facilities,”](#) on page 13.

Topics in this section include

- ♦ [Section 5.1, “Preparing for Core Driver Installation,”](#) on page 51
- ♦ [Section 5.2, “Step-By-Step Installation Instructions,”](#) on page 55
- ♦ [Section 5.3, “Activating the Driver After Evaluation,”](#) on page 69
- ♦ [Section 5.4, “Performance Tuning,”](#) on page 69

5.1 Preparing for Core Driver Installation

Please review this section carefully for a high-level overview of the tasks and considerations you will encounter during the installation of the Core Driver. This information will help you later as you determine which steps are relevant to your particular installation scenario(s).

5.1.1 Essentials

- ♦ Verify that you meet minimum system requirements. For details, see [Section 4.3, “Requirements,”](#) on page 44.
- ♦ To improve your readiness, complete the worksheet in [Section 4.4, “Installation Planning Worksheet,”](#) on page 46 before you begin. You can refer to it during the installation process.
- ♦ Obtain the Core Driver distribution package for your target operating system from the [Novell downloads site \(http://download.novell.com\)](http://download.novell.com). In other words, you will need the package that is designed for the operating environment in which Identity Manager is running.
- ♦ Always check the [Novell Support Web Site \(http://support.novell.com\)](http://support.novell.com) for the latest support pack and product update information. Check the Release Notes and Readme files for the version you are installing for any special actions that might be required.

5.1.2 Other Advance Considerations

Topics in this section include:

- ♦ [“Migrating From NetWare”](#) on page 52
- ♦ [“Installing the LDAP Client SDK Libraries”](#) on page 52
- ♦ [“Specifying Primary and Secondary Core Drivers”](#) on page 52
- ♦ [“Fan-Out Driver Schema Extensions for Identity Manager 3.5”](#) on page 53
- ♦ [“Complete Checklist of Considerations Before Installation”](#) on page 53

Migrating From NetWare

The release of Identity Manager 3.6 ends support for NetWare. Therefore, if you wish to upgrade a Core Driver running in a NetWare environment, you will need to migrate to one of the other supported environments (Linux, Solaris, Windows).

You can do this by completing one of the following step-by-step installation tasks for upgrading a Core Driver, depending on your environment:

- ♦ “Upgrading a Local Core Driver Shim on Linux or Solaris” on page 57
- ♦ “Upgrading a Local Core Driver Shim on Windows Systems” on page 61

As you follow these steps, bear in mind that any information you provide about a Core Driver during this task should reflect the identity, settings and configuration of the Core Driver you are migrating from NetWare.

Installing the LDAP Client SDK Libraries

If you meet one of the following conditions, you will need to install the LDAP Client SDK libraries prior to the Core Driver installation:

- ♦ You are installing the 64-bit Linux Fan-Out Core Driver
- ♦ You are installing the Core Driver on a Linux or Solaris system not hosting eDirectory

To install and configure the LDAP Client SDK:

- 1 Download the installation package labeled *LDAP Libraries for C* from the [Novell Developer Site \(http://developer.novell.com\)](http://developer.novell.com) suitable for your system architecture.
 - 2 Install the package according to the instructions that accompany it.
 - 3 Setup your runtime library path to include the newly installed files:
 - ♦ On Linux, you may modify your `/etc/ld.so.conf` to include the additional path. For example:
`/opt/novell/ndk/novell-cldap-devel-2007.10.04-1linux64/lib64/`
 - ♦ Alternatively, you can modify the `LD_LIBRARY_PATH` environment variable to include your path:

```
LD_LIBRARY_PATH=/opt/novell/ndk/novell-cldap-devel-2007.10.04-1linux64/lib64/:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
```
- Be sure to modify your logon profile and Core Driver startup script to include this.

Specifying Primary and Secondary Core Drivers

During software installation, you will be asked if you are establishing a primary Core Driver or adding a secondary Core Driver. Following are some guidelines for determining how to respond:

- ♦ You must have one primary Core Driver. If you are installing a Core Driver for the first time, it will automatically be designated as the primary.

- ♦ The primary Core Driver must have access to a read/write replica of the entire ASAM System container and all User and Group objects defined by the Census.
- ♦ Secondary drivers can service authentication requests and deliver events to connected platforms but will not perform tasks such as trawls or update enterprise objects in the Census. Therefore, the primary Core Driver must be active and running in order to provide connected platforms with new provisioning information.

For additional information on assessing secondary driver requirements, see [Section 5.4, “Performance Tuning,” on page 69](#).

Fan-Out Driver Schema Extensions for Identity Manager 3.5

The Identity Manager 3.6 Fan-Out Driver is compatible with Identity Manager 3.5. However, if you do install it on an Identity Manager 3.5 system, you will need to take the extra step of manually installing the Fan-Out driver’s schema extensions.

- ♦ If you are running eDirectory on Linux or Solaris, locate the `/fan-out/Schema/fanout.sch` schema definition file from the installation media and run the `ndssch` utility with the following command:

```
ndssch admin.acme fanout.sch
```
- ♦ If you are running eDirectory on Windows, follow these steps:
 1. Open Novell eDirectory Services from the Windows Control Panel.
 2. Select the *install.dlm* entry in the list and click the *Start* button.
 3. Select *Install additional schema files* and click *Next*.
 4. Enter your administrative user name, context and password.
 5. Enter the file `\fan-out\Schema\fanout.sch` from your installation media and click *Finish*.
- ♦ Secondary drivers can service authentication requests and deliver events to connected platforms but will not perform tasks such as trawls or update enterprise objects in the Census. Therefore, the primary Core Driver must be active and running in order to provide connected platforms with new provisioning information.

Complete Checklist of Considerations Before Installation

- ♦ A *Quick Start* guide for installing the Fan-Out Driver is available for each target platform. Although this *Administration Guide* includes detailed procedures for all installation scenarios, you may find the *Quick Start* helpful in focusing on primary steps. The quick starts, listed below, are available at the [Identity Manager 3.6 Drivers Documentation Web site \(http://www.novell.com/documentation/idm36drivers\)](http://www.novell.com/documentation/idm36drivers).
 - ♦ *Fan-Out Driver Installation Quick Start for Linux and UNIX Systems*
 - ♦ *Fan-Out Driver Installation Quick Start for Midrange Systems*
 - ♦ *Fan-Out Driver Installation Quick Start for Mainframe Systems*
- ♦ Be sure to pre-install the LDAP SDK Libraries if you meet one of the unique conditions discussed in [“Installing the LDAP Client SDK Libraries” on page 52](#)
- ♦ During software installation, you will be asked if you are establishing a primary Core Driver or adding a secondary Core Driver. For guidelines, see [“Specifying Primary and Secondary Core Drivers” on page 52](#).

- ♦ If you install version 3.6 of the Identity Manager Fan-Out Driver on an Identity Manager 3.5 system, you will need to manually install driver's schema extensions as discussed in [“Fan-Out Driver Schema Extensions for Identity Manager 3.5” on page 53](#).
- ♦ To complete the Core Driver installation you will use one of two available application interfaces for configuration:

iManager Newer versions of this standard Novell Web interface include a Fan-Out Driver application plug-in for driver configuration. The Core Driver software includes a copy of this plug-in in case you have an older version of iManager. The installation instructions include steps for installing this plug-in after you have run the initial installation software.

Designer This interface, which comes as part of the Identity Manager 3.6 product, is an offline tool you can use to plan and model large deployments of the Fan-Out Driver. Designer includes its own Fan-Out Driver application plug-in, which is already installed as part of the Designer interface. For more information on Designer, see [Section 6.4, “Applications For Configuration,” on page 75](#).
- ♦ Once you have installed the Core Driver and completed its initial configuration in iManager, you still won't be able to test the installation until you have installed Platform Services on the system(s) you will connect to. This will involve an additional software installation and configuration on each of these systems. Therefore, you may want to preview Part IV of this *Administration Guide*, “Platform Services Administration,” for details about this additional process.
- ♦ Installation of the Core Driver will create an ASAM directory in the file system on each server that includes any of its components. Access to each copy of this directory should be restricted to the driver itself and its administrators to ensure protection of sensitive identity information.

5.1.3 General Installation Sequence

Following is a general overview of the process for installing the Core Driver.

NOTE: This section is provided to help you prepare for installation. More detailed instructions are provided later in [Section 5.2, “Step-By-Step Installation Instructions,” on page 55](#).

- 1 Read [Section 5.1.1, “Essentials,” on page 51](#) and [Section 5.1.2, “Other Advance Considerations,” on page 51](#).
- 2 Know in advance which of the following installation scenarios you wish to perform:
 - ♦ New installation of a primary Core Driver running on Linux, Solaris or Windows
 - ♦ New installation of a secondary Core Driver running on Linux, Solaris or Windows
 - ♦ Upgrade of an existing Core Driver running in “Local” mode (default, not using Remote Loader) running on Linux, Solaris or Windows
 - ♦ Upgrade of an existing Core Driver running in “Remote” mode (already using Novell Remote Loader) running on Linux, Solaris or Windows
 - ♦ Upgrade of an existing Core Driver running on NetWare
- 3 Run the Core Driver installation program and respond to the prompts. This will install the Core Driver software components also known as the Driver Shim.
- 4 If necessary, install the iManager plug-in for the Fan-Out Driver Web application.

- 5 Using iManager and the plug-in, create objects in the Identity Vault to support the Core Driver. This includes importing an XML default configuration file that comes with the Core Driver installation software.
- 6 Populate your Census with the users and groups that you will use for your initial testing. This includes defining Census Search objects and then running a Census Trawl. For details about this procedure, see [“Configuring the Census” on page 79](#).
- 7 Assign users of the Fan-Out Web program (in iManager) the rights they need. For details, see [Section , “Rights Required for Web Application Use,” on page 76](#).
- 8 If you will use the Fan-Out driver to connect to Linux or UNIX platforms, define the UID/GID sets that you will use for your initial testing. For details, see [“Configuring Linux/UNIX UID/GID Sets” on page 93](#).
- 9 Define the Platform Sets that you will use for your initial testing. For details, see [“Configuring Platform Sets” on page 87](#).
You must define at least one UID/GID Set before you can define a Platform Set.
- 10 Define the platforms that you will use for your initial testing. For details, see [“Configuring Platforms” on page 89](#).
- 11 Use iManager to start the Core Driver object in Identity Manager.
- 12 Use system tools to start the Driver Shim in the the local operating environment.
- 13 Install and configure the desired Platform Services to match the platforms you defined in iManager during the previous steps.

IMPORTANT: This step involves individual software installations and configurations on each system you will connect to with the Fan-Out Driver. For detailed information about this separate process, see Part IV of this *Administration Guide*, “Platform Services Administration.”

- 14 After testing, install additional Core Drivers for performance and redundancy according to the guidelines in [Section 5.4, “Performance Tuning,” on page 69](#).
- 15 Before the 90-day evaluation period expires, activate the Identity Manager Fan-Out Driver. You can use the driver for evaluation purposes for 90 days. The driver will not work thereafter unless it has been activated. For details, see [Section 5.3, “Activating the Driver After Evaluation,” on page 69](#).
- 16 Fully deploy the Fan-Out Driver throughout your enterprise as you gain confidence and experience.

5.2 Step-By-Step Installation Instructions

This section presents the various step-by-step tasks that can be combined to cover all Core Driver installation scenarios. The tasks are grouped first into four basic categories:

- ♦ [Section 5.2.1, “Installing the Driver Shim on Linux or Solaris,” on page 56](#)
- ♦ [Section 5.2.2, “Installing the Driver Shim on Windows Systems,” on page 60](#)
- ♦ [Section 5.2.3, “Setting Up the Core Driver in iManager,” on page 64](#)
- ♦ [Section 5.2.4, “Other Tasks Following Installation,” on page 67](#)

NOTE: Be aware of the following:

- ♦ Identity Manager 3.6 does not support NetWare. Therefore, if you wish to upgrade a Core Driver running in a NetWare environment, you will need to migrate to one of the other supported environments. See [“Migrating From NetWare” on page 52](#) for more details.
 - ♦ Some of the configuration tasks discussed briefly in this section are covered again in more detail in [Chapter 6, “Configuring and Administering the Core Driver,” on page 71](#)
-

5.2.1 Installing the Driver Shim on Linux or Solaris

Core Driver installation on Linux and UNIX begins with one of the following tasks, depending on your scenario:

- ♦ [“Installing a New Core Driver Shim on Linux or Solaris” on page 56](#)
- ♦ [“Upgrading a Local Core Driver Shim on Linux or Solaris” on page 57](#)
- ♦ [“Upgrading a Remote Core Driver Shim on Linux or Solaris” on page 58](#)

Installing a New Core Driver Shim on Linux or Solaris

To install a new Driver Shim on Linux or Solaris:

- 1 From your installation media, locate and execute the appropriate self-extracting installer:

```
sh linux_x86_coredriver.bin
sh linux_x86_64_coredriver.bin
sh solaris_sparc_coredriver.bin
```

NOTE: The x86 (32-bit) installer is compatible with both 32-bit and 64-bit versions of Linux. To use the x86_64 (64-bit) installer, you must first install and configure the 64-bit LDAP SDK (see [“Installing the LDAP Client SDK Libraries” on page 52](#)).

- 2 Accept the license, select your installation directory and proceed to install the product files.
- 3 The installer will next assist you in configuring the Driver Shim. You will be prompted for the Remote Loader password, which is used to encrypt driver network communications. Enter a password and remember it, as you will use it when configuring the driver in iManager.
- 4 You will then be prompted for a Driver Object password. This is used to access the driver object in eDirectory. This password will also be used when configuring the driver in iManager.
- 5 The next entry you are asked for is the eDirectory server/port, so the installer can retrieve an SSL certificate from eDirectory using your SSL-configured LDAP server. Enter the DNS name or IP address of the LDAP server that the Core Driver Shim will use to communicate with. Typically, this will be *localhost* on port *636*.
- 6 When prompted for an eDirectory admin name/password, enter the eDirectory administrator’s ID in LDAP dot format (example: *admin.acme*), followed by the password.

NOTE: The installer must get a successful directory connection in order to proceed. Consult your eDirectory LDAP documentation for troubleshooting.

- 7 When prompted for an ASAM System Container Context, specify the distinguished name (DN) of the container in which the organizational unit ASAM System should be created. The driver will store configuration and synchronization information in this container. Enter the DN in LDAP dot format. Example: *idm.acme*.
- 8 You are next prompted for information about the Core Driver, beginning with a descriptive name and the network port it will use (default 3451).
- 9 For the Core Driver network address, select a DNS name or IP address for the Core Driver. If your system has multiple addresses, use one which other systems (platforms) can use to communicate with the driver.

The installer will then create the eDirectory objects and indexes needed to completed the configuration.

- 10 Immediately after installation, you may need to change the port setting for the Core Driver's built-in remote loader. This is especially likely if you are also using the standard remote loader that comes with Identity Manager, since both versions of the remote loader use 8090 as their default port setting.

The port setting for the Core Driver's built-in remote loader resides in the `fanout.conf` file, which is located in `/usr/local/ASAM/data/`.

Edit the following line in `fanout.conf` to reflect the desired port:

```
-connection "ca=/usr/local/ASAM/keys/ca.pem port=8090"
```

NOTE: If you have a firewall, be sure to add the Driver's network port (default 3451) to its open ports list.

At the completion of this installation task, go next to [Section 5.2.3, "Setting Up the Core Driver in iManager,"](#) on page 64.

Upgrading a Local Core Driver Shim on Linux or Solaris

Upgrading a driver running in Local mode on your Linux or Solaris eDirectory server is the most common upgrade scenario. Once upgraded, it will run as a Remote Driver, with both the Driver object and Driver Shim integrated on the same physical host.

To upgrade a Driver Shim that is running in Local mode in Linux or Solaris:

- 1 Stop the the Core Driver object in iManager.
- 2 From your installation media, locate and execute the appropriate self-extracting installer:

```
sh linux_x86_coredriver.bin
sh linux_x86_64_coredriver.bin
sh solaris_sparc_coredriver.bin
```

NOTE: The x86 (32-bit) installer is compatible with both 32-bit and 64-bit versions of Linux. To use the x86_64 (64-bit) installer, you must first install and configure the 64-bit LDAP SDK (see ["Installing the LDAP Client SDK Libraries"](#) on page 52).

- 3 Accept the license, select the same installation directory as your previous installation (usually this is `/usr/local`) and proceed to install the product files.
- 4 The installer will next assist you in configuring the Driver Shim. You will be prompted for the Remote Loader password, which is used to encrypt driver network communications. Enter a password and remember it, as you will use it when configuring the driver in iManager.

- 5 You will then be prompted for a Driver Object password. This password will also be used when configuring the driver in iManager.
- 6 The next entry you are asked for is the eDirectory server/port, so the installer can retrieve an SSL certificate from eDirectory using your SSL-configured LDAP server. Enter the DNS name or IP address of the LDAP server that the Core Driver Shim will use to communicate with. Typically, this will be *localhost* on port *636*.
- 7 When prompted for an eDirectory admin name/password, enter the eDirectory administrator's ID in LDAP dot format (example: *admin.acme*), followed by the password.

NOTE: The installer must get a successful directory connection in order to proceed. Consult your eDirectory LDAP documentation for troubleshooting.

- 8 When prompted for an ASAM System Container Context, specify the distinguished name (DN) of the container in which the organizational unit ASAM System resides. You can find this information on the Provisioning Status Details page using the Fan-Out Driver Web application plug-in in iManager. Enter the DN in LDAP dot format. Example: *idm.acme*.
- 9 When prompted whether to create a new Driver or upgrade an existing one, enter *U* to upgrade an existing Driver.
- 10 When a list of drivers displays, enter the number corresponding to the Driver you plan to upgrade.

The installer will then generate an updated configuration file and install indexes needed to completed the configuration. (You may receive a warning message regarding indexes since they will already exist.)

- 11 Immediately after installation, you may need to change the port setting for the Core Driver's built-in remote loader. This is especially likely if you are also using the standard remote loader that comes with Identity Manager, since both versions of the remote loader use 8090 as their default port setting.

The port setting for the Core Driver's built-in remote loader resides in the `fanout.conf` file, which is located in `/usr/local/ASAM/data/`.

Edit the following line in `fanout.conf` to reflect the desired port:

```
-connection "ca=/usr/local/ASAM/keys/ca.pem port=8090"
```

NOTE: If you have a firewall, be sure to add the Driver's network port (default 3451) to its open ports list.

At the completion of this installation task, go next to [Section 5.2.3, "Setting Up the Core Driver in iManager," on page 64](#), which includes the task for upgrading a Core Driver configuration

Upgrading a Remote Core Driver Shim on Linux or Solaris

If you are running a Fan-Out Driver in Remote mode, you can upgrade the Driver Shim on the Linux or Solaris system and eliminate the need to run the Novell Java Remote Loader.

To upgrade a Driver Shim that is running in Remote mode in Linux or Solaris:

- 1 Stop the the Core Driver object and the Remote Loader instance for the current driver in iManager.

- 2 Still in iManager, open the *Identity Manager Remote Loader Console*, select the Fan-Out Driver instance and click *Edit*. Make a note of the *Connection Port*, *Trace Level* and *Trace File* fields.
- 3 Because you no longer need the standard Novell Remote Loader, you may disable or remove it as follows, depending on whether you have other Remote Drivers:
 - ♦ If you're running other Remote Drivers on the system, simply remove the Fan-Out Driver instance by selecting it in the Remote Loader Console and clicking *Remove*.
 - ♦ If you aren't running other Remote Drivers on the system, open *Control Panel* and run | *Add or Remove Programs (Programs and Features* on Windows Server 2008) to remove the Novell Identity Manager Connected System program.

- 4 From your installation media, locate and execute the appropriate self-extracting installer:

```
sh linux_x86_coredriver.bin
sh linux_x86_64_coredriver.bin
sh solaris_sparc_coredriver.bin
```

NOTE: The x86 (32-bit) installer is compatible with both 32-bit and 64-bit versions of Linux. To use the x86_64 (64-bit) installer, you must first install and configure the 64-bit LDAP SDK (see [“Installing the LDAP Client SDK Libraries”](#) on page 52).

- 5 Accept the license, select the same installation directory as your previous installation (usually this is `/user/local`) and proceed to install the product files.
- 6 The installer will next assist you in configuring the Driver Shim. You will be prompted for the Remote Loader password, which is used to encrypt driver network communications. Enter a password and remember it, as you will use it when configuring the driver in iManager.
- 7 You will then be prompted for a Driver Object password. This password will also be used when configuring the driver in iManager.
- 8 The next entry you are asked for is the eDirectory server/port, so the installer can retrieve an SSL certificate from eDirectory using your SSL-configured LDAP server. Enter the DNS name or IP address of the LDAP server that the Core Driver Shim will use to communicate with. Typically, this will be *localhost* on port *636*.
- 9 When prompted for an eDirectory admin name/password, enter the eDirectory administrator's ID in LDAP dot format (example: *admin.acme*), followed by the password.

NOTE: The installer must get a successful directory connection in order to proceed. Consult your eDirectory LDAP documentation for troubleshooting.

- 10 When prompted for an ASAM System Container Context, specify the distinguished name (DN) of the container in which the organizational unit ASAM System resides. You can find this information on the Provisioning Status Details page using the Fan-Out Driver Web application plug-in in iManager. Enter the DN in LDAP dot format. Example: *idm.acme*.
- 11 When prompted whether to create a new Driver or upgrade an existing one, enter *U* to upgrade an existing Driver.
- 12 When a list of drivers displays, enter the number corresponding to the Driver you plan to upgrade.

The installer will then generate an updated configuration file and install indexes needed to complete the configuration. (You may receive a warning message regarding indexes since they will already exist.)

- 13** Immediately after installation, you may need to change the port setting for the Core Driver's built-in remote loader. This is especially likely if you are also using the standard remote loader that comes with Identity Manager, since both versions of the remote loader use 8090 as their default port setting.

The port setting for the Core Driver's built-in remote loader resides in the `fanout.conf` file, which is located in `/usr/local/ASAM/data/`.

Edit the following line in `fanout.conf` to reflect the desired port:

```
-connection "ca=/usr/local/ASAM/keys/ca.pem port=8090"
```

NOTE: If you have a firewall, be sure to add the Driver's network port (default 3451) to its open ports list.

At the completion of this installation task, go next to [Section 5.2.3, "Setting Up the Core Driver in iManager," on page 64](#), which includes the task for upgrading a Core Driver configuration.

5.2.2 Installing the Driver Shim on Windows Systems

Core Driver installation on Windows begins with one of the following tasks, depending on your scenario:

- ♦ ["Installing the Core Driver Shim on Windows Systems" on page 60](#)
- ♦ ["Upgrading a Local Core Driver Shim on Windows Systems" on page 61](#)
- ♦ ["Upgrading a Remote Core Driver Shim on Windows Systems" on page 62](#)

Installing the Core Driver Shim on Windows Systems

To install a Driver Shim on a Windows System:

- 1** From your installation media, run the following command:

```
fan-out\IDMCoreDrivers\Win\win_x86_coredriver.exe
```

This x86 (32-bit) executable is compatible with both x86 and x64 versions of Windows.
- 2** Accept the license, select your installation directory and proceed to install the product files.
- 3** The installer will next assist you in configuring the Driver Shim. You will be prompted for the Remote Loader password, which is used to encrypt driver network communications. Enter a password and remember it, as you will use it when configuring the driver in iManager.
- 4** You will then be prompted for a Driver Object password. This is used to access the driver object in eDirectory. This password will also be used when configuring the driver in iManager.
- 5** The next entry you are asked for is the eDirectory server/port, so the installer can retrieve an SSL certificate from eDirectory using LDAP. Enter the DNS name or IP address of an eDirectory server, and the LDAP secure port (default 636). In the console window that appears, enter `y` to accept the certificate.

NOTE: If you are running both eDirectory and Active Directory on the Windows server, you may need to change the LDAP ports of either eDirectory or Active Directory so that they do not interfere with each other. See your product documentation for more information.

- 6 When prompted for an eDirectory admin name/password, enter the eDirectory administrator's ID in LDAP dot format (example: *admin.acme*), followed by the password.

NOTE: The installer must get a successful directory connection in order to proceed. Consult your eDirectory LDAP documentation for troubleshooting.

- 7 When prompted for an ASAM System Container Context, specify the distinguished name (DN) of the container in which the organizational unit ASAM System should be created. The driver will store synchronization information in this container. This is usually the top-level organization in the tree. Example: *acme*.
- 8 You are next prompted for information about the Core Driver, beginning with a descriptive name and the network port it will use (default 3451).
- 9 For the Core Driver network address, select a DNS name or IP address for the Core Driver. If your system has multiple addresses, use one which other systems (platforms) can use to communicate with the driver.

The installer will then create the eDirectory objects and indexes needed to completed the configuration.

- 10 Immediately after installation, you may need to change the port setting for the Core Driver's built-in remote loader. This is especially likely if you are also using the standard remote loader that comes with Identity Manager, since both versions of the remote loader use 8090 as their default port setting.

The port setting for the Core Driver's built-in remote loader resides in the `fanout.conf` file, which is located in `C:\Novell\ASAM\data\`.

Edit the following line in `fanout.conf` to reflect the desired port:

```
-connection "ca=/usr/local/ASAM/keys/ca.pem port=8090"
```

NOTE: If you have a firewall, be sure to add the Driver's network port (default 3451) to its open ports list.

At the completion of this installation task, go next to [Section 5.2.3, "Setting Up the Core Driver in iManager,"](#) on page 64.

Upgrading a Local Core Driver Shim on Windows Systems

You can upgrade a driver running in Local mode on your Windows eDirectory server. The upgraded Driver will run as a Remote Driver, with both the Driver objects and Driver Shim on the same system.

To upgrade a Driver Shim that is running in Local mode in Windows:

- 1 Stop the the Core Driver object in iManager.
- 2 From your installation media, run the following command:

```
fan-out\IDMCoreDrivers\Win\win_x86_coredriver.exe
```

This x86 (32-bit) executable is compatible with both x86 and x64 versions of Windows.
- 3 Accept the license, select the same installation directory as your previous installation (usually `C:\Novell\ASAM`) and proceed to install the product files.
- 4 The installer will next assist you in configuring the Driver Shim. You will be prompted for the Remote Loader password, which is used to encrypt driver network communications. Enter a password and remember it, as you will use it when configuring the driver in iManager.

- 5 You will then be prompted for a Driver Object password. This password will also be used when configuring the driver in iManager.
- 6 The next entry you are asked for is the eDirectory server/port, so the installer can retrieve an SSL certificate from eDirectory using LDAP. Enter the DNS name or IP address of an eDirectory server, and the LDAP secure port (default 636). In the console window that appears, enter *y* to accept the certificate.

NOTE: If you are running both eDirectory and Active Directory on the Windows server, you may need to change the LDAP ports of either eDirectory or Active Directory so that they do not interfere with each other. See your product documentation for more information.

- 7 When prompted for an eDirectory admin name/password, enter the eDirectory administrator's ID in LDAP dot format (example: *admin.acme*), followed by the password.

NOTE: The installer must get a successful directory connection in order to proceed. Consult your eDirectory LDAP documentation for troubleshooting.

- 8 When prompted for an ASAM System Container Context, specify the distinguished name (DN) of the container in which the organizational unit ASAM System resides. This is usually the top-level organization in the tree. Enter the DN in LDAP dot format. Example: *idm.acme*.
- 9 When prompted whether to create a new Driver or upgrade an existing one, click *No* to upgrade an existing Driver.
- 10 When a list of drivers displays, select the Driver associated with the system you are upgrading. The installer will then generate an updated configuration file and install indexes needed to completed the configuration. (You may receive a warning message regarding indexes since they will already exist.)
- 11 Immediately after installation, you may need to change the port setting for the Core Driver's built-in remote loader. This is especially likely if you are also using the standard remote loader that comes with Identity Manager, since both versions of the remote loader use 8090 as their default port setting.

The port setting for the Core Driver's built-in remote loader resides in the `fanout.conf` file, which is located in `C:\Novell\ASAM\data\`.

Edit the following line in `fanout.conf` to reflect the desired port:

```
-connection "ca=/usr/local/ASAM/keys/ca.pem port=8090"
```

NOTE: If you have a firewall, be sure to add the Driver's network port (default 3451) to its open ports list.

At the completion of this installation task, go next to [Section 5.2.3, "Setting Up the Core Driver in iManager," on page 64](#), which includes the task for upgrading a Core Driver configuration

Upgrading a Remote Core Driver Shim on Windows Systems

If you're running a Fan-Out Driver in Remote mode, you can upgrade the Driver Shim on the Windows system running the Connected System portion of the Driver.

To upgrade a Driver Shim that is running in Remote mode in Windows:

- 1 Stop the the Core Driver object and the Remote Loader instance for the current driver in iManager.

- 2 Still in iManager, open the *Identity Manager Remote Loader Console*, select the Fan-Out Driver instance and click *Edit*. Make a note of the *Connection Port*, *Trace Level* and *Trace File* fields.
- 3 Because you no longer need the standard Novell Remote Loader, you may disable or remove it as follows, depending on whether you have other Remote Drivers:
 - ♦ If you're running other Remote Drivers on the system, simply remove the Fan-Out Driver instance by selecting it in the Remote Loader Console and clicking *Remove*.
 - ♦ If you aren't running other Remote Drivers on the system, open *Control Panel* and run | *Add or Remove Programs (Programs and Features* on Windows Server 2008) to remove the Novell Identity Manager Connected System program.
- 4 From your installation media, run the following command:

```
fan-out\IDMCoreDrivers\Win\win_x86_coredriver.exe
```

This x86 (32-bit) executable is compatible with both x86 and x64 versions of Windows.
- 5 Accept the license, select the same installation directory as your previous installation (usually C:\Novell\ASAM) and proceed to install the product files.
- 6 The installer will next assist you in configuring the Driver Shim. You will be prompted for the Remote Loader password, which is used to encrypt driver network communications. Enter a password and remember it, as you will use it when configuring the driver in iManager.
- 7 You will then be prompted for a Driver Object password. This password will also be used when configuring the driver in iManager.
- 8 The next entry you are asked for is the eDirectory server/port, so the installer can retrieve an SSL certificate from eDirectory using LDAP. Enter the DNS name or IP address of an eDirectory server, and the LDAP secure port (default 636). In the console window that appears, enter *y* to accept the certificate.

NOTE: If you are running both eDirectory and Active Directory on the Windows server, you may need to change the LDAP ports of either eDirectory or Active Directory so that they do not interfere with each other. See your product documentation for more information.

- 9 When prompted for an eDirectory admin name/password, enter the eDirectory administrator's ID in LDAP dot format (example: *admin.acme*), followed by the password.
-
- NOTE:** The installer must get a successful directory connection in order to proceed. Consult your eDirectory LDAP documentation for troubleshooting.
-
- 10 When prompted for an ASAM System Container Context, specify the distinguished name (DN) of the container in which the organizational unit ASAM System resides. This is usually the top-level organization in the tree. Enter the DN in LDAP dot format. Example: *idm.acme*.
 - 11 When prompted whether to create a new Driver or upgrade an existing one, click *No* to upgrade an existing Driver.
 - 12 When a list of drivers displays, select the Driver associated with the system you are upgrading. The installer will then generate an updated configuration file and install indexes needed to completed the configuration. (You may receive a warning message regarding indexes since they will already exist.)
 - 13 Immediately after installation, you may need to change the port setting for the Core Driver's built-in remote loader. This is especially likely if you are also using the standard remote loader that comes with Identity Manager, since both versions of the remote loader use 8090 as their default port setting.

The port setting for the Core Driver's built-in remote loader resides in the `fanout.conf` file, which is located in `C:\Novell\ASAM\data\`.

Edit the following line in `fanout.conf` to reflect the desired port:

```
-connection "ca=/usr/local/ASAM/keys/ca.pem port=8090"
```

NOTE: If you have a firewall, be sure to add the Driver's network port (default 3451) to its open ports list.

At the completion of this installation task, go next to [Section 5.2.3, "Setting Up the Core Driver in iManager," on page 64](#), which includes the task for upgrading a Core Driver configuration.

5.2.3 Setting Up the Core Driver in iManager

You will use the Fan-Out Driver's Web application to complete the Core Driver installation. This application resides in recent versions of iManager as a standard plug-in. If your version of iManager does not include the plug-in, you can install it from the software that comes with the Core Driver.

NOTE: In addition to iManager, you can use Designer, an application interface that comes with Identity Manager, for setting up and modelling large deployments of the Fan-Out Driver. A Fan-Out Driver application plug-in is included as part of the Designer installation. For more information on using Designer, see [Section 6.4, "Applications For Configuration," on page 75](#).

After you have installed or upgraded a Driver Shim, the installation process continues in iManager with the following tasks, depending on your scenario:

- ♦ ["Importing a Configuration for a Newly Installed Core Driver" on page 64](#)
- ♦ ["Upgrading a Core Driver Configuration" on page 65](#)

If your version of iManager does not include the plug-in or if you are not familiar with iManager, you can refer to two additional topics at the end of this section before starting:

- ♦ ["Installing the iManager Plug-In \(if Necessary\)" on page 66](#)
- ♦ ["Using the iManager Interface" on page 66](#)

Importing a Configuration for a Newly Installed Core Driver

Use iManager to configure a Core Driver in the Identity Vault (eDirectory). To import a Core Driver configuration:

- 1 Login to iManager for your tree and select the *Import Configuration* task under *Identity Manager Utilities* on the left.
- 2 Keep the Driver Set selection and, if this is a new Driver Set, select the server in eDirectory where the Driver will run.
- 3 From the *Configurations* menu, select `Fan-Out-IDM3_6_0-V1.xml`. If this file is not available, select *Import a configuration from the client* and select the file `rules\Fan-Out-IDM3_6_0-V1.xml` under the directory where the Driver Shim is installed (`C:\Novell\ASAM` by default).
- 4 Enter the following configuration fields. The installer will have filled in some of these fields:
 - ♦ *Driver Name*: Enter a descriptive name.

- ♦ *Activation Group*: Choose the selection that corresponds to the activation you purchased. The Driver will operate in evaluation mode for 90 days if you don't have an activation.
 - ♦ *LDAP Host and Port*: Enter the DNS name or IP address and TCP port of your LDAP host.
 - ♦ *Remote Host Name and Port*: Enter the DNS name or IP address and TCP port used by the system where the Driver Shim runs.
 - ♦ *ASAM Master User/Password*: Enter an LDAP account that will be used to manage Driver information.
 - ♦ *Driver Object Password/Remote Loader Password*: Enter the passwords you entered when installing the Driver Shim.
- 5 Click *Define Security Equivalences* and add your ASAM Master User.
 - 6 Click *Exclude Administrative Roles* and add the admin user, the ASAM Master User and other high-privilege users to the *Excluded Users* list.
 - 7 Click *Finish* to complete the import.

Upgrading a Core Driver Configuration

If you upgrade a Core Driver Shim, you also need to upgrade its configuration in iManager. To upgrade a configuration:

- 1 Login to iManager for your tree and select the *Import Configuration* task under *Identity Manager Utilities* on the left.
- 2 Select the Driver Set that contains the Driver to be upgraded.
- 3 Click *Next* to keep the Driver Set selection.
- 4 From the *Configurations* menu, select `Fan-Out-IDM3_6_0-V1.xml`. If this file is not available, select *Import a configuration from the client* and select the file `rules\Fan-Out-IDM3_6_0-V1.xml` under the directory where the Driver Shim is installed (C:\Novell\ASAM by default).
- 5 On the next page, to the right of the *Driver Name* field, select the driver you wish to upgrade from the *Existing Drivers* drop-down box.
- 6 Enter the following configuration fields, consistent with your current installation. The installer will have filled in some of these fields:
 - ♦ *Activation Group*: Choose the selection that corresponds to the activation you purchased. The Driver will operate in evaluation mode for 90 days if you don't have an activation.
 - ♦ *LDAP Host and Port*: Enter the DNS name or IP address and secure TCP port of your LDAP host.
 - ♦ *Remote Host Name and Port*: Enter the DNS name or IP address and TCP port used by the system where the Driver Shim runs.
 - ♦ *ASAM Master User/Password*: Enter an LDAP account that will be used to manage Driver information.
 - ♦ *Driver Object Password/Remote Loader Password*: Enter the passwords you entered when installing the Driver Shim.
- 7 On the next page, select *Update everything about that driver and policy libraries* and click *Next*.

- 8 Click *Finish* to complete the import.
- 9 If necessary, apply any manual customizations.

NOTE: You must install the new version of the iManager Plug-in before using the Driver. See [“Installing the iManager Plug-In \(if Necessary\)” on page 66.](#)

Installing the iManager Plug-In (if Necessary)

If your installation of iManager does not display the Fan-Out Driver Configuration role (Roles and Tasks menu on the left), you can install the iManager plug-in manually.

To install the iManager plug-in:

- 1 Login to iManager as an administrative user.
- 2 Click the *Configure* icon at the top.
- 3 Click *Available Novell Plug-in Modules* under *Plug-in Installation* on the left menu.
- 4 Click *Add* above the list of plug-ins.
- 5 Select `fan-out\iManagerPlugIn\FanOutWeb.npm` from your installation media and click *OK*.
- 6 Check the box next to *Novell Identity Manager - Fan-Out Driver Plug-in* and click *Install* above the list of plug-ins.
- 7 Restart the Tomcat or Tomcat5 service on your iManager system, and exit and log back into iManager.
- 8 If the Fan-Out Driver Configuration role has not appeared, continue with the following steps.
- 9 Click the *Configure* icon at the top.
- 10 Click *RBS Configuration* under *Role Based Services* on the left menu.
- 11 Click the number under the *Not-Installed* column in the table.
- 12 Check the box next to *FanOutWeb* and click *Install* above the list.
- 13 Click the *Roles and Tasks* icon at the top.

With the plug-in now installed, you can proceed to your next task.

NOTE: For additional information about using iManager with the Fan-Out Driver application plug-in, see [Section 6.4, “Applications For Configuration,” on page 75.](#)

Using the iManager Interface

To use the iManager interface for setting up a Core Driver:

- 1 In iManager, select the *Configure iManager Plug-In* task under *Fan-Out Driver Configuration*.
- 2 Enter the DNS name or IP address and port of the system running the Driver Shim and click *Apply*.
- 3 You may now use any of the items under *Fan-Out Driver Configuration* and *Fan-Out Driver Utilities* in iManager.

5.2.4 Other Tasks Following Installation

After the initial installation or upgrade of a Core Driver, other tasks that you may need to perform from time to time include the following:

- ♦ “Starting the Core Driver” on page 67
- ♦ “Stopping the Core Driver” on page 67
- ♦ “Reconfiguring the Driver Shim” on page 67
- ♦ “Installing Secondary Drivers” on page 68
- ♦ “Installing a New Primary Driver” on page 69

Starting the Core Driver

Both the Identity Manager Driver object and Driver Shim service must be running for the Core Driver to operate. To start the Core Driver:

- 1 In iManager, select the *Identity Manager Overview* task under *Identity Manager*.
- 2 Select the Driver Set where the Driver is installed.
- 3 Click the status indicator (stop line) in the upper right corner of the driver icon, then click *Start Driver*.
- 4 On the Driver Shim system, start the Fan-Out Driver as follows:
 - ♦ If the system is running Linux/UNIX, enter the following command:
`/etc/init.d/asamcdrvd start`
 - ♦ If the system is running Windows, open *Control Panel* and run *Administrative Tools > Services*. Start the Novell IDM Fan-Out Driver service.

Stopping the Core Driver

To stop the Core Driver:

- 1 On the Driver Shim system, stop the Fan-Out Driver as follows:
 - ♦ If the system is running Linux/UNIX, enter the following command:
`/etc/init.d/asamcdrvd stop`
 - ♦ If the system is running Windows, open *Control Panel* and run *Administrative Tools > Services*. Stop the Novell IDM Fan-Out Driver service.
- 2 In iManager, stop the Driver in the Driver Set Overview.

Reconfiguring the Driver Shim

The Driver Shim can be reconfigured in a number of areas, as itemized below.

NOTE: Always be sure to stop the Driver Shim before starting any of these reconfiguration tasks as described in [“Stopping the Core Driver” on page 67](#).

- ◆ To re-run the installer’s configuration wizard in Windows, open *Control Panel* and run *Add or Remove Programs (Programs and Features on Windows Server 2008)*. Click the *Change* button under *Novell IDM Fan-Out Core Driver*. Then select *Modify or Repair* from the dialog box that opens.

You can use the installer to create a new installation, create a new driver or upgrade an existing driver. Note that the installer doesn't remember previously configured fields, so you'll have to enter the fields like you did on the first-time install.

- ◆ To change the Remote Loader and/or Driver Object passwords:
 - ◆ If the system is running Linux or Solaris, enter the following command and select menu item 1:
`/user/local/ASAM/setup/fandrv-config`
 - ◆ If the system is running Windows, execute the following command from the installation directory (C:\Novell\ASAM by default):
`bin\CoreDriver\asamdrv.exe -sp`
- ◆ To retrieve a new SSL certificate from eDirectory:
 - ◆ If the system is running Linux or Solaris, enter the following command and select menu item 2:
`/user/local/ASAM/setup/fandrv-config`
 - ◆ If the system is running Windows, execute the following command from the installation directory (C:\Novell\ASAM by default):
`bin\CoreDriver\asamdrv.exe -s`
- ◆ To install or remove the Driver Shim service in Windows, execute the following command from the installation directory (C:\Novell\ASAM by default) using either the `installService` or `removeService` parameter:
`bin\CoreDriver\asamdrv.exe -parameter`

Installing Secondary Drivers

For scaleability, you can install secondary drivers to handle platform synchronization and password requests. A system can run only one Driver Shim. See [Section 5.4, “Performance Tuning,” on page 69](#) for more information on specifying secondary drivers.

To install a secondary driver.

- 1 On the secondary Driver Shim’s system, follow steps 1-7 under one of the following tasks, depending on your operating environment:

NOTE: In step 7, be sure to specify the container that holds the ASAM System Container.

- ◆ [“Installing a New Core Driver Shim on Linux or Solaris” on page 56](#)
 - ◆ [“Installing the Driver Shim on Windows Systems” on page 60](#)
- 2 You will be prompted to create a new Driver or upgrade a Driver. Click *Yes* to create a new Driver.
 - 3 Specify a distinct name for the Driver as well as its port.

- 4 Select the Driver's network address.
- 5 When you are prompted whether to make the new Driver primary, click *No* to keep the Driver secondary.
- 6 Once the Driver Shim is installed, import a new Driver following the steps in “[Importing a Configuration for a Newly Installed Core Driver](#)” on page 64. Be sure to use the XML configuration file generated for the secondary Driver.

You can now run the secondary Driver as you would the primary Driver.

Installing a New Primary Driver

Depending on your configuration needs, you may decide to install a Driver Shim for a new primary driver.

NOTE: If the Core Driver you wish to make primary is already installed, you can use the *Configure Core Drivers* menu task in iManager to do this.

- 1 Stop all Identity Manager Driver objects and Driver Shims. See “[Stopping the Core Driver](#)” on page 67.
- 2 Follow all the steps in the task, “[Installing Secondary Drivers](#)” on page 68, with the following exception:
In step 5, click *Yes* to make the driver primary.
- 3 Start all Identity Manager Driver objects and Driver Shims. See “[Starting the Core Driver](#)” on page 67.

5.3 Activating the Driver After Evaluation

Identity Manager and Identity Manager drivers must be activated within 90 days of installation, or they shut down. You can activate Identity Manager products to a fully licensed state at any time.

To activate Identity Manager products:

- 1 Purchase the appropriate licenses.
- 2 Generate a Product Activation Request.
- 3 Submit the Product Activation Request to Novell.
- 4 Install the Product Activation Credential received from Novell.

For detailed information about completing these steps, see *Activating Identity Manager Products* in the *Identity Manager Administration Guide*, available at [the Novell Identity Manager 3.6 documentation site \(http://www.novell.com/documentation/idm36\)](http://www.novell.com/documentation/idm36).

5.4 Performance Tuning

The Fan-Out Driver provides a unique Identity Management solution by extending its services to many clients both simultaneously and among mixed environments. This section describes best practices for ensuring optimal performance in the Core Driver Shim's ability to deliver this functionality.

5.4.1 Secondary Drivers

Adding secondary Drivers can provide your Fan-Out platform clients with failover and load balancing. If you have multiple eDirectory servers and plan to deploy the Fan-Out Platform Services to many different systems, it's recommended that you install and deploy the Fan-Out driver to more than one server.

5.4.2 Platform Operation Modes

The Fan-Out Platform Receiver, the component that connects to the Core Driver to receive events from Identity Manager, can be configured to run in five different operation modes (see [Section 8.8.1, "Modes of Operation," on page 109](#)). Three of these modes, in particular, may have an impact on the performance of the Core Driver Shim, as described in [Table 5-1 on page 70](#).

Table 5-1 *Effect of Operation Modes on Core Driver performance.*

Mode	Function	Effect on Performance
Persistent	The Platform Receiver connects and remains connected to the Fan-Out Driver to receive events in real-time, which is desirable if it is necessary for your platform to receive events as they occur in the Identity Vault.	Maintaining the open connection and its resources does carry an overhead in both memory and CPU usage for the Driver Shim.
Polling	The Platform Receiver connects to the Fan-Out Driver on configurable intervals to catch up on events since its last poll.	<p>Positive: Can allow the Driver Shim to release its connection and free up resources. Because each and every event will not be delivered to the Platform immediately, this mode will deliver a single event with all of the needed provisioning information to the Polling platform, making the delivery more efficient.</p> <p>Negative: Can also cause delayed event delivery and, depending on the polling interval, you may see the same memory issues if you have too many platforms connecting too frequently.</p>
Scheduled	Very similar to Polling mode with one exception: it only runs once. Allows the system to decide when to launch the Platform Receiver using another facility, such as a cron.	Can provide your systems with nightly or weekly updates and also allow you to stagger the event deliveries, which safeguards against overloading your Fan-Out Driver with too many requests at once.

Configuring and Administering the Core Driver

6

After you have installed the Core Driver of the Novell® Identity Manager Fan-Out Driver, use the information in this section for further configuration and administration.

Topics include

- ♦ [Section 6.1, “Configuration Overview,” on page 71](#)
- ♦ [Section 6.2, “Driver System Security Overview,” on page 71](#)
- ♦ [Section 6.3, “Administration Overview,” on page 73](#)
- ♦ [Section 6.4, “Applications For Configuration,” on page 75](#)
- ♦ [Section 6.5, “Management Tasks,” on page 78](#)

6.1 Configuration Overview

Before beginning, remember that the Fan-Out Driver includes two principal parts: the Core Driver and Platform Services. Information in this section focuses on the Core Driver’s configuration, and additional configuration will need to be performed on each platform.

6.1.1 Core Driver Configuration

Core Driver configuration information is maintained in the Driver object and in objects in the ASAM System container. The Core Driver installation process creates the initial configuration.

You use iManager to maintain the configuration information.

- ♦ For information about managing the Driver object configuration parameters, see [“Driver Object Configuration Parameters” on page 82](#).
- ♦ For information about managing the objects in the ASAM System container, see [Section 6.4, “Applications For Configuration,” on page 75](#) and [Section 6.5, “Management Tasks,” on page 78](#).

6.1.2 Platform Services Configuration

The Core Driver maintains configuration objects that represent each target platform for its own use in the ASAM System container.

Target platforms each obtain local configuration information from their respective platform configuration file. For more information about the platform configuration file, see [Part III, “Platform Services Planning,” on page 103](#).

6.2 Driver System Security Overview

System security is maintained through connection certificates between driver components and password-protected access to objects in Novell eDirectory™.

6.2.1 Connection Security

The connections between Core Driver components and between Event Journal Services and Platform Receivers use Secure Sockets Layer (SSL). Some types of the Platform Services Process use SSL for their connections to Authentication Services, and others use DES encryption. SSL connections are authenticated through the use of certificates.

The certificates used by the Identity Manager Fan-Out Driver are minted by the Certificate Services component of the Core Driver. When you install and configure a new component, you obtain a certificate.

Because platforms cannot examine the configuration objects for the Core Driver in the ASAM System container, Core Driver network address information is included in their certificates. This enables platforms to verify Core Driver component identity. If the network configuration of a Core Driver is changed, a new certificate must be minted.

The Core Driver certificate is minted when you start the Core Driver for the first time. When you update network address information for a Core Driver, a new certificate is automatically minted for it. You must restart a Core Driver after changing its network address information in order for the new certificate to take effect.

Obtain a new certificate for a platform by starting the Platform Receiver with the appropriate command line parameter. For details, see [Part IV, “Platform Services Administration,” on page 139](#).

Identity Manager Fan-Out Driver components store their security certificates and related information in their certs directory. Ensure that access to the certs directory is restricted to the driver system itself and to its administrators.

- ♦ **Core Driver:** `asam\data\coredriver\certs`
- ♦ **Platform Services:** `asam\data\platformservices\certs`

6.2.2 ASAM Master User Security

The Core Driver performs an LDAP bind as the ASAM Master User to gain access to eDirectory. You must not place restrictions on the ASAM Master User object that would interfere with its use by the driver. Set maximum password length for the ASAM Master User to at least 32 characters. Disable intruder detection for the ASAM Master User object so that it cannot be disabled by someone without the appropriate rights.

The ASAM Master User must have Supervisor rights to the container in eDirectory that holds the users and groups that can be added to the Census. This is known as the User and Group Subtree. These rights are granted during installation.

To use the AS Client API to access objects outside of the User and Group subtree, you must grant additional rights to the ASAM Master User.

- ♦ You must grant the ASAM Master User Browse object rights and Compare property rights to any object that is accessed through the AS Client API.
- ♦ You must grant the ASAM Master User Read property rights to any object whose Security Equals list or Group Membership list, or other attribute value is accessed through the AS Client API.

Because the ASAM Master User is granted significant rights, you must ensure that its password remains secure.

The Core Driver obtains the password of the ASAM Master User from the Driver object. If your security practices prescribe periodic password changes, you can create a second User object to be used as an alternate ASAM Master User. Then you can swap back and forth between these User objects when it is necessary to change the password.

Creating an Alternate ASAM Master User Object

- 1 Use iManager to create a new User object. We recommend that you use the same directory context as the original ASAM Master User object.
- 2 Use iManager to assign the new User object Security Equivalence to the original ASAM Master User object.

Now you have two User objects with the necessary rights to act as the ASAM Master User.

Changing the Password and Updating the Configuration

The following procedure assumes you have created a second ASAM Master User object as described in the preceding section. It assumes one object is named ASAM1 and the other is named ASAM2. We also assume that ASAM1 is in use and that it is the one listed in the driver configuration parameters.

To change the ASAM Master User object to use a new password:

- 1 Use iManager to set the new password for ASAM2.
- 2 Update the Driver object for each Core Driver, specifying *ASAM2* for the Authentication ID, and the new password for the Application Password.
 - 2a In iManager, select *Identity Manager Management > Overview*.
 - 2b Locate the driver in its driver set.
 - 2c Click the driver status indicator in the upper right corner of the driver icon, then click *Edit Properties*.
 - 2d Click *Identity Manager > Driver Configuration*. *Authentication ID* and *Application Password* are located under the *Authentication* heading.
- 3 Use iManager to change the password of ASAM1 to an undisclosed randomly chosen value.

ASAM2 is now the ASAM Master User, using a new password. The old password (of the ASAM1 user) can no longer be used.

NOTE: The Core Driver does an LDAP bind as the ASAM Master User upon startup. There is no need to restart the driver now. It will use ASAM2 the next time it is started.

6.3 Administration Overview

You use the Web interface for most Core Driver configuration and administration tasks. For details about using the Web interface, see [Section 6.4, “Applications For Configuration,” on page 75](#).

The ongoing tasks of administering the driver can be grouped into the following categories.

- ♦ Monitoring the operation of the Core Drivers
- ♦ Monitoring the operation of Platform Services

- ♦ Maintaining the Census
- ♦ Reviewing your overall system management plan and making changes to the driver as appropriate

6.3.1 Monitoring Core Drivers

You can use the Fan-Out Driver Web application to view component status. For additional information, see [“Displaying Component Status” on page 94](#).

From time to time, depending on the size of the organization and the amount of activity, a system administrator should review the logs written by Core Driver components in order to check the health of the system. For example, you might want to investigate the cause of a large number of denied SSL connection requests. For more information about viewing logs, see [“Viewing Logs” on page 94](#).

The Fan-Out Driver also generates its own messages for purposes of monitoring and troubleshooting. These messages, which are documented in [Appendix D, “Messages,” on page 219](#), can also be redirected into your own custom programs and status documents to free your administrators from manual Fan-Out log-tracking.

NOTE: For more about the configuration parameter for using this capability, see [“Publish Fan-Out Log Messages” on page 83](#).

6.3.2 Monitoring Platform Services

It is a good idea to monitor the logs written by Platform Services. For details about these logs, see the administration guide for your platform operating system.

6.3.3 Maintaining the Census

Administrative personnel should periodically use the Web interface to monitor naming exceptions and inactive users and groups. The frequency at which this is done is a function of the size of the organization being managed, the rate at which changes take place, and the rules in place within the organization concerning unique user IDs in eDirectory. If the organizational structure is appropriate, the same people who manage User objects also maintain the Census.

The actions taken depend on the policies of the individual organization. Some lines follow. For a review of the concepts, see the [Part I, “Concepts and Facilities,” on page 13](#).

Naming Exceptions

A naming exception results if a new User object or a new Group object is encountered with the same name as an Enterprise User object or Enterprise Group object that is already in the Census. In an organization with a policy of unique usernames, this is generally the result of a mistake when adding a new user or group. In this case, the name of the new user or group should be changed to a unique name.

It is also possible that the new user or group was inadvertently added to the Census as a result of a mistake in changing the Census parameters so that the driver is now looking in unintended places for users or groups. In this case, correct the Census parameters.

It is also possible that a departmental administrator is attempting to breach security by taking the user ID of a previously existing user.

Inactive Users and Groups

An inactive user or group is an Enterprise User or Enterprise Group whose corresponding User object or Group object has been deleted from the directory or is no longer covered by a Census Search object. Deletion of a user might be the legitimate result of someone leaving the organization. If this is true, the entry should be removed from the Census.

It is possible that the user or group has been inadvertently omitted from the Census as a result of a mistake in changing the Census parameters. If this is the case, correct the Census parameters.

You can use the Web interface to set Census parameters to automatically remove inactive users and groups from the Census if this is appropriate for your organization. For details, see [“Specifying Automatic Removal of Inactive Users and Groups” on page 81](#).

Enterprise User objects in the Census relate to eDirectory User objects using a globally unique identifier (GUID). Identity Provisioning uses the GUID to prevent the reuse of a User object name from resulting in inappropriate access to the old user's accounts on platform systems. You must ensure that the deleted ID has been appropriately removed from all target platforms not managed by Identity Provisioning.

A user that is inactive or is not present in the Census, but does exist in eDirectory, is able to log in to eDirectory directly, but is not able to authenticate through the driver where the Enterprise User ID is required (such as is the case with z/OS or UNIX).

A user that is present in the Census but is not present in eDirectory is not able to authenticate through the driver.

6.4 Applications For Configuration

Novell provides two application interfaces to support the installation and configuration of the Fan-Out driver:

- ♦ iManager, which is a live Web interface for real-time administration
- ♦ Designer, which is a planning tool for deployment modeling, testing and implementation

A Fan-Out Driver application plug-in exists for each of these interfaces.

In recent versions of iManager, the Fan-Out Driver plug-in comes installed as a standard feature. If you have an older version of iManager, you can use a copy of the plug-in that comes with your Fan-Out Driver software. To add it to iManager, see [“Installing the iManager Plug-In \(if Necessary\)” on page 66](#).

In Designer, the plug-in is included as part of the Designer installation.

See the following topics for more about these two interfaces:

- ♦ [Section 6.4.1, “Using iManager With the Fan-Out Driver Plug-In,” on page 76](#)
- ♦ [Section 6.4.2, “Using Designer With the Fan-Out Driver Plug-In,” on page 77](#)

6.4.1 Using iManager With the Fan-Out Driver Plug-In

For detailed information about using iManager, refer to the *iManager Administration Guide* for your version of the product at the [Novell Documentation site \(http://www.novell.com/documentation\)](http://www.novell.com/documentation).

You configure and administer the Identity Manager Fan-Out Driver using iManager Roles and Tasks. The left side of the display lists actions you can take. Information pertaining to the action you select is displayed on the right side.

Rights Required for Web Application Use

To use iManager and the Fan-Out Driver Web application, a user must have greater than normal user rights as shown in the following table.

Table 6-1 *Web Interface User Rights*

Function	Rights
Log in and perform basic functions	Read object rights to the ASAM System container
Configure objects	Read and Create object rights to the ASAM System container
Start a Trawl	Read and Create object rights to the ASAM System container

Accessing the Web Application

To access the Web application, log in to iManager, click the Roles and Tasks icon at the top of the iManager screen, and click the desired Fan-Out Driver Configuration or Fan-Out Driver Utilities task.

Logging Out

To log out of the Web application, click the exit door icon at the top of the iManager screen.

Obtaining Additional Information

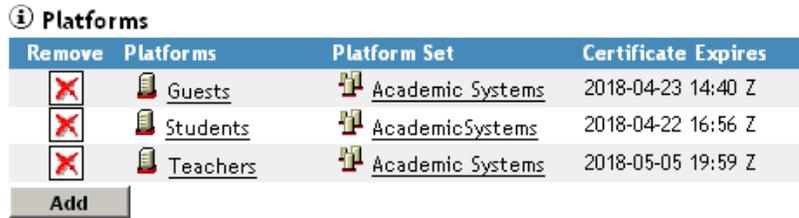
Figure 6-1 *Additional Information Icon*



Additional information is available for the items and procedures in the Web application. To display this information, click the Additional Information icon.

Maintaining Lists

Figure 6-2 List of Items



Remove	Platforms	Platform Set	Certificate Expires
	 Guests	 Academic Systems	2018-04-23 14:40 Z
	 Students	 AcademicSystems	2018-04-22 16:56 Z
	 Teachers	 Academic Systems	2018-05-05 19:59 Z

Add

Many items in the Fan-Out Driver Web application are grouped into lists.

To add an item to a list, click the *Add* button.

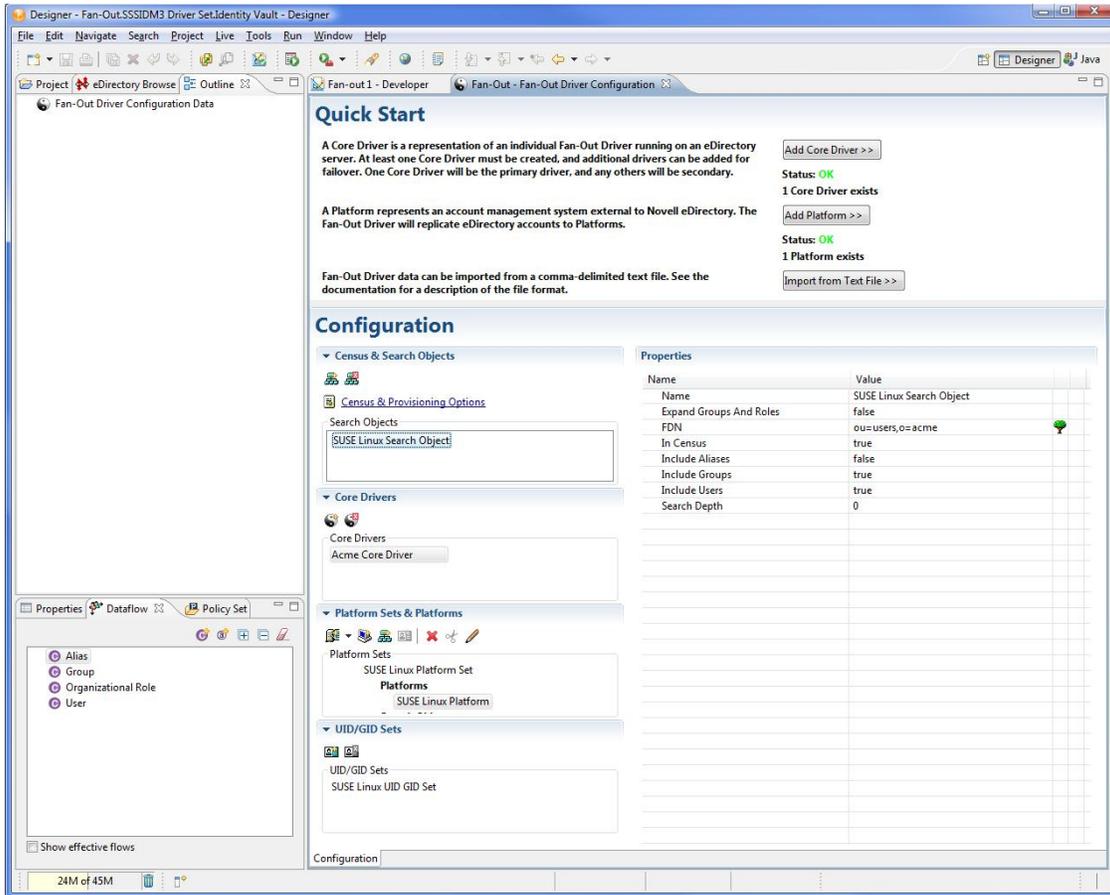
To view or change the attributes of an item, click its name in the list.

To remove an item from a list, click the *Remove* button for that item. A confirmation page is displayed. Click *Yes* to confirm removal, or click your Web browser's *Back* button to abort.

6.4.2 Using Designer With the Fan-Out Driver Plug-In

Novell Designer is a graphical user interface tool that allows you to model and deploy Identity Manager installations. Designer includes a Fan-Out Driver plug-in that enables you to configure Fan-Out data (such as Search Objects, Platforms, and Platform Sets) before deploying the Driver to eDirectory. With Designer you can also perform mass imports much more efficiently than with iManager.

Figure 6-3 Identity Manager Designer Interface with Fan-Out Driver Plug-in.



Getting Started

Refer to the [Identity Manager 3.6 Documentation site \(http://www.novell.com/documentation/idm36\)](http://www.novell.com/documentation/idm36) for detailed information about using Designer.

To get started in using the Fan-Out Driver plug-in in Designer:

- 1 Create an Identity Vault and Driver Set.
- 2 Drag-and-drop a *Fan-Out Application* from the *Tools* section of the Palette from the right.
- 3 Select the Fan-Out Driver configuration file and create the Driver.
- 4 Right-click the Driver line and select *Edit Fan-Out Driver*.

The plug-in application will open. Consult the Fan-Out Driver Plug-In section of the Designer online help for more information on modeling and deploying your Fan-Out Driver installation.

6.5 Management Tasks

This section describes tasks you can complete in iManager, using the Fan-Out Driver Web Interface plug-in, to manage the Driver:

- ♦ [Section 6.5.1, “Configuring the Census,” on page 79](#)
- ♦ [Section 6.5.2, “Configuring Core Drivers,” on page 82](#)

- ◆ Section 6.5.3, “Configuring the iManager Plug-In,” on page 87
- ◆ Section 6.5.4, “Configuring Logs,” on page 87
- ◆ Section 6.5.5, “Configuring Platform Sets,” on page 87
- ◆ Section 6.5.6, “Configuring Platforms,” on page 89
- ◆ Section 6.5.7, “Configuring Provisioning,” on page 91
- ◆ Section 6.5.8, “Configuring Search Objects,” on page 92
- ◆ Section 6.5.9, “Configuring Linux/UNIX UID/GID Sets,” on page 93
- ◆ Section 6.5.10, “Displaying Component Status,” on page 94
- ◆ Section 6.5.11, “Viewing Driver Documentation,” on page 94
- ◆ Section 6.5.12, “Viewing Logs,” on page 94
- ◆ Section 6.5.13, “Displaying Provisioning Details,” on page 96
- ◆ Section 6.5.14, “Reviewing Naming Exceptions,” on page 96
- ◆ Section 6.5.15, “Reviewing Platform Errors,” on page 97
- ◆ Section 6.5.16, “Managing Trawls,” on page 97

6.5.1 Configuring the Census

Configuring the Census includes the following tasks:

- ◆ “Specifying Search Objects” on page 80
- ◆ “Specifying Trawl Times” on page 81
- ◆ “Specifying Automatic Removal of Inactive Users and Groups” on page 81
- ◆ “Delaying Password Expiration Until Midnight” on page 81
- ◆ “Specifying a Platform Object Delete Pending Duration” on page 82

Before You Begin a Core Driver’s First Trawl

The installation of the Core Driver creates indexes for specific attributes of objects in the Identity Vault. For installation into a very large tree, these indexes take some time to bring online. Before you begin your first Trawl, verify that the indexes are in the online state. For the attributes indexed by the Core Driver installation, see the list that follows this procedure.

To view the Server object indexes and their state:

1. In iManager, select *eDirectory Maintenance > Index Management*.
2. Select the Server object for the Core Driver.

GUID	Object_Class
ASAM_inputGUID	ASAM_eGroupMembers
ASAM_eGroupMembership	ASAM_inputReference
ASAM_platformAssociation	ASAM_platformAssociation_SS
ASAM_aliases	ASAM-NetAddressList
ASAM_eventsUpTo	ASAM_deletesUpTo

ASAM_deletePendingsUpTo	ASAM_passwordsUpTo
ASAM_eventError_SS	ASAM_platformSetAssociation_SS
ASAM_UIDGIDAssociation_SS	ou
Country	Locality
Organization	Tree_Root
State	

Specifying Search Objects

Search objects specify how users and groups are selected from eDirectory to be included in the Census. For details about Search objects, see [“Configuring Search Objects” on page 92](#).

To update the Census after you make Search object changes, start a Trawl. For details about starting a Trawl, see [“Starting a Census Trawl” on page 97](#).

To add a new Census Search object:

- 1 Click *Fan-Out Driver Configuration > Configure Census*. The *Census Configuration* page is displayed.
- 2 Click *Search Objects > Add*. The *Add a Search Object* page is displayed.
- 3 Specify the Search object distinguished name and attributes as desired, then click *Apply*.
For details about Search object attributes, see [“Search Object Attributes” on page 92](#).

To change a Census Search object:

- 1 Click *Fan-Out Driver Configuration > Configure Census*. The *Census Configuration* page is displayed.
- 2 In the list of Search objects, click the name of the Search object to modify. The *Modify Search Object* page is displayed.
- 3 Update the attributes of the Search object as desired, then click *Apply*.
For details about Search object attributes, see [“Search Object Attributes” on page 92](#).

To remove a Census Search object:

- 1 Click *Fan-Out Driver Configuration > Configure Census*. The *Census Configuration* page is displayed.
- 2 In the list of Search objects, click the name of the Search object to be deleted. The *Modify Search Object* page is displayed.
- 3 In the list of Platform Sets under *Platform Set Associations*, click each *Remove* button. The *Remove Search Object* confirmation page is displayed each time you click a *Remove* button. Click *Yes* for each.
- 4 Under the *In Census* heading, click the *Remove* button. The *Remove Search Object* confirmation page is displayed. Click *Yes*.

Specifying Trawl Times

Object Services is notified by the Event Subsystem of events in eDirectory that affect the Census. Object Services also periodically verifies the consistency of the Census by examining objects in the directory in a procedure known as a Trawl. Use the Web interface to specify the times when a Trawl runs.

- 1 Click *Fan-Out Driver Configuration > Configure Census*. The *Census Configuration* page is displayed.
- 2 Trawl times are listed (24-hour clock) under *Trawl Time Configuration*. If no times are listed, Object Services does not automatically start any Trawls.

Time of day values used by the driver are specified in Universal Time, formerly known as GMT, and commonly abbreviated as Z.

To add a new Trawl time, click *Add*.

To remove a Trawl time from the list, click its *Remove* button.

Specifying Automatic Removal of Inactive Users and Groups

You can choose to have Enterprise Users and Enterprise Groups whose corresponding User object or Group object is deleted from eDirectory or no longer covered by a Census Search object remain in the Census in an inactive state. This prevents another person from receiving access to resources as an unintended result of the reuse of a user name. Inactive users cannot authenticate through Authentication Services.

You can also specify that inactive users and groups be removed from the Census automatically during a Trawl after they have reached a given number of inactive days.

To specify inactive user and group options:

- 1 Click *Fan-Out Driver Configuration > Configure Census*. The *Census Configuration* page is displayed.
- 2 Inactive user and group options are listed on the *Census Configuration* page under *Inactive Enterprise User and Group Actions*. Specify the action you want, then click *Apply*.

To view inactive users and groups, use the *Provisioning Details* utility and specify *Search Type > Inactive Users and Groups*. For more information about using the Provisioning Details utility, see [“Displaying Provisioning Details” on page 96](#).

Delaying Password Expiration Until Midnight

You can choose to delay the expiration of user passwords by Authentication Services from the exact date and time set for them in eDirectory until the end of the day (local time of the Core Driver host server) on which they expire. This can result in smoother operation for users on platforms with third-party systems that cache and reuse passwords during the day.

To specify password expiration options:

- 1 Click *Fan-Out Driver Configuration > Configure Census*. The *Census Configuration* page is displayed.
- 2 Password expiration options are listed on the *Census Configuration* page under *Delay User Password Expiration*. Select the option you prefer, then click *Apply*.

Specifying a Platform Object Delete Pending Duration

You can use the Web interface to specify a Delete Pending Duration. During this interval, User and Group objects associated with a platform that have either been deleted from eDirectory or are no longer covered by a Search object, are not deleted from their corresponding platforms. The results of a Delete User or Delete Group Receiver script can be difficult to reverse. This provides a grace period to allow recovery from a mistake affecting many users.

The User Delete Pending or Group Delete Pending script is called when a delete event becomes pending for a user or group, but the Delete User or Delete Group script is not called until the Delete Pending Duration expires.

To specify when users and groups are deleted from platforms:

- 1 Click *Fan-Out Driver Configuration > Configure Census*. The *Census Configuration* page is displayed.
- 2 Deletion options are listed under *Platform Object Delete Pending Duration*. Select the option you prefer, then click *Apply*.

6.5.2 Configuring Core Drivers

Core Drivers provide the Web interface, perform Census maintenance functions, and provide Authentication Services and Identity Provisioning to platforms.

Starting a Core Driver

- 1 In iManager, select *Identity Manager Management > Overview*.
- 2 Locate the driver in its driver set.
- 3 Click the driver status indicator in the upper right corner of the driver icon, then click *Start Driver*.

Stopping a Core Driver

- 1 In iManager, select *Identity Manager Management > Overview*.
- 2 Locate the driver in its driver set.
- 3 Click the driver status indicator in the upper right corner of the driver icon, then click *Stop Driver*.

Driver Object Configuration Parameters

The Core Driver uses Driver object configuration parameters to identify the ASAM System container, the ASAM Master User object, an LDAP Services for eDirectory host server, and to obtain other related information. The Driver object is created during Core Driver installation.

To view and modify Driver object configuration parameters:

- 1 In iManager, select *Identity Manager Management > Overview*.
- 2 Locate the driver in its driver set.
- 3 Click the driver status indicator in the upper right corner of the driver icon, then click *Edit Properties*.

- 4 Click *Identity Manager > Driver Configuration*. Driver configuration parameters are located under the *Driver Settings* heading.
- 5 Update the settings as desired. Then click *OK* or *Apply*. To end without saving any changes, click *Cancel*.

Core Driver Config DN

Displays the name of this Driver object.

LDAP Host and Port

Specifies the IP address or DNS name and the TCP port of the LDAP Services for eDirectory host server that the Core Driver components use to access the ASAM System container. The LDAP host server must hold a writable replica of the ASAM System container.

The default is port 636 on the local host. For best performance, use the local host.

ASAM System Container

Specifies the fully distinguished name of the ASAM System container. The ASAM System container holds system configuration and operational objects.

Activation Group

Displays the Identity Manager integration modules that you have activated.

Publish Fan-Out Log Messages

Enables/disables redirection of Fan-Out messages into your own custom programs and status documents to free your administrators from manual Fan-Out log-tracking.

Locale

Specifies the two-character ISO 639 language identifier for the language to be used by the Core Driver. The default value of Locale is en (English)

Lower Password Case

Specifies whether Event Journal Services changes password case when sending password replication information to Platform Receivers.

Password replication information is provided to the Core Driver from many different sources. Maintaining password case can be undesirable because some sources of password information present passwords in uppercase.

By default, Event Journal Services converts passwords to lowercase before sending password replication events to Platform Receivers.

Migration Mode Password

Specifies the special password that is used with Password Migration on the z/OS operating system. Users with this password and with Login Disabled set are in the migration state. For more information about Password Migration, see the *Identity Manager Fan-Out Driver for Mainframes Administration Guide*.

Change Password Exit Library

Specifies the file path for the optional Password Change Validation Exit library. For information about the Password Change Validation Exit, see [Appendix B, “Password Change Validation Exit,” on page 213](#).

Change Password Exit Function

Specifies the function name for the optional Password Change Validation Exit exported in the library identified by the Change Password Exit Library parameter. For information about the Password Change Validation Exit, see [Appendix B, “Password Change Validation Exit,” on page 213](#).

External Password Sources

Specifies a comma-separated list of names of additional trees from which the Core Driver accepts password change information.

If no *External Password Source* is specified, the Core Driver rejects password change information from trees other than the tree the Core Driver is installed in.

Password change information is provided to the Core Driver by Identity Manager Fan-Out Driver password intercepts, and is used to provide password information to password replication platforms.

Verify serial number of incoming platform connection

Enables/disables whether Core Driver checks the platform’s certificate serial number against the serial number listed in the Core Driver configuration. This is a useful security measure to detect and reject certificates that may have been compromised.

Network Connect Timeout

Specifies timeout in seconds for the Core Driver to use when opening a network connection to another network system.

Network Read Timeout

Specifies timeout in seconds for the Core Driver to use when reading data from a network connection. Higher timeout values can prevent premature disconnects.

Network Write Timeout

Specifies timeout in seconds for the Core Driver to use when writing data to a network connection. Higher timeout values can prevent premature disconnects.

Core Driver System Configuration Object Attributes

Descriptions for each attribute follow.

Network Address

The Core Driver configuration must list all of the network addresses of the Core Driver's host server. Network address information for the host server is entered when the Core Driver is installed. You must update this information if the host server network address is changed or if an additional network interface is installed in the server.

One network address is designated as the default. Identity Manager Fan-Out Driver Core Driver components use the default address to connect to each other.

The platform configuration file used by a Platform Services component specifies the network address of each Core Driver that is used by that component. If you change the network address of a Core Driver that is specified in a platform configuration file, you must update that platform configuration file. For details about the platform configuration file, see [Part III, “Platform Services Planning,” on page 103](#).

If you change the network address configuration of a Core Driver, a new certificate is automatically minted for the Core Driver.

IMPORTANT: You must restart the Core Driver for the new certificate to take effect.

Core Driver Port

The TCP port number used by the Core Driver defaults to 3451. You can change the Core Driver TCP port number if necessary.

If you change a Core Driver TCP port number, you must also make the corresponding changes to each platform configuration file that references the Core Driver.

Authentication Services z/OS and NDS-AS Compatibility Port

The TCP port number used by the Core Driver to communicate with Platform Services for z/OS and with NDS[®] Authentication Services (NDS-AS) version 3 Clients. The default is 2000.

Cache Size and Time to Live

Authentication Services maintains an encrypted cache of recent successful authentication requests to provide better performance for applications, such as Web servers, that make large bursts of requests to authenticate the same user in a short period of time.

You can specify the amount of memory that is allocated for this cache and the maximum length of time an entry is to be kept in the cache.

Primary Core Driver

One Core Driver is designated as the primary Core Driver. Other Core Drivers are known as secondary Core Drivers. The primary Core Driver serves the Web interface and provides environmental information during the installation process for other Core Drivers. Only the primary Core Driver listens for events from eDirectory and performs Trawls.

Designating the Primary Core Driver

- 1 In the Web interface, click *Fan-Out Driver Configuration > Configure Core Drivers*. The *Core Driver Configuration* page is displayed.
- 2 Click *Set as Primary*.
- 3 Click *Yes* to confirm.
- 4 Restart the previous and new Core Drivers. For details about this procedure, see [“Stopping a Core Driver” on page 82](#) and [“Starting a Core Driver” on page 82](#).
- 5 Configure the iManager plug-in to use the new primary Core Driver. For details, see [Section 6.5.3, “Configuring the iManager Plug-In,” on page 87](#).

Before changing which Core Driver is the primary Core Driver, ensure that the proposed new primary Core Driver holds replicas of all objects covered by Census Search objects.

Adding a Core Driver

For step-by-step instructions to add a Core Driver, see [Chapter 5, “Installing the Core Driver,”](#) on page 51.

Changing the Core Driver Configuration

- 1 In the Web interface, click *Fan-Out Driver Configuration > Configure Core Drivers*. The *Core Driver Configuration* page is displayed.
- 2 Click the name of the Core Driver whose configuration you want to change. The *Modify Core Driver* page is displayed.
- 3 Specify attributes for the Core Driver as appropriate.

Removing a Core Driver

- 1 Remove the Core Driver from the platform configuration file of all platforms where it is present. For information about the platform configuration file, see the [Part III, “Platform Services Planning,”](#) on page 103.
- 2 Stop the Core Driver.
For details, see [“Stopping a Core Driver”](#) on page 82.
- 3 Uninstall the Core Driver software and related files from the Core Driver host.
 - ♦ If the host server operating system is Linux/UNIX, delete the ASAM directory from the file system.
 - ♦ If the host server operating system is Windows, use *Windows Control Panel > Add/Remove Programs*.
- 4 Remove the Driver object from Identity Manager.
 - 4a In iManager, select *Identity Manager Management > Overview*.
 - 4b Locate the driver set for the driver, then click *Delete Driver*.
 - 4c Select the Core Driver from the list and confirm its deletion.
- 5 In the Web interface, click *Fan-Out Driver Configuration > Configure Core Drivers*. The *Core Driver Configuration* page is displayed.
- 6 Click the *Remove* button for the Core Driver to be removed. The *Remove Core Driver* confirmation page is displayed. Click *Yes* to confirm.

Maintaining Logs Used by the Core Driver

Audit Services writes operational and audit log messages for the Core Driver to the `asam\data\coredriver\logs` directory.

You can use the Web interface to view logs and to configure how messages are managed. For information about viewing the logs, see [“Viewing Logs”](#) on page 94. For details about configuring the logs, see [Section 6.5.4, “Configuring Logs,”](#) on page 87.

6.5.3 Configuring the iManager Plug-In

Each administrative user must configure the iManager plug-in to use the primary Core Driver.

- 1 In the Web interface, click *Fan-Out Driver Configuration > Configure iManager Plug-In*. The *Configure iManager Plug-In* page is displayed.
- 2 Specify the DNS name or IP address of the primary Core Driver host server.
- 3 Specify the TCP port number for the primary Core Driver. The default is 3451.
- 4 Click *Apply*.

6.5.4 Configuring Logs

Audit Services maintains the Operational Log and Audit Log files written by the Core Driver. You can use the Web interface to manage log files. You can choose to have log messages kept for a given number of days, or you can choose to have log messages kept permanently. You can also specify the components whose messages are written to the logs.

- 1 Click *Fan-Out Driver Configuration > Configure Logs*. The *Log Configuration* page is displayed.
- 2 Select the option that you want for log retention.
- 3 Select the components whose messages you want included in the logs.
- 4 Click *Apply*.

You can use the Web interface to view log messages. For more information, see [“Viewing Logs” on page 94](#).

The Log Configuration page is also used to configure debugging logging. For more information, see [Section 13.1, “Obtaining Debugging Output,” on page 149](#).

6.5.5 Configuring Platform Sets

A Platform Set contains one or more Platform objects that share the same users and groups.

When you add a new Platform Set, you give it a name and associate a UID/GID Set with it. You may also specify an alternate naming attribute, which identifies a name to be given to each user or group on platforms within the Platform Set. Then you add Search objects that describe what users and groups are provisioned to the platforms that belong to the Platform Set.

You may specify an Alternate Naming Attribute. When a user or group is provisioned to a Platform within this Platform Set, the Alternate Naming Attribute indicates the name that will be used.

After you have defined a Platform Set, you can create the Platform objects that represent its target platforms. For information about creating Platform objects, see [“Configuring Platforms” on page 89](#).

The Platform Set object's user and group population is described by one or more Search objects. For details about Search objects, see [“Configuring Search Objects” on page 92](#).

Platform Set Attributes

Descriptions for each attribute follow.

UID/GID Set Association

When you create a Platform Set, you specify a UID/GID Set that is used to assign UID numbers and GID numbers to Linux/UNIX platforms that are members of the Platform Set. You cannot change the UID/GID Set assigned to Platform Set after the Platform Set has been created.

Leave this option empty to use the `posixAccount` and `posixGroup` `uidNumber` and `gidNumber` attributes.

Alternate Naming Attribute

By default the name given to a user or group on each platform is the CN. By using the Alternate Naming Attribute, you also can indicate a name associated with each platform within the Platform Set. If you use this extra attribute in eDirectory, then each user or group must include a value for it.

The content of an attribute that is designated as an Alternate Naming Attribute should be either a single value or multiple values of the form `<platformset>:<name>`.

The actual entry you make on the *Modify Platform Set* window should reflect the attribute name used by LDAP. You can find this information under the LDAP group object for the server, which includes the mapping between eDirectory and LDAP attribute names.

Search Objects

Search Objects designate the users and groups from the Census that are used to populate the platforms that are members of the Platform Set. For information about Search objects, see [“Configuring Search Objects” on page 92](#).

Platforms

Upon creation, each Platform object is associated with exactly one Platform Set.

Adding a Platform Set

- 1 In the Web interface, click *Fan-Out Driver Configuration > Configure Platform Sets*. The *Platform Set* page is displayed.
- 2 Click *Add*. The *New Platform Set* page is displayed.
- 3 Specify an Alternate Naming Attribute if one should be used.
- 4 Type a name for the new Platform Set, select the UID/GID Set that is to be used by the new Platform Set, then click *Apply*. The *Modify Platform Set* page is displayed.
- 5 Add one or more Search objects to describe the user and group population for the Platform Set. Click *Search Objects > Manage*.

For details about Search objects, see [“Configuring Search Objects” on page 92](#).

- 6 Add one or more Platform objects to describe the target platforms that constitute the Platform Set. Click *Platforms > Add* to create a new Platform object and add it to the Platform Set.

For details about adding Platform objects, see [“Configuring Platforms” on page 89](#).

Changing Platform Set Attributes

- 1 In the Web interface, click *Fan-Out Driver Configuration > Configure Platform Sets*. The *Platform Set* page is displayed.

- 2 In the list of Platform Sets, click the name of the Platform Set to modify. The *Modify Platform Set* page is displayed.
- 3 Update the attributes of the Platform Set as desired.

Removing a Platform Set

- 1 Remove all platforms associated with the Platform Set.
For information about removing a platform, see [“Removing a Platform” on page 91](#).
- 2 In the Web interface, click *Fan-Out Driver Configuration > Configure Platform Sets*. The *Platform Set* page is displayed.
- 3 Click the *Remove* button of the Platform Set you want to remove. The *Remove Platform Set* confirmation page is displayed. Click *Yes* to confirm.

6.5.6 Configuring Platforms

A Platform object contains the configuration information the Core Driver uses to serve a platform for Authentication Services and Identity Provisioning. Additional configuration of Platform Services is performed on the platform. For detailed information about configuring and administering Platform Services, see the [Part III, “Platform Services Planning,” on page 103](#).

Authentication Mode

For platforms that may restrict password length or case sensitivity (mainframes, for example), the Authentication Mode can be used to allow case-sensitive, shorter passwords. This mode allows you to continue to use and enforce complex passwords in eDirectory while providing an authentication method for systems that do not or cannot adhere to the same password policies.

- ◆ Check Passwords
 - ◆ Select *Case Insensitive* if you want the Core Driver to check passwords without considering case.
 - ◆ Select *Case Sensitive* to enforce case sensitive passwords.
- ◆ Check only the first number of characters
Enter an integer, greater than or equal to 0, to indicate how many characters should be checked in the correct password. Examples:
 - ◆ If you select 8, the Core Driver will only check the first 8 characters of the password for validity.
 - ◆ Indicating 0 disables the entire Authentication Mode feature.

Platform Attributes

Descriptions for each attribute follow.

Platform Set

Each platform is a member of exactly one Platform Set. The Platform Set is used to associate users and groups with its member platforms. You specify the Platform Set that a platform belongs to when you create the Platform object. The Platform Set a platform belongs to cannot be changed after the Platform object is created.

Permit Password Replication

You can specify whether or not requests from a platform for password replication information are honored. Enable this only for those platforms that need, and are trusted with, password information from eDirectory.

No: No password information is provided to the platform.

Yes: Password information is provided to the platform. No events for an account are sent to the platform unless password information for the account is available to the driver.

If Available: Password information is provided to the platform when it is available. Events for an account are sent to the platform even if no password information is available for the account. This setting can result in accounts being unprotected if it is used without password redirection.

After you enable password replication for a platform, you must restart the Platform Receiver if it is running in Persistent Mode or Polling Mode.

In order for password replication information to be available to a platform, the appropriate password change intercepts must be installed and correctly configured on all systems that can change passwords in eDirectory. For more information, see [“Password Replication Requirements” on page 44](#) and the [Part I, “Concepts and Facilities,” on page 13](#).

Platform Network Address

The DNS name or IP address of the platform system. If the platform system has more than one network interface, list all of the network addresses.

DES Key

Information about the DES key that is used to encrypt communications with platforms that use the DES interface is stored in the Platform Configuration object. The platform configuration file of platforms that use the DES interface must contain the same DES key as the Platform Configuration object or communication attempts fail.

When you change the DES key, the previous key is saved in the Platform Configuration object. You can specify a time interval during which communications using the old key are accepted from the platform system. Specify an interval that gives you enough time to update the platform configuration file with the new DES key.

Adding a New Platform

- 1 In the Web interface, click *Fan-Out Driver Configuration > Configure Platforms*. The *Platform Configuration* page is displayed.
- 2 Click *Add*. The *New Event Driven Platform* page is displayed.
- 3 Type a name for the platform, select the Platform Set the new platform is to join, then click *Apply*. The *Modify Platform* page is displayed.
- 4 Specify attributes for the platform as appropriate.
For details, see [“Platform Attributes” on page 89](#).
- 5 Install the platform distribution package on the target server. For details, see [Part IV, “Platform Services Administration,” on page 139](#) and the Quick Start for your platform operating system type.

Changing Platform Attributes

- 1 In the Web interface, click *Fan-Out Driver Configuration > Configure Platforms*. The *Platform Configuration* page is displayed.
- 2 In the list of platforms, click the name of the platform to modify. The *Modify Platform* page is displayed.
- 3 Update the attributes of the platform as desired. For details, see “[Platform Attributes](#)” on [page 89](#).

Removing a Platform

- 1 Remove the driver installation from the platform system. For details, see [Part IV, “Platform Services Administration,” on page 139](#).
- 2 In the Web interface, click *Fan-Out Driver Configuration > Configure Platforms*. The *Platform Configuration* page is displayed.
- 3 In the list of platforms, click the *Remove* button of the Platform object that you want to remove. The *Remove Platform* confirmation page is displayed. Click *Yes* to confirm.

6.5.7 Configuring Provisioning

Provisioning configuration involves three attributes.

Provisioning Configuration Attributes

Descriptions of each attribute follow.

Objects Excluded from Provisioning

You can specify that certain objects are globally excluded from Identity Provisioning by the Identity Manager Fan-Out Driver. You can list a fully distinguished LDAP object name or a simple common name. If you add an object that has already been provisioned to target platforms, the object is deleted from the target platforms.

Web Interface LDAP Time-Out

The time-out interval, in seconds, for LDAP operations initiated by the Web interface. If an LDAP request does not return within the time-out value, the operation fails.

Trawl and Provisioning LDAP Time-Out

The time-out interval, in seconds, for Core Driver LDAP operations. If an LDAP request by a Core Driver does not return within the time out-value, the operation fails.

Changing Provisioning Attributes

- 1 In the Web interface, click *Fan-Out Driver Configuration > Configure Provisioning*. The *Provisioning Configuration* page is displayed.
- 2 Modify provisioning attributes as desired.

6.5.8 Configuring Search Objects

Search objects determine the users and groups that are included in the Census and Platform Set populations.

Start a Trawl after you make Search object changes. For details about starting a Trawl, see [“Starting a Census Trawl” on page 97](#).

Search Object Types

Search objects can be any of the following:

- ♦ **User Objects:** Users who are Search objects are added to the Census.
- ♦ **Group Objects:** Groups that are Search objects are added to the Census. Members of groups that are Search objects are added to the Census.
- ♦ **Organizational Role Objects:** Occupants of Organizational Role objects that are Search objects are added to the Census.
- ♦ **Organization Objects and Organizational Unit Objects:** Container objects are scanned for users and groups to add to the Census.
- ♦ **Dynamic Group Objects:** Members of Dynamic Group objects that are Search objects are added to the Census.

The settings of the *Include Users* and *Include Groups* attributes described in the following section take precedence in determining which objects are added to the Census.

Search Object Attributes

Include Users: Determines if users covered by the Search object are added to the Census.

Include Groups: Determines if groups covered by the Search object are added to the Census.

Expand: Determines if users who are members of Group objects or occupants of Organizational Role objects found inside a container Search object are added to the Census.

Include Aliases: Determines if Alias objects covered by the Search object are added to the Census. The User or Group object that is represented by an Alias object is provisioned to platforms.

Depth: Determines how many steps down the eDirectory tree hierarchy the Core Driver looks beyond the container object for users and groups to add to the Census. A Depth of zero causes the search to stop at the Search object container.

In Census: Determines if users and groups covered by this Search object are included in the Census.

Platform Set Associations: Specifies which Platform Sets this Search object is used to populate.

Adding Search Objects

- 1 Click *Fan-Out Driver Configuration > Configure Search Objects*. The *Search Objects* page is displayed.
- 2 Click *Add*. The *Add a Search Object* page is displayed.
- 3 Specify the Search object distinguished name and attributes as desired, then click *Apply*.

For details about Search object attributes, see [“Search Object Attributes” on page 92](#).

Changing Search Object Attributes

- 1 Click *Fan-Out Driver Configuration > Configure Search Objects*. The *Search Objects* page is displayed.
- 2 In the list of Search objects, click the name of the Search object to modify. The *Modify Search Object* page is displayed.
- 3 Update the attributes of the Search object as desired, then click *Apply*.

For details about Search object attributes, see [“Search Object Attributes” on page 92](#).

Removing Search Objects

- 1 Click *Fan-Out Driver Configuration > Configure Search Objects*. The *Search Objects* page is displayed.
- 2 In the list of Search objects, click the name of the Search object to be deleted. The *Modify Search Object* page is displayed.
- 3 In the list of Platform Sets under *Platform Set Associations*, click each *Remove* button. The *Remove Search Object* confirmation page is displayed each time you click a *Remove* button. Click *Yes* for each.
- 4 Under the *In Census* heading, click the *Remove* button. The *Remove Search Object* confirmation page is displayed. Click *Yes*.

6.5.9 Configuring Linux/UNIX UID/GID Sets

A UID/GID Set contains entries for users and groups, together with their corresponding Linux/UNIX UID and GID numbers.

A UID/GID Set is associated with each Platform Set so that the UID and GID numbers of users and groups managed by driver are the same on all Linux/UNIX platforms within the Platform Set.

When a new user or group is added to a Platform Set, it receives the next available UID/GID number.

You can reserve a range of numbers for local use by the platform administrators. The driver does not assign UID or GID numbers within the reserved range.

Leave this option empty to use the `posixAccount` and `posixGroup` `uidNumber` and `gidNumber` attributes.

UID/GID Set Attributes

Descriptions of each attribute follow.

Highest Used UID/GID

The highest UID/GID number that has been assigned to a user or group.

Reserved UID/GID Range

Specifies a range of UID/GID numbers that the driver does not assign to users or groups.

Associated Platform Sets

The Platform Sets that use this UID/GID Set.

Adding a UID/GID Set

- 1 In the Web interface, click *Fan-Out Driver Configuration > Configure UID/GID Sets*. The *UID/GID Set Configuration* page is displayed.
- 2 Click *Add*. The *New UID/GID Set* page is displayed.
- 3 Type a name for the UID/GID Set configuration object in the ASAM System container.
- 4 Specify the lowest and highest numbers to be reserved for local system administrator use, then click *Apply*.

The value for these numbers cannot be changed after you create the UID/GID Set.

Viewing UID/GID Set Details

- 1 In the Web interface, click *Fan-Out Driver Configuration > Configure UID/GID Sets*. The *UID/GID Set Configuration* page is displayed.
- 2 In the list, click the name of the UID/GID Set you want to view. The *UID/GID Set Details* page is displayed.

Removing a UID/GID Set

- 1 Remove all Platform Sets associated with the UID/GID Set.
For information about removing a Platform Set, see [“Removing a Platform Set” on page 89](#).
- 2 In the Web interface, click *Fan-Out Driver Configuration > Configure UID/GID Sets*. The *UID/GID Set Configuration* page is displayed.
- 3 Click the *Remove* button of the UID/GID Set you want to remove. The *Remove UID/GID Set* confirmation page is displayed. Click *Yes*.

6.5.10 Displaying Component Status

To display a status overview of your Identity Manager Fan-Out Driver system, click *Fan-Out Driver Utilities > Component Status* in the Web interface.

To display status details for a component, click the component name on the *Status Overview* page.

6.5.11 Viewing Driver Documentation

You can use the Web interface to view documentation for the Identity Manager Fan-Out Driver. To view the documentation, click *Fan-Out Driver Utilities > Documentation*.

6.5.12 Viewing Logs

You can use the Web interface to view log files. You can also use Audit to view log information.

Starting the Log Viewer

To start the Log Viewer, click *Fan-Out Driver Utilities > Log Viewer*. The *Component Log Viewer* page is displayed.

Controlling the Display

You can control the display of log files by setting the values of the following fields. Click *Update Criteria* after you have specified new values.

- ♦ **Component:** The component whose log you want to view.
- ♦ **Log Type:** The type of log for the selected component that you want to view. This can be the Operational Log or the Audit Log.
- ♦ **Lines to Return:** The number of lines to be returned at one time. This value determines the size of a set of lines used in scrolling operations, such as Next and Find Next.
- ♦ **Date:** The date of the log information that you want to display. Type this date in yyyy-mm-dd format. For example, specify June 26, 2004 as 2004-06-26.
- ♦ **Time:** The time of the first log record to display. Type the time in hh:mm:ss (24-hour clock) format. Seconds are optional. For example, specify 1:30 p.m. as 13:30.
- ♦ **Find:** A search string. The first set of lines containing the Find String, written after the time specified, is displayed. All lines containing the find string are marked with an icon so that they can be easily identified as you scroll through the log.

Obtaining an Explanation for a Message

To view the documentation for a given message, click its Message ID in the log display. An explanation for the message is displayed.

Navigating through a Log

Navigation links are provided for the following functions:

- ♦ **Beginning of Day:** Scrolls the log display to the beginning of the day specified by Date.
- ♦ **End of Day:** Scrolls the log display to the end of the day specified by Date.
- ♦ **Most Current:** Scrolls the log display to most current set of lines.
- ♦ **Previous:** Scrolls the display to the previous set of lines. Scrolling stops at the beginning of the day specified by Date.
- ♦ **Next:** Scrolls the display to the next set of lines. Scrolling stops at the end of the day specified by Date.
- ♦ **Find Previous:** Scrolls the display to the previous set of lines that include the find string. Scrolling stops at the beginning of the day specified by Date.
- ♦ **Find Next:** Scrolls the display to the next set of lines that include the find string. Scrolling stops at the end of the day specified by Date.

6.5.13 Displaying Provisioning Details

You can use the Web interface to search for and display objects in the Census and in the user and group population of a Platform Set.

- 1 Click *Fan-Out Driver Utilities > Provisioning Details*. The *Provisioning Details* page is displayed.
- 2 Select the Search Type you want.
- 3 Type the Search String.
Objects whose name begins with the string you type are matched. If you leave this field blank, all objects are matched.
- 4 Select the maximum number of results to be returned.
- 5 Click *Search*.
Objects matching the search string are returned, up to the maximum count that you specified.
- 6 Click the name of an object in the results list to view its attributes. The *Object Details* page is displayed for that object.

6.5.14 Reviewing Naming Exceptions

Naming exceptions are produced when a new User or Group object covered by a Census Search object is found with the same name as an Enterprise User or Enterprise Group object that is already present in the Census.

To review naming exceptions in the Web interface, click *Fan-Out Driver Utilities > Review Naming Exceptions*.

Resolving Naming Exceptions

- 1 In the Web interface, click *Fan-Out Driver Utilities > Review Naming Exceptions*.
- 2 Use iManager or a similar utility to change the name of the User or Group object that is causing the conflict.
- 3 To remove the record of the naming exception, click the Remove button for the exception in the list on the *Review Naming Exceptions* page of the Web interface.
Trawl processing automatically removes naming exceptions from the list after you have resolved them.

Excluding Recurring Exceptions

If you have recurring exceptions that are normal for you, you can permanently exclude them from the Census and the exception list.

- 1 In the Web interface, click *Fan-Out Driver Utilities > Review Naming Exceptions*.
- 2 In the desired row, click the button for the action you want to take.
 - ♦ To exclude all users and groups with this common name, click *Exclude Name*.

If the user or group already exists in the Census and on platforms, the platforms receive a Delete Pending event and, after the Delete Pending Duration, a Delete event. The object is not provisioned to the platform again.

- ♦ To exclude this specific user or group, click *Exclude DN*.

6.5.15 Reviewing Platform Errors

The Platform Receiver can return an error indication for its processing of a provisioning event. Such events remain pending for the platform, but are marked in an error state. You can use the Web interface to clear these events or to mark them to be sent to the Platform Receiver again.

- 1 In the Web interface, click *Fan-Out Driver Utilities > Review Platform Errors*. The *Review Platform Errors* page is displayed.
- 2 Click the name of the platform whose errors you want to handle. The *Errors on Platform* page is displayed.
- 3 Select the desired action for the events that are in error.

If additional events for a user or group associated with a platform are created (by a Trawl or by the Event Subsystem), pending events that are in error are cleared for that user or group.

6.5.16 Managing Trawls

Object Services examines portions of eDirectory specified by Census Search objects to build and maintain the Census. This process is known as a Trawl. Only the primary Core Driver performs Trawls. For information about setting the primary Core Driver, see [“Designating the Primary Core Driver” on page 85](#).

You can use the Web interface to display information about the last Trawl, to start a Trawl, and to stop a Trawl that is in progress.

Displaying Trawl Information

To display Trawl information, click *Fan-Out Driver Utilities > Trawl*.

Starting a Census Trawl

- 1 Click *Fan-Out Driver Utilities > Trawl*. The *Trawl Status* page is displayed.
- 2 Click *Start*.

For information about scheduling Trawls to run automatically, see [“Specifying Trawl Times” on page 81](#).

Stopping a Census Trawl

You can use the Web interface to stop a Trawl.

- 1 Click *Fan-Out Driver Utilities > Trawl*. The *Trawl Status* page is displayed.
- 2 Click *Stop*. It can take a few moments for the Trawl to stop.

Troubleshooting the Core Driver

7

Novell® Identity Manager Fan-Out Driver components record messages to their Audit Log, Operational Log, and their host system log. Examining these should be foremost in your troubleshooting efforts.

The Audit and Operational logs of Core Driver components are maintained in their logs directory.

By its very nature, the Fan-Out Driver is highly dependent upon the proper operation of your network and Novell eDirectory™. If you are having problems with the driver, ensure that the various driver components are able to communicate with one another, and that eDirectory is functioning properly.

For information pertaining to performance issues, see [Section 4.2, “Configuration and Performance Guidelines,” on page 41](#).

IMPORTANT: Make sure that you upgrade the Identity Manager Fan-Out Driver, including all of your platforms, when new versions or support packs become available.

7.1 Obtaining Debugging Output

Identity Manager Fan-Out Driver components support the option to produce extensive debugging output. Although this output is intended primarily for use by Novell Technical Support, you might find it useful for your own troubleshooting efforts.

Because debugging mode adversely affects performance, it should not be used for routine operations.

7.1.1 Debugging the Core Driver

To obtain debugging output for the Core Driver:

- 1 Specify the destination for debugging information by setting the Debug Log File and Debug to DSTrace configuration parameters as desired.

For more information, see [“Driver Object Configuration Parameters” on page 82](#).

- 2 Specify the debugging information to be produced.

2a In iManager, click *Fan-Out Driver Configuration > Configure Logs*. The *Log Configuration* page is displayed.

2b Select the Core Driver components you want debugging information for.

2c Click *Apply*.

7.2 Troubleshooting Core Driver Configuration Issues

Problems with Core Driver configuration can result in improper driver operation. The messages logged by Identity Manager Fan-Out Driver components can lead you to the source of such problems.

7.2.1 Rights Issues

Ensure that rights are properly set to enable the driver to perform its functions.

- ♦ For the rights required by the ASAM Master User object, see [Section 6.2.2, “ASAM Master User Security,” on page 72.](#)
- ♦ For the rights required by administrative users, see [Section , “Rights Required for Web Application Use,” on page 76.](#)

7.2.2 Platform Services Process / Authentication Services Issues

- ♦ Ensure that your platform has a valid certificate. For more information, see [Part IV, “Platform Services Administration,” on page 139.](#)
- ♦ For platforms that use the DES interface, ensure that the DES encryption key specified for the platform in the Platform Configuration object and the DES encryption key specified in the platform configuration file are identical.
- ♦ Ensure that the Core Drivers referred to by your platform configuration file are running and that they are using the network address and TCP port number specified in the platform configuration file.
- ♦ Examine the Core Driver and Platform Services Process log files.
- ♦ Ensure that your platforms have been upgraded to the current version when you upgrade the Core Driver.

7.2.3 Platform Receiver / Event Journal Services Issues

- ♦ Verify that LDAP is running on the servers that hold replicas of User and Group objects covered by Census Search objects.
- ♦ Verify that the platform certificate has been created on the host running the Platform Receiver.
- ♦ Ensure that the Core Drivers referred to by your platform configuration file are running and that they are using the network address and TCP port number specified in the platform configuration file.
- ♦ Examine the logs generated by the Platform Receiver and the Core Driver for error messages.

7.2.4 Census Issues

- ♦ Ensure that Census Trawls are being run at appropriate intervals and that they complete without errors.

You can check to see when daily Census Trawls are scheduled through the Web interface. For details, see [“Specifying Trawl Times” on page 81.](#)

You can view the Exceptions to see if any naming exceptions have occurred. For details, see [Section 6.5.14, “Reviewing Naming Exceptions,” on page 96.](#)

You can check the Core Driver Operational Log to see when the last Census Trawl was completed. Examine the Core Driver Operational Log for any potential errors that could have prevented the successful creation of Census information.

- ◆ Ensure that your Census Search object parameters are set so that all intended users are included in the Census. For details, see “[Configuring Search Objects](#)” on page 92. You can review the contents of the Census by using the Web interface. For further details, see “[Displaying Provisioning Details](#)” on page 96.
- ◆ Census entries relate to objects in eDirectory using their globally unique identifier (GUID). This prevents accidental reuse of a name from resulting in unintended access to resources. If a user that is represented in the Census is removed from eDirectory, the Census entry is marked inactive. If a new User object with the same name and context is created in eDirectory, the old Census entry remains inactive and a naming exception occurs.

7.2.5 LDAP Issues

For problems with LDAP secure communications, see Novell Technical Information Document (TID) 10050254 at the [Novell Support Web Site \(http://support.novell.com/cgi-bin/search/searchtid.cgi?/10050254.htm\)](http://support.novell.com/cgi-bin/search/searchtid.cgi?/10050254.htm).

7.3 Troubleshooting Network Issues

The details of network troubleshooting are beyond the scope of this document and depend on a number of factors particular to r environment. This section can help you determine if the various Identity Manager Fan-Out Driver components can communicate with one another.

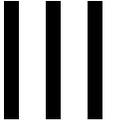
To verify IP connections between platforms and the Core Driver, use the ping command. From a command prompt on z/OS, OS/400, Linux, UNIX, or Windows, enter `ping ipaddr`, where *ipaddr* is the IP address of the remote computer.

If your installation uses router filters to prevent the use of ping, consult with those responsible for managing your network to learn how to verify connectivity.

7.4 Troubleshooting eDirectory Issues

eDirectory is complex, and the details of troubleshooting are beyond the scope of this document. Refer to your eDirectory documentation for further details. Additional information can be found on the [Novell Support Web site \(http://support.novell.com\)](http://support.novell.com).

Platform Services Planning



Part III provides you with information you need to plan deployment of Platform Services for the Novell[®] Identity Manager Fan-Out Driver. It includes the following chapters:

- ♦ Chapter 8, “Platform Services Overview,” on page 105
- ♦ Chapter 9, “Planning for Platform Services,” on page 115
- ♦ Chapter 10, “The Platform Configuration File,” on page 119

Platform Services Overview

8

This section presents an overview of the Platform Services part of the Novell® Identity Manager Fan-Out Driver. Platform Services makes requests to Core Drivers for Authentication Services and provisioning events.

If the Fan-Out Driver is new to you, read the information presented in [Part I, “Concepts and Facilities,”](#) on page 13.

Topics in this section are

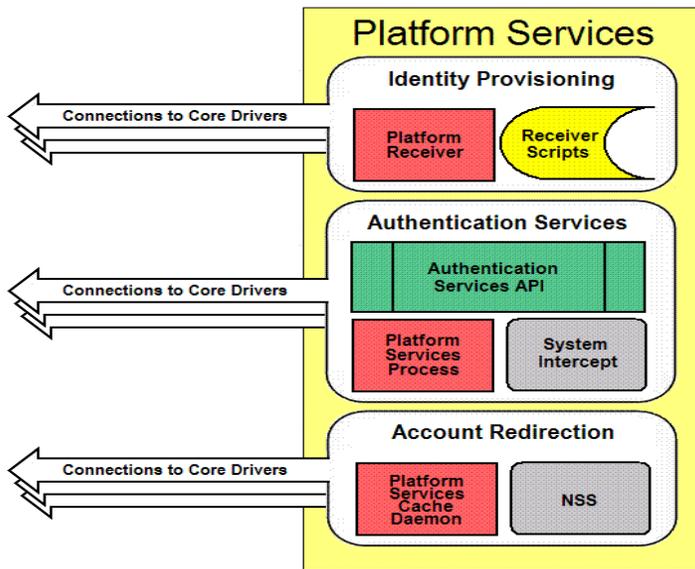
- ◆ [Section 8.1, “Platform Services Component Summary,”](#) on page 105
- ◆ [Section 8.2, “Authentication Services,”](#) on page 106
- ◆ [Section 8.3, “Identity Provisioning,”](#) on page 107
- ◆ [Section 8.4, “Account Redirection,”](#) on page 107
- ◆ [Section 8.5, “The Platform Services Process,”](#) on page 108
- ◆ [Section 8.6, “The Platform Services Cache Daemon,”](#) on page 108
- ◆ [Section 8.7, “The System Intercept,”](#) on page 108
- ◆ [Section 8.8, “The Platform Receiver,”](#) on page 109
- ◆ [Section 8.9, “Receiver Scripts,”](#) on page 111
- ◆ [Section 8.10, “Standard Exclude List,”](#) on page 112

8.1 Platform Services Component Summary

The Platform Services Process makes requests to Core Drivers for Authentication Services functions such as authentication, user name resolution, and password changes.

NOTE: For an overview of how Platform Services works with the components of the Core Driver—the other major part of Identity Manager Fan-Out Driver architecture—see [Part I, “Concepts and Facilities,”](#) on page 13.

Figure 8-1 Platform Services



The Platform Services System Intercept is hooked into the login process of a system using standard, vendor-provided mechanisms. It provides password verification and password change functions.

The Platform Receiver obtains provisioning events from Event Journal Services and acts on them by running Receiver scripts to create and maintain users and groups as appropriate.

Platform Services also provides an application programming interface (API) that you can use for your own applications. For more information, see [Part V, “API Development,” on page 151](#).

Some types of platforms communicate with the Core Driver for Authentication Services using Secure Sockets Layer (SSL). Others use DES encryption. All platform communication with Event Journal Services uses SSL.

The Platform Services Cache Daemon obtains provisioning events from Event Journal Services and stores them in a local memory cache for efficient retrieval by the Name Service Switch. This information contains a complete record for a Linux or UNIX account or group, which may be accessed by services that use the Name Service Switch system calls.

The Name Service Switch is a system library providing complete account redirection as an alternative to storing user and group accounts and passwords locally. By providing such services through a memory cache, this data is protected from interactive accounts on the local system. In addition, the data remains centrally managed by eDirectory™ and a large number of accounts may be accessed by a single Linux or UNIX system, improving on traditional `/etc/passwd` methods for accomplishing this, which can be inefficient to update or access.

8.2 Authentication Services

Authentication Services uses eDirectory for functions such as user authentication. The Platform Services Process, together with the System Intercept, provides Authentication Services on a platform.

z/OS*, Linux and UNIX systems can redirect password verification and password changes through Authentication Services to eDirectory. An IBM* i5/OS (OS/400) system can authenticate users locally, but uses Authentication Services to replicate passwords in its password store from the passwords of objects in eDirectory that correspond to its users. z/OS, Linux and UNIX systems can supplement password redirection with password replication for fail-safe operation.

The Identity Manager Fan-Out Driver uses the system intercept on Windows* systems to capture password change information and store it in eDirectory. Password change information from eDirectory is delivered to authorized systems as provisioning events, replicating password information from eDirectory.

You can use the platform configuration file to specify which users use Authentication Services and which ones authenticate locally. The driver has a built-in list of special users that, by default, are excluded from Authentication Services. For more information about the platform configuration file, see [Chapter 10, “The Platform Configuration File,” on page 119](#). For more information about the standard exclude list, see [Section 8.10, “Standard Exclude List,” on page 112](#).

8.3 Identity Provisioning

Identity Provisioning uses events from eDirectory to provision user and group account information to the platform. The Platform Receiver, together with the Receiver scripts, provides Identity Provisioning on a platform.

You can use the platform configuration file to specify which users and groups are managed using Identity Provisioning and which ones are managed locally. The driver has a built-in list of special users and groups that, by default, are excluded from Identity Provisioning. For more information about the platform configuration file, see [Chapter 10, “The Platform Configuration File,” on page 119](#). For more information about the standard exclude list, see [Section 8.10, “Standard Exclude List,” on page 112](#).

Each managed user and group is assigned the same UID and GID number across all Linux/UNIX platforms in a Platform Set.

8.4 Account Redirection

NOTE: This section only applies to Linux and UNIX connected systems.

Account Redirection uses eDirectory for user and group account information. By extending your users and groups with the posixAccount and posixGroup auxiliary classes, you can assign the following Posix attributes for your users and groups:

- ◆ loginName
- ◆ uidNumber
- ◆ gidNumber
- ◆ gecos
- ◆ homeDirectory
- ◆ loginShell
- ◆ groupName
- ◆ memberUid

These attributes correspond to the following lines in `/etc/passwd`:

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

For groups, they correspond to the following lines in `/etc/group`:

```
groupName:x:gidNumber:memberUid
```

This Posix information will be synchronized to a local memory cache on your Linux or UNIX system and accessed by the Name Service Switch. The Name Service Switch runs when a user logs on and also when a command that requires information from this or another user or group account is invoked.

8.5 The Platform Services Process

The Platform Services Process provides the Authentication Services interface to eDirectory by communicating with the Core Driver. This interface is called by the System Intercept for such functions as checking a user's password at log in. It is also used by the Authentication Services (AS) Client API to provide eDirectory access to your own applications. For details about using the AS Client API, see [Part V, “API Development,” on page 151](#).

The Platform Services Process maintains persistent connections with Core Drivers for Authentication Services and performs load balancing and failover.

The Platform Services Process obtains configuration information, such as the location of Core Drivers, from the platform configuration file. For additional information about the platform configuration file, see [Chapter 10, “The Platform Configuration File,” on page 119](#).

On platforms where the volume of traffic with Core Drivers is so low that running the Platform Services Process is not justified by the performance benefits, you can connect the System Intercept and AS Client API directly to core services. For details, see [“DIRECTTOAUTHENTICATION Statement” on page 127](#).

8.6 The Platform Services Cache Daemon

NOTE: This section only applies to Linux and UNIX connected systems.

The Platform Services Cache Daemon obtains provisioning events from the Event Journal Services component of the Core Driver. The Platform Services Cache Daemon examines each event and updates the local cache for the appropriate account record. When the daemon is run for the first time, a full sync is performed to synchronize all users and groups with the local cache. This may take some time, depending on how many users and groups you have to synchronize with the platform. However, once this full sync occurs, the data is written to a protected and encrypted local file cache for temporary storage when the cache daemon is shut down. Upon startup, this cache is loaded into memory again and updated by Event Journal Services when changes are made in eDirectory.

8.7 The System Intercept

System integration of Platform Services makes use of standard, vendor-provided system control points.

Details about configuring and administering Platform Services are provided in later sections of this guide. Also be aware that this guide is one of three available administration guides for the Fan-Out Driver, each tailored to the range of platforms with which it can work:

- ♦ *Identity Manager Fan-Out Driver for Linux and UNIX Administration Guide*
- ♦ *Identity Manager Fan-Out Driver for Mainframes Administration Guide (z/OS)*
- ♦ *Identity Manager Fan-Out Driver for Midrange Administration Guide (OS/400, i5/OS)*

System integration of Platform Services for z/OS makes use of standard exits provided by the security system in use (RACF*, CA* ACF2*, or CA Top Secret*). For more information, see the *Identity Manager Fan-Out Driver for Mainframes Administration Guide*.

System integration of Platform Services for most Linux/UNIX systems makes use of the Pluggable Authentication Module (PAM) framework that is defined by OSF RFC 86.0. Applications must make the appropriate PAM API calls in order to be PAM-aware. You can also modify your applications to use the AS Client API directly. For more information, see the *Identity Manager Fan-Out Driver for Linux and UNIX Administration Guide*.

System integration of Platform Services for AIX* supports both the Loadable Authentication Module (LAM) system provided by AIX and the PAM framework; you choose which you wish to use. The PAM framework is only recommended for AIX versions 5.3 and later.

Password changes on an OS/400 system are provided to the Core Driver through the Password Change Validation Program Exit, which is controlled by system value QPWDVLDPGM. Password changes in eDirectory are received by the platform as provisioning events. For additional information, see the *Identity Manager Fan-Out Driver for Midrange Administration Guide*.

8.8 The Platform Receiver

The Platform Receiver obtains provisioning events from the Event Journal Services component of the Core Driver. The Platform Receiver examines each event and the current status of users and groups on the platform. Then the Platform Receiver calls Receiver scripts as necessary to perform needed changes. On password replication platforms, the Platform Receiver also updates the local password store.

The Platform Receiver provides failover support for connections to Event Journal Services.

The Platform Receiver obtains configuration information, such as its mode of operation and the location of the Core Driver, from the platform configuration file. For additional information about the platform configuration file, see [Chapter 10, “The Platform Configuration File,” on page 119](#).

8.8.1 Modes of Operation

The Platform Receiver can be configured to obtain and process provisioning events in five different modes.

Full Sync Mode

In Full Sync Mode, the Platform Receiver connects to Event Journal Services and requests a Full Sync. Event Journal Services provides, and the Platform Receiver processes, a complete set of provisioning events to populate the users and groups for the platform. Then the Platform Receiver ends.

The first time a Platform Receiver is run for a new platform, it automatically receives provisioning events for all users and groups for the platform. If this process is interrupted, processing resumes the next time the Platform Receiver is run. There is no need to run the Platform Receiver in Full Sync Mode during routine installation.

You can run the Platform Receiver in Full Sync Mode to recover from a disaster on the platform that affects the user or group population.

You can run the Platform Receiver in Full Sync Mode any other time as appropriate to ensure that the user and group population on the platform is consistent with eDirectory.

If a Full Sync operation is interrupted, the provisioning process resumes the next time the Platform Receiver is run in Persistent Mode, Polling Mode, or Scheduled Mode. Do not start the Platform Receiver in Full Sync Mode to recover from an interrupted Full Sync operation, because Full Sync processing starts from the beginning each time.

Check Mode

Check Mode functions similarly to Full Sync Mode, except that Receiver scripts are invoked in Check Mode. In Check Mode, the base scripts take no actions to alter the user or group population on the platform.

If you extend the base scripts, take no actions that alter the user or group population while Check Mode is in effect.

Operation in Check Mode does not affect the queue of pending events maintained by Event Journal Services for the platform.

Check Mode is useful for testing your extensions to Receiver scripts.

You can use Check Mode at any time to verify that the user and group population on the platform is consistent with eDirectory.

Persistent Mode

In Persistent Mode, the Platform Receiver connects to Event Journal Services, obtains queued provisioning events, and processes them. It then remains connected, processing additional events as they become available.

Polling Mode

In Polling Mode, the Platform Receiver connects to Event Journal Services, obtains queued provisioning events, and processes them. The Platform Receiver then closes the connection, waits for five minutes, and repeats the process until you stop it. For details about specifying the interval between polling cycles, see [“POLLINT Statement” on page 131](#) and [“POLLRAND Statement” on page 131](#).

Scheduled Mode

In Scheduled Mode, the Platform Receiver connects to Event Journal Services, obtains queued provisioning events, and processes them. It then closes the connection and ends. Scheduled Mode is designed for use with external job schedulers, such as the UNIX cron utility.

8.8.2 Selecting a Mode of Operation

You specify the mode of operation for the Platform Receiver through the `RUNMODE` statement in the platform configuration file or through a command line parameter. For details about specifying the `RUNMODE` statement, see [“RUNMODE Statement” on page 132](#).

You can periodically run the Platform Receiver in Full Sync Mode to ensure that accounts on the platform are consistent with eDirectory.

For routine operations, we recommend that, unless you need the real-time processing of events provided by Persistent Mode, you run the Platform Receiver in Polling Mode or Scheduled Mode. This reduces the number of concurrent connections that must be serviced by the Core Driver host. The frequency of change activity in the population, the operating schedule of the platform, and the nature of the connection between the platform and the Core Driver should help you determine which of these modes to use.

You can use Check Mode for testing extensions to Receiver scripts.

8.9 Receiver Scripts

NOTE: This section only applies to Linux and UNIX connected systems.

The Platform Receiver examines the provisioning events it obtains from Event Journal Services and inspects the state of users and groups on the platform. Then the Platform Receiver calls Receiver scripts as needed to make the state of users and groups on the platform consistent with eDirectory.

The Identity Manager Fan-Out Driver provides a set of fully functional base scripts. You can extend these base scripts as appropriate for your needs. For example, if you have a third party system that uses its own user ID database, you can extend the base scripts to add new users to it based on membership in a special group, and to remove users from it when they are removed from the group.

The Receiver script functions are

- ◆ Add User
- ◆ Modify User
- ◆ Delete User
- ◆ Delete User Pending
- ◆ Enable User
- ◆ Disable User
- ◆ Rename User
- ◆ Add User to Group
- ◆ Remove User from Group
- ◆ Add Group
- ◆ Modify Group
- ◆ Delete Group
- ◆ Delete Group Pending
- ◆ Rename Group

NOTE: These are the functions performed by Receiver scripts. The actual implementation is platform-OS dependent. Multiple functions might be combined into a single script, or the steps of a single function might be distributed across several scripts. Not all functions are meaningful for all platform-OS types.

In addition to the scripts that perform actions on users and groups, there are utility scripts that are used for such functions as testing for the existence of a user and checking group membership.

The base scripts are extensively commented. For details on the operation of the base scripts, examine the scripts themselves. Become thoroughly familiar with the operation of a base script before you attempt to extend it.

8.10 Standard Exclude List

Platform Services normally excludes certain special users from Authentication Services processing and Identity Provisioning. You can use the platform configuration file to override this or to specify additional users and groups to be excluded.

Users excluded from Authentication Services are authenticated using the local security system. Provisioning events are not processed for users and groups excluded from Identity Provisioning.

For details about Include/Exclude processing, see

- ◆ [Section 10.4, “Using Include and Exclude Configuration Statements,” on page 135](#)
- ◆ [“AM.GROUP.INCLUDE Statement / AM.GROUP.EXCLUDE Statement” on page 122](#)
- ◆ [“AM.USER.INCLUDE Statement / AM.USER.EXCLUDE Statement” on page 123](#)
- ◆ [“AS.USER.INCLUDE Statement / AS.USER.EXCLUDE Statement” on page 124](#)

Following is the standard list of users and groups that are excluded from Authentication Services and Identity Provisioning processing.

Account Operators	adm	admin
administrator	administrators	audit
Backup Operators	bin	Cert Publishers
cron	daemon	DB2XML
DHCP Administrators	dip	disk
DnsAdmins	DnsUpdateProxy	Domain Admins
Domain Computers	Domain Controllers	ecs
Enterprise Admins	floppy	ftp
games	gdm	gopher
Group Policy Creator Owners	guest	halt
hpdb	ibmuser	ident
imnadm	IUSR_WIN2KEDIR	IWAM_WIN2KEDIR
kmem	krbtgt	ldap

listen	lock	lp
lpd	mail	mailnull
man	mem	MTS Impersonators
news	nfsnobody	noaccess
nobody	nobody4	nogroup
nscd	ntp	nusers
nuucp	nwgroup	nwdap
nwprint	nwroot	nwuser
operator	other	perf
Print Operators	printq	QAUTPROF
QBRMS	QCLUMGT	QCLUSTER
QCOLSRV	QDBSHR	QDBSHRDO
QDESADM	QDESUSR	QDFTOWN
QDIRSRV	QDLFM	QDOC
QDSNX	QEJB	QFNC
QGATE	QIJS	QIPP
QLPAUTO	QLPINSTALL	QMSF
QNETSPLF	QNETWARE	QNFSANON
QNTP	QPEX	QPGMR
QPM400	QRJE	QSECOFR
QSNADS	QSPL	QSPLJOB
QSRV	QSRVBAS	QSVCDRCTR
QSYS	QSYSOPR	QTCM
QTCP	QTFTP	QTMHHTTP1
QTMHHTTP	QTMLPD	QTSTRQS
QUMB	QUSER	QYPSJSVR
radvd	RAS and IAS Servers	Replicator
root	rpc	rpcuser
rpm	Schema Admins	security
Server Operators	shutdown	slocate
staff	sync	sys
sys1	sysadmin	system
TsInternetUser	tty	users
usr	utmp	uucp

wheel

wine

www

xfz

Planning for Platform Services

9

This section helps you plan for deployment of Platform Services for the Novell® Identity Manager Fan-Out Driver. If the Fan-Out Driver is new to you, read the information presented in **Part I, “Concepts and Facilities,”** on page 13 before proceeding.

- ◆ **Section 9.1, “Basic Considerations,”** on page 115
- ◆ **Section 9.2, “Security Planning Considerations,”** on page 115
- ◆ **Section 9.3, “Planning Considerations for Authentication Services,”** on page 116
- ◆ **Section 9.4, “Planning Considerations for Identity Provisioning,”** on page 117
- ◆ **Section 9.5, “Planning Considerations for Password Replication Platforms,”** on page 118
- ◆ **Section 9.6, “Planning Considerations for Account Redirection Platforms,”** on page 118

9.1 Basic Considerations

Before you can install and use Platform Services, you must complete the installation of at least one Core Driver and have it running.

The installation planning process for the Core Driver addresses a number of installation-wide issues. Review **Part II, “Core Driver Administration,”** on page 37, especially the planning section, before you proceed.

For information about required software and systems, as well as supported platforms and operating environments, see the [Identity Manager 3.6 Drivers Documentation Web site \(http://www.novell.com/documentation/idm36drivers\)](http://www.novell.com/documentation/idm36drivers). From this index page, you can select a readme file associated with the platform(s) for which you need support.

9.2 Security Planning Considerations

Topics in this section include

- ◆ **Section 9.2.1, “Users, Passwords, and Groups,”** on page 115
- ◆ **Section 9.2.2, “Connection Security,”** on page 116
- ◆ **Section 9.2.3, “Administrative Password Resets,”** on page 116
- ◆ **Section 9.2.4, “Securing the AS Client API,”** on page 116

9.2.1 Users, Passwords, and Groups

In order for users to be able to log in to the operating system using Authentication Services, they must be defined to the operating system on the platform. You can automate account maintenance through the use of provisioning events. For details about managing accounts, see **Section 8.3, “Identity Provisioning,”** on page 107.

Users, passwords, and groups in eDirectory™ that do not conform to the character set and length restrictions imposed by your operating system cannot participate in Authentication Services or Identity Provisioning on your platform.

The Identity Manager Fan-Out Driver does not support authentication or password change for users having a null password.

In some cases, a system other than eDirectory might contain the users that you want to participate with the Identity Manager Fan-Out Driver. There are tools, such as LDIF, that you can use to import these users into eDirectory. If you cannot extract the passwords for the affected user accounts, you can use the Password Migration component of the Fan-Out Driver. This component can help you accomplish a smooth transition to basing your user accounts in eDirectory. The Password Migration component requires z/OS. For details about the Password Migration component, see the *Identity Manager Fan-Out Driver for Mainframes Administration Guide*.

9.2.2 Connection Security

The connection between the Platform Receiver and Event Journal Services uses Secure Sockets Layer (SSL). SSL connections are authenticated through the use of certificates. Some types of the Platform Services Process use SSL for their connections to the Core Drivers for Authentication Services, and others use DES encryption.

Obtaining a security certificate for your platform from the Core Driver requires that you supply the fully distinguished name and password of an eDirectory user with Read and Create object rights to the ASAM System container.

Identity Manager Fan-Out Driver platform certificates are stored in the `data\platformservices\certs` subdirectory of the ASAM directory of their host server file system. Ensure that access to the `certs` directory is limited to the appropriate users.

9.2.3 Administrative Password Resets

Administrative password resets must be done through an eDirectory utility, such as iManager, or through a program that uses the AS Client API.

9.2.4 Securing the AS Client API

Use of the AS Client API is secured on OS/400 and UNIX platforms through SSL and a token that is stored in the `asam\data\platformservices\certs` directory by the Platform Services Process. Ensure that access to the `certs` directory is limited to the appropriate users.

Use of the AS Client API is secured on z/OS Platforms through the Authorized Program Facility (APF). Ensure that access to the z/OS Platform Services Load Library is limited to the appropriate users.

9.3 Planning Considerations for Authentication Services

When planning for Authentication Services, include the following considerations:

- ♦ If you don't plan to use Authentication Services to authenticate system users or provide password change information to Core Drivers, you don't need to install the System Intercept.
- ♦ If you don't plan to use the AS Client API or Authentication Services, you don't need to run the Platform Services Process.

- ♦ If your use of Authentication Services and the AS Client API is infrequent and does not require high performance, consider using the `DIRECTTOAUTHENTICATION` statement in the platform configuration file. This configuration does not use the Platform Services Process. For details about the `DIRECTTOAUTHENTICATION` statement, see [“DIRECTTOAUTHENTICATION Statement” on page 127](#).
- ♦ You might need to permanently exclude some users from Authentication Services processing. You might want to phase in your implementation by using a subset of your users to start with. For details about excluding users from Authentication Services processing, see [“AS.USER.INCLUDE Statement / AS.USER.EXCLUDE Statement” on page 124](#).
- ♦ You must specify which Core Drivers are used for Authentication Services. You might want to establish different preference groups for sets of these Core Drivers based on their network connectivity or other issues. For details, see [Section 10.3.10, “AUTHENTICATION Statement,” on page 125](#).

9.4 Planning Considerations for Identity Provisioning

When planning for Identity Provisioning, include the following considerations:

- ♦ If you don't plan to use Identity Provisioning, you don't need to run the Platform Receiver.
- ♦ You might need to permanently exclude some users and groups from Identity Provisioning. You might want to phase in your implementation by using a subset of your users and groups to start with. For details about excluding users and groups from Identity Provisioning, see [“AM.USER.INCLUDE Statement / AM.USER.EXCLUDE Statement” on page 123](#) and [“AM.GROUP.INCLUDE Statement / AM.GROUP.EXCLUDE Statement” on page 122](#).
- ♦ If the base Receiver scripts do not meet your needs, you can write your own extensions. Decide what additional processing you will perform and how you will test your extensions.
- ♦ All platforms in a Platform Set have the same population of users and groups associated with them for Identity Provisioning. Users and groups on Linux/UNIX platforms in Platform Sets that share a common UID/GID Set have the same UID or GID on each participating platform. Decide how you will organize your Platform Sets and UID/GID Sets.
- ♦ Only on OS/400 platforms, you might need to map eDirectory attribute names to System Distribution Directory names. For more information, see the *Identity Manager Fan-Out Driver for Midrange Administration Guide*.
- ♦ You must specify which Core Drivers are used for Identity Provisioning. For details, see [Section 10.3.26, “POLLINT Statement,” on page 131](#).
- ♦ You must choose the mode of operation used by the Platform Receiver to obtain events. For details, see [“Modes of Operation” on page 109](#) and [“Selecting a Mode of Operation” on page 111](#).

9.5 Planning Considerations for Password Replication Platforms

When planning for Replication Platforms, include the following considerations:

- ♦ By default, the Core Driver converts passwords to lowercase before sending them to the Platform Receiver. For more information, see [“Lower Password Case” on page 83](#).
- ♦ The Permit Password Replication attribute of a Platform object determines whether provisioning events for user accounts are sent to the platform before the passwords for these accounts are known to the Identity Manager Fan-Out Driver.

Platforms configured with Permit Password Replication set to *Yes* do not receive Provisioning events for user accounts until the account passwords are known to the driver.

Platforms configured with Permit Password Replication set to *If Available* do receive Provisioning events when they occur for an account, even if the password is not known to the driver.

The driver uses system intercepts to collect password information. To be provisioned onto a platform configured with Permit Password Replication set to *Yes*, users must either change their passwords on a platform where the system intercepts are installed and configured, or authenticate on a participating redirection platform.

By planning a staged deployment of the driver so that most users have authenticated using other platforms first, you can ensure the availability of these users to password replication platforms when you are ready to deploy the driver on them.

For more information, see [Section 6.5.6, “Configuring Platforms,” on page 89](#).

9.6 Planning Considerations for Account Redirection Platforms

NOTE: This section only applies to Linux and UNIX connected systems.

When planning for Account Redirection Platforms, include the following considerations:

- ♦ Use the Account Redirection option if you wish to redirect all account information, including loginName, uidNumber, gidNumber, gecos, homeDirectory, loginShell, memberUid fields and passwords.
- ♦ If you plan to use Account Redirection, you do not need to run the Platform Receiver or the Platform Services Process. Instead, you need to configure your system for the Name Service Switch and configure the Platform Services Cache Daemon for system startup.
- ♦ If you plan to use Account Redirection, you must populate your user and group accounts with the posixAccount and posixGroup auxiliary classes. This can be done manually on a per-object basis or through a bulk LDIF import process. Alternatively, you may run the Linux and UNIX User Settings Driver to automatically populate this information when users and groups are created or modified. For details on this driver, see the Identity Manager Driver documentation for the Linux and UNIX user settings.

The Platform Configuration File

10

This section describes the platform configuration file and its use. Novell® Identity Manager Fan-Out Driver Platform Services components use the platform configuration file to locate Core Driver components, to locate their run-time files, and to control their operation.

IMPORTANT: Because the platform configuration file contains sensitive information, you should use the appropriate access controls to restrict its use to the driver system itself, and to its administrators.

Topics in this section include

- ♦ [Section 10.1, “Platform Configuration File Location,” on page 119](#)
- ♦ [Section 10.2, “Platform Configuration File Syntax,” on page 120](#)
- ♦ [Section 10.3, “Configuration Statements,” on page 120](#)
- ♦ [Section 10.4, “Using Include and Exclude Configuration Statements,” on page 135](#)

10.1 Platform Configuration File Location

The location of the platform configuration file depends on which operating system the platform is using. More information than the summaries in this section can be found in [Part IV, “Platform Services Administration,” on page 139](#). Also be aware that this guide is one of three available administration guides for the Fan-Out Driver, each tailored to the range of platforms with which it can work:

- ♦ *Identity Manager Fan-Out Driver for Linux and UNIX Administration Guide*
- ♦ *Identity Manager Fan-Out Driver for Mainframes Administration Guide (z/OS)*
- ♦ *Identity Manager Fan-Out Driver for Midrange Administration Guide (OS/400, i5/OS)*

10.1.1 z/OS

For z/OS, the default configuration file for the Platform Services Process is member ASCPRM00 in the partitioned data set allocated to the ASCPARMS ddname. The default configuration file for the Platform Receiver is the z/OS sequential data set or partitioned data set member allocated to ddname ASAMCONF. You can use a JCL EXEC statement PARM to specify a different configuration file.

10.1.2 i5/OS

For i5/OS (OS/400 operating system), the default configuration file is `/usr/local/ASAM/data/asamplat.conf`. You can use a command line parameter to specify a different platform configuration file.

10.1.3 Linux and UNIX

For Linux and UNIX, the default configuration file is `/usr/local/ASAM/data/asamplat.conf`. You can use a command line parameter to specify a different platform configuration file.

10.2 Platform Configuration File Syntax

Use the following syntax guidelines for the platform configuration file:

- ♦ Any line beginning with an asterisk (*), a semicolon (;), or an octothorpe (#) is a comment. All text that follows a semicolon or an octothorpe is a comment.
- ♦ Configuration file statements are case-insensitive.
- ♦ Except as noted, the order in which statements appear in the file does not matter.
- ♦ Any parameter value that contains spaces must be enclosed in quotes. Do not use quotes with other values. For example:

```
PASSWORDPROMPT "Password: "  
PROVISIONING cdriver1.digitalairlines.com
```

10.3 Configuration Statements

This section describes the following platform configuration file statements:

- ♦ [Section 10.3.1, “ACF2.DISABLE Statement,” on page 121](#)
- ♦ [Section 10.3.2, “ACF2.EXPIREWARN Statement,” on page 121](#)
- ♦ [Section 10.3.3, “ADMINPASSWORD Statement,” on page 122](#)
- ♦ [Section 10.3.4, “ADMINUSER Statement,” on page 122](#)
- ♦ [Section 10.3.5, “AM.GROUP.INCLUDE Statement / AM.GROUP.EXCLUDE Statement,” on page 122](#)
- ♦ [Section 10.3.6, “AM.USER.INCLUDE Statement / AM.USER.EXCLUDE Statement,” on page 123](#)
- ♦ [Section 10.3.7, “AS.USER.INCLUDE Statement / AS.USER.EXCLUDE Statement,” on page 124](#)
- ♦ [Section 10.3.8, “AS.USER.NONNDS Statement,” on page 124](#)
- ♦ [Section 10.3.9, “ASAMDIR Statement,” on page 125](#)
- ♦ [Section 10.3.10, “AUTHENTICATION Statement,” on page 125](#)
- ♦ [Section 10.3.11, “CODEPAGE Statement,” on page 126](#)
- ♦ [Section 10.3.12, “DEBUGLOGFILE Statement,” on page 126](#)
- ♦ [Section 10.3.13, “DEBUGTOSTDOUT Statement,” on page 126](#)
- ♦ [Section 10.3.14, “DIRECTTOAUTHENTICATION Statement,” on page 127](#)
- ♦ [Section 10.3.15, “ENTROPY Statement,” on page 127](#)
- ♦ [Section 10.3.16, “HONORMVSDISABLE Statement,” on page 127](#)
- ♦ [Section 10.3.17, “IGNORESTANDARD EXCLUDES Statement,” on page 128](#)
- ♦ [Section 10.3.18, “KEY Statement,” on page 128](#)

- ◆ Section 10.3.19, “LOCALE Statement,” on page 129
- ◆ Section 10.3.20, “PASSWORDPROMPT Statement,” on page 129
- ◆ Section 10.3.21, “PASSWORDPROMPTCURRENT Statement,” on page 129
- ◆ Section 10.3.22, “PASSWORDPROMPTCHANGE Statement,” on page 129
- ◆ Section 10.3.23, “PASSWORDPROMPTCHANGEAGAIN Statement,” on page 130
- ◆ Section 10.3.24, “PLATFORMNAME Statement,” on page 130
- ◆ Section 10.3.25, “PASSWORDSOURCE Statement,” on page 130
- ◆ Section 10.3.26, “POLLINT Statement,” on page 131
- ◆ Section 10.3.27, “POLLRAND Statement,” on page 131
- ◆ Section 10.3.28, “PROVISIONING Statement,” on page 131
- ◆ Section 10.3.29, “RUNMODE Statement,” on page 132
- ◆ Section 10.3.30, “SECURITY Statement,” on page 132
- ◆ Section 10.3.31, “SMF Statement,” on page 133
- ◆ Section 10.3.32, “SYSLOGFACILITY Statement,” on page 133
- ◆ Section 10.3.33, “TRACEFILE Statement,” on page 133
- ◆ Section 10.3.34, “TRACETOSTDOUT Statement,” on page 134
- ◆ Section 10.3.35, “UPDATEPASSWORD Statement,” on page 134
- ◆ Section 10.3.36, “UPDATESAMBA Statement,” on page 134
- ◆ Section 10.3.37, “USEFILEIPC Statement,” on page 135

10.3.1 ACF2.DISABLE Statement

z/OS only.

The ACF2 .DISABLE statement specifies the handling for ACF2* users who have the Login Disabled attribute set in their corresponding eDirectory™ User objects.

Syntax:

```
ACF2.DISABLE Action
```

Action must be either `suspend` or `cancel`. Specifying either one causes the corresponding CA ACF2 logonid attribute to be set.

If no ACF2 .DISABLE statement is present, the default *Action* is `suspend`.

Example:

```
ACF2.DISABLE cancel
```

10.3.2 ACF2.EXPIREWARN Statement

z/OS only.

The ACF2 .EXPIREWARN statement specifies the number of days before a User object password in eDirectory expires that Platform Services begins to warn the ACF2 user.

Syntax:

ACF2.EXPIREWARN *Days*

Days specifies the number of days that warning messages appear before password expiration.

If no ACF2.EXPIREWARN statement is present, the default is 5 days.

Example:

```
ACF2.EXPIREWARN 14
```

10.3.3 ADMINPASSWORD Statement

The ADMINPASSWORD statement specifies the password of the administrative user specified by the ADMINUSER statement. If there is no ADMINPASSWORD statement, you are prompted to enter the password when obtaining a platform security certificate.

Syntax:

```
ADMINPASSWORD Pswd
```

Pswd specifies the password of the administrative user.

Example:

```
ADMINPASSWORD 18emf25dhf
```

10.3.4 ADMINUSER Statement

The ADMINUSER statement specifies the fully distinguished name of an eDirectory user with Read and Create object rights to the ASAM System container. If there is no ADMINUSER statement, you are prompted to enter a user object name when obtaining a platform security certificate.

Syntax:

```
ADMINUSER Fdn
```

Fdn specifies the fully distinguished name of an eDirectory user with Read and Create object rights to the ASAM System container.

Example:

```
ADMINUSER .Admin.DigitalAirlines
```

10.3.5 AM.GROUP.INCLUDE Statement / AM.GROUP.EXCLUDE Statement

AM.GROUP.INCLUDE and AM.GROUP.EXCLUDE provide a list of specific groups or group masks to be included or excluded from Identity Provisioning. This can be useful for installation verification and early implementation and for special groups that should be managed locally. Multiple AM.GROUP.INCLUDE and AM.GROUP.EXCLUDE statements can be coded, and they can be mixed together. There is no limit to the number of groups that can be included or excluded.

Syntax:

```
AM.GROUP.INCLUDE GroupMask[ GroupMask, GroupMask ...]
```

```
AM.GROUP.EXCLUDE GroupMask[ GroupMask, GroupMask ...]
```

GroupMask can be a single complete group name, or it can include masking characters to represent more than one group. If more than one *GroupMask* matches a given group, the most specific *GroupMask* is used. *GroupMask* is case-insensitive. For more information, see “[Mask Characters and Examples](#)” on page 136.

Unless `AM.GROUP.EXCLUDE *` is coded, `AM.GROUP.INCLUDE *` is always assumed. Certain special groups are always excluded unless the `IGNORESTANDARDEXCLUDES` statement is specified. For details, see “[IGNORESTANDARDEXCLUDES Statement](#)” on page 128.

Do not code both an `AM.GROUP.INCLUDE` statement and an `AM.GROUP.EXCLUDE` statement.

For more information, see [Section 10.4, “Using Include and Exclude Configuration Statements,”](#) on page 135.

Example:

```
AM.GROUP.EXCLUDE sales, mkt*
```

10.3.6 AM.USER.INCLUDE Statement / AM.USER.EXCLUDE Statement

`AM.USER.INCLUDE` and `AM.USER.EXCLUDE` provide a list of specific user IDs or user ID masks to be included or excluded from Identity Provisioning. This can be useful for installation verification and early implementation and for special user IDs that should be managed locally. Multiple `AM.USER.INCLUDE` and `AM.USER.EXCLUDE` statements can be coded, and they can be mixed together. There is no limit to the number of users that can be included or excluded.

Syntax:

```
AM.USER.INCLUDE UserMask[ UserMask, UserMask ...]
```

```
AM.USER.EXCLUDE UserMask[ UserMask, UserMask ...]
```

UserMask can be a single complete user ID, or it can include masking characters to represent more than one user ID. If more than one *UserMask* matches a given user ID, the most specific *UserMask* is used. *UserMask* is case-insensitive. For more information, see “[Mask Characters and Examples](#)” on page 136.

Unless `AM.USER.EXCLUDE *` is coded, `AM.USER.INCLUDE *` is always assumed. Certain special users are always excluded unless the `INGORESTANDARDEXCLUDES` statement is specified. For details, see “[IGNORESTANDARDEXCLUDES Statement](#)” on page 128.

Do not code both an `AM.USER.INCLUDE *` statement and an `AM.USER.EXCLUDE *` statement.

Identity Manager Fan-Out Driver Linux/UNIX platforms always manage root locally.

For more information, see [Section 10.4, “Using Include and Exclude Configuration Statements,”](#) on page 135.

Example:

```
AM.USER.EXCLUDE act*, billing%, sys*, sales48
```

10.3.7 AS.USER.INCLUDE Statement / AS.USER.EXCLUDE Statement

AS.USER.INCLUDE and AS.USER.EXCLUDE provide a list of specific user IDs or user ID masks to be included or excluded from Authentication Services. This can be useful for installation verification and early implementation and for special user IDs that should be authenticated locally. Multiple AS.USER.INCLUDE and AS.USER.EXCLUDE statements can be coded, and they can be mixed together. There is no limit to the number of users that can be included or excluded, although a large list can cause a performance impact because it must be searched for every user login. These statements apply to system authentications only and are not used by the AS Client API routines (although there is an API call to test whether a user ID is excluded).

Syntax:

```
AS.USER.INCLUDE UserMask[ UserMask, UserMask ...]
AS.USER.EXCLUDE UserMask[ UserMask, UserMask ...]
```

UserMask can be a single complete user ID, or it can include masking characters to represent more than one user ID. If more than one *UserMask* matches a given user ID, the most specific *UserMask* is used. *UserMask* is case-insensitive. For more information, see [“Mask Characters and Examples” on page 136](#).

Unless AS.USER.EXCLUDE * is coded, AS.USER.INCLUDE * is always assumed. Certain special users are always excluded unless the IGNORESTANDARD EXCLUDES statement is specified. For details, see [“IGNORESTANDARD EXCLUDES Statement” on page 128](#).

Do not code both an AS.USER.INCLUDE * statement and an AS.USER.EXCLUDE * statement.

For more information, see [Section 10.4, “Using Include and Exclude Configuration Statements,” on page 135](#).

Example:

```
AS.USER.EXCLUDE act*, billing%, sys*, sales48
```

10.3.8 AS.USER.NONNDS Statement

z/OS only.

The AS.USER.NONNDS statement specifies how Platform Services handles users that are defined in the local security system but not in eDirectory. This allows an installation to avoid confusion between the Platform Services security system exit and standard TSO full-screen logon if a user is not defined in eDirectory.

Syntax:

```
AS.USER.NONNDS Action
```

Action specifies the action to take for a user that is defined in the local security system but not in eDirectory. Possible values follow.

Table 10-1 Possible Values for Specified Action

Value	Description
DISABLED	The user ID is handled as though it were revoked in the local security system.

Value	Description
UNDEFINED	The user ID is handled as though it were not defined to the local security system.
AUTHLOCAL	The user ID is authenticated locally, as though the Core Driver was not available. It is not allowed to change its password.
EXCLUDED	The user ID is authenticated as though it were excluded. The user can change its password in the local security system.

In each case, a message is written to the ASCLIENT log describing how the user's authentication request was modified.

If no `AS.USER.NONNDS` statement is provided, the default *Action* is *undefined*.

Example:

```
AS.USER.NONNDS authlocal
```

10.3.9 ASAMDIR Statement

The `ASAMDIR` statement specifies the file path where the Identity Manager Fan-Out Driver is installed. Fan-Out Driver components use `ASAMDIR` to find files needed for execution.

Syntax:

```
ASAMDIR FilePath
```

FilePath specifies the location in file system where the component is installed. If there is no `ASAMDIR` statement, *FilePath* defaults as follows:

- ♦ **z/OS:** /usr/local/ASAM in HFS
- ♦ **OS/400:** /usr/local/ASAM
- ♦ **Linux and UNIX:** /usr/local/ASAM
- ♦ **Windows:** c:\novell\asam

Example:

```
ASAMDIR c:\novell\asam
```

10.3.10 AUTHENTICATION Statement

The `AUTHENTICATION` statement specifies the network address and port of one Core Driver used for Authentication Services. In order to use Authentication Services, you must have at least one `AUTHENTICATION` statement in your configuration file.

A maximum of 100 `AUTHENTICATION` statements can be coded.

Syntax:

```
AUTHENTICATION Address [PORT PortNumber] [PREF PrefGroup]
```

Address specifies the DNS name or IP address of a Core Driver used for Authentication Services.

PortNumber specifies the TCP port number that is to be used to communicate with this Core Driver. *PORT* is optional. *PortNumber* defaults to 3451.

IMPORTANT: If you specify a port number other than the default, you must also use the Web interface to specify the same port number for the Core Driver configuration object.

PrefGroup specifies the Preference Group Number that determines the way a Core Driver is selected. It is optional, and the default is for all Core Drivers listed to be in Preference Group 1. Core Drivers within a Preference Group are selected equally for load balancing. Core Drivers with the lowest Preference Group Number are always tried first, followed by the Core Drivers with the next Preference Group Number, and so on, until a Core Driver can be contacted. Preference Group Number must be coded as a positive integer.

Examples:

```
AUTHENTICATION cdriver1.digitalairlines.com
AUTHENTICATION cdriver2.digitalairlines.com
AUTHENTICATION cdriver5.digitalairlines.com PORT 5009 PREF 2
```

10.3.11 CODEPAGE Statement

Linux/UNIX only.

The CODEPAGE statement specifies a code page to be used by the Platform Receiver. Data received and sent by the Platform Receiver is encoded in UTF-8.

Syntax:

```
CODEPAGE CodepageID
```

CodepageID specifies the code page to be used by the Platform Receiver for converting values from and to UTF-8. For information about the available choices for *CodepageID*, see the man page for iconv on your system.

If no CODEPAGE statement is present, UTF-8 values are used without conversion.

Example:

```
CODEPAGE iso88591
```

10.3.12 DEBUGLOGFILE Statement

The DEBUGLOGFILE statement specifies a destination file for debugging data written when the -d command line parameter is present for a component.

The DEBUGLOGFILE statement is not available for z/OS.

Syntax:

```
DEBUGLOGFILE FilePath
```

FilePath specifies the location in file system where the debugging output is to be written.

Example:

```
DEBUGLOGFILE /usr/local/ASAM/debug.txt
```

10.3.13 DEBUGTOSTDOUT Statement

The DEBUGTOSTDOUT statement specifies that debugging data is to be written to the standard output channel stdout when the -d command line parameter is present for a component.

The `DEBUGTOSTDOUT` statement is not available for z/OS.

Syntax:

```
DEBUGTOSTDOUT
```

Example:

```
DEBUGTOSTDOUT
```

10.3.14 DIRECTTOAUTHENTICATION Statement

The `DIRECTTOAUTHENTICATION` statement causes Authentication Services to connect directly to a Core Driver for Authentication Services without using the Platform Services Process. Use the `DIRECTTOAUTHENTICATION` statement on platforms where the volume of traffic with Core Drivers is so low that running the Platform Services Process is not justified.

Platforms using the `DIRECTTOAUTHENTICATION` statement do not perform Core Driver load balancing, although failover support is available if you specify multiple `AUTHENTICATION` statements.

The `DIRECTTOAUTHENTICATION` statement is not available for z/OS.

Syntax:

```
DIRECTTOAUTHENTICATION
```

Example:

```
DIRECTTOAUTHENTICATION
```

10.3.15 ENTROPY Statement

Linux/UNIX only.

The `ENTROPY` statement specifies the file where components obtain entropy for SSL.

Syntax:

```
ENTROPY FilePath
```

FilePath specifies the file that contains entropy.

If no `ENTROPY` statement is coded, the default is to use the `/dev/random` device for entropy. If there is no `/dev/random` device, the default entropy file is `/etc/entropy`.

If your platform has a `/dev/random` device, you do not need to code an `ENTROPY` statement.

Example:

```
ENTROPY /etc/entropy
```

10.3.16 HONORMVSDISABLE Statement

z/OS only.

The `HONORMVSDISABLE` statement controls whether or not the local z/OS platform's security system user disabled status is honored if the user is enabled for login in eDirectory.

Users whose Login Disabled attribute is set in eDirectory cannot log on through Authentication Services regardless of the setting in the local security system.

Syntax:

```
HONORMVSDISABLE Action
```

Action must be either YES or NO.

If *Action* is YES, a user that is enabled in eDirectory but disabled in the local security system is not allowed to log on.

If *Action* is NO, a user that is enabled in eDirectory but disabled in the local security system is allowed to log on, and the flag in the local security system is set to enable logons.

For RACF, the Revoke attribute is used to determine the local disabled status.

For CA ACF2, the logonid attribute specified by the ACF2 . DISABLE statement is used to determine the local disabled status.

For CA Top Secret, the PSUSPEND flag is used to determine the local disabled status. The ASUSPEND flag is not programmatically accessible. Users with the ASUSPEND flag set are always prevented by CA Top Secret from logging on.

If no HONORMVSDISABLE statement is present, the default *Action* is NO.

Example:

```
HONORMVSDISABLE YES
```

10.3.17 IGNORESTANDARDEXCLUDES Statement

The IGNORESTANDARDEXCLUDES statement specifies that the standard list of users and groups excluded from Identity Provisioning and Authentication Services processing is not used. If this statement is not present, the standard list of excludes is used. For the standard list of excluded users and groups, see [Section 8.10, “Standard Exclude List,” on page 112](#).

Syntax:

```
IGNORESTANDARDEXCLUDES
```

Example:

```
IGNORESTANDARDEXCLUDES
```

10.3.18 KEY Statement

z/OS only.

The KEY statement specifies a 56-bit DES encryption key for communications between a platform that uses DES encryption, and Core Drivers.

Syntax:

```
KEY KeyValue
```

KeyValue specifies the value of the key. *KeyValue* must be entered as sixteen hexadecimal digits (0-9, A-F, a-f). The Web interface is used to enter this value in the corresponding Platform object. You must specify the same key value in both places. For details about using the Web interface to set the attributes of a Platform object, see [Section 6.5.6, “Configuring Platforms,” on page 89](#).

Example:

```
KEY 0f4b9692d8bca5f6
```

10.3.19 LOCALE Statement

The LOCALE statement identifies the language to be used by the component.

Syntax:

```
LOCALE Id
```

Id specifies the two-character ISO 639 language identifier.

Example:

```
LOCAL en
```

10.3.20 PASSWORDPROMPT Statement

Linux/UNIX only.

The PASSWORDPROMPT statement specifies the prompt issued by the PAM module to request the user's password for authentication.

Syntax:

```
PASSWORDPROMPT Text
```

If there is no PASSWORDPROMPT statement, *Text* defaults to

```
"NDS Password: "
```

Example:

```
PASSWORDPROMPT "Password: "
```

10.3.21 PASSWORDPROMPTCURRENT Statement

Linux/UNIX only.

The PASSWORDPROMPTCURRENT statement specifies the prompt issued by the PAM module to request the user's current password for password changes.

Syntax:

```
PASSWORDPROMPTCURRENT Text
```

If there is no PASSWORDPROMPTCURRENT statement, *Text* defaults to

```
"Current NDS Password: "
```

Example:

```
PASSWORDPROMPTCURRENT "Enter Current Password: "
```

10.3.22 PASSWORDPROMPTCHANGE Statement

Linux/UNIX only.

The PASSWORDPROMPTCHANGE statement specifies the prompt issued by the PAM module to request the user's new password for password changes.

Syntax:

```
PASSWORDPROMPTCHANGE Text
```

If there is no `PASSWORDPROMPTCHANGE` statement, *Text* defaults to "New NDS Password: "

Example:

```
PASSWORDPROMPTCHANGE "New Password: "
```

10.3.23 PASSWORDPROMPTCHANGEAGAIN Statement

Linux/UNIX only.

The `PASSWORDPROMPTCHANGEAGAIN` statement specifies the prompt issued by the PAM module to verify the user's new password for password changes.

Syntax:

```
PASSWORDPROMPTCHANGEAGAIN Text
```

If there is no `PASSWORDPROMPTCHANGEAGAIN` statement, *Text* defaults to "Re-enter New NDS Password: "

Example:

```
PASSWORDPROMPTCHANGEAGAIN "Verify New Password: "
```

10.3.24 PLATFORMNAME Statement

The `PLATFORMNAME` statement specifies the common name of the Platform object. If there is no `PLATFORMNAME` statement, you are prompted to enter the name when obtaining a platform security certificate.

Syntax:

```
PLATFORMNAME Cn
```

Cn specifies the common name of the Platform configuration object that was specified in the Web interface when platform was defined.

Example:

```
PLATFORMNAME WestCentral
```

10.3.25 PASSWORDSOURCE Statement

Linux/UNIX only. The `PASSWORDSOURCE` statement specifies the location of the `passwd` or `shadow` formatted file where the encrypted passwords reside on the system and is typically not used.

Syntax:

```
PASSWORDSOURCE fully-qualified-path-to-file
```

An important example of the `PASSWORDSOURCE` statement's use is on HP-UX, when shadow passwords are enabled. The default location to look for encrypted passwords on non-trusted-mode HP-UX is `/etc/passwd`. When shadow passwords are enabled on HP-UX, `PASSWORDSOURCE` should be set like the following example:

```
PASSWORDSOURCE /etc/shadow
```

10.3.26 POLLINT Statement

Linux/UNIX only.

The `POLLINT` statement specifies the interval in number of seconds between polling cycles. It can be used to spread the amount of traffic and load placed on the Core Driver during event processing. The default for the polling interval is 300 seconds (5 minutes), which, in some circumstances, may be too frequent.

Syntax:

```
POLLINT seconds
```

Example:

```
POLLINT 600
```

In the above example, `POLLINT` will set the polling interval to 600 seconds (10 minutes).

10.3.27 POLLRAND Statement

Linux/UNIX only.

The `POLLRAND` statement specifies the minimum and maximum polling interval in seconds from which to choose a random interval between polling cycles. It can be used to spread the amount of traffic and load placed on the Core Driver during event processing. The default for the polling interval is 300 seconds (5 minutes), which, in some circumstances, may be too frequent.

Syntax:

```
POLLRAND minsec maxsec
```

Example:

```
POLLRAND 300 1200
```

In the above example, `POLLRAND` will set the platform receiver (`asamrcvr`) to select a random integer between 300 and 1200 to represent the interval in number of seconds before the next polling cycle begins.

10.3.28 PROVISIONING Statement

The `PROVISIONING` statement specifies the network address and port of one Provisioning Manager Core Driver. A `PROVISIONING` statement must appear in the configuration file for the Platform Receiver and when obtaining a security certificate.

You can code more than one `PROVISIONING` statement. The first `PROVISIONING` statement in the file identifies the Provisioning Manager that is tried first. If a connection with the Provisioning Manager identified by the first `PROVISIONING` statement fails, the Provisioning Managers identified by any other `PROVISIONING` statements are tried, in the order the `PROVISIONING` statements appear in the configuration file, until a connection is successful or there are no more `PROVISIONING` statements.

Syntax:

```
PROVISIONING Address [PORT PortNumber]
```

Address specifies the DNS name or IP address of a Provisioning Manager.

PortNumber specifies the TCP port number that is to be used to communicate with the Provisioning Manager. *PORT* is optional. The default port number for the Provisioning Manager is 3451.

IMPORTANT: If you specify a port number other than the default, you must also use the Web interface to specify the same port number for the Core Driver configuration object.

Example:

```
PROVISIONING cdriver1.digitalairlines.com
PROVISIONING cdriver4.digitalairlines.com
```

10.3.29 RUNMODE Statement

The RUNMODE statement specifies the mode of operation the Platform Receiver uses. Command line parameters that specify a mode of operation override the mode specified on the RUNMODE statement.

Syntax:

```
RUNMODE Mode
```

Mode specifies the mode of operation for the Platform Receiver. Possible values follow.

Table 10-2 Values For Specified Mode

Value	Description
PERSISTENT	The Platform Receiver uses Persistent Mode.
POLLING	The Platform Receiver uses Polling Mode.
SCHEDULED	The Platform Receiver uses Scheduled Mode.

If no RUNMODE statement or mode-related command line parameter is present, the Platform Receiver uses *persistent* Mode.

For more information about Platform Receiver modes of operation, see [“Modes of Operation” on page 109](#).

Example:

```
RUNMODE polling
```

10.3.30 SECURITY Statement

z/OS only.

The SECURITY statement specifies the type of security system in use. ASCLIENT attempts to determine the security system type by examining the subsystem table. RACF and CA ACF2 normally have subsystem table entries, but CA Top Secret does not.

If ASCLIENT cannot determine the security system that is in use, it writes a message to the log and disables password migration.

Syntax:

```
SECURITY SecuritySystemType
```

SecuritySystemType must be ACF2, RACF, or TSS.

Example:

```
SECURITY TSS
```

10.3.31 SMF Statement

z/OS only.

The SMF statement specifies the SMF record type for the z/OS platform's performance statistics record. If the SMF statement is not present, the z/OS platform does not produce SMF records.

Syntax:

```
SMF RecordType
```

RecordType must be an integer between 128 and 255.

Example:

```
SMF 240
```

10.3.32 SYSLOGFACILITY Statement

Linux/UNIX only.

The SYSLOGFACILITY statement specifies the SYSLOG facility name to use for message logging on Linux/UNIX systems.

Syntax:

```
SYSLOGFACILITY FacilityName
```

FacilityName specifies the SYSLOG facility to use for logging messages. The possible values for a specific Linux/UNIX system are mapped by the syslog.h file of that particular system.

If no SYSLOGFACILITY statement is coded, the default value is LOG_DAEMON.

Example:

```
SYSLOGFACILITY LOG_DAEMON
```

10.3.33 TRACEFILE Statement

The TRACEFILE statement specifies that debugging output is generated and the file path where it is written. If the TRACEFILE statement is present in the platform configuration file, full debugging output is generated even if the -d command line parameter is not present.

To obtain debugging output from the system intercepts when you use the DIRECTTOAUTHENTICATION statement, you must use either the TRACEFILE or the TRACETOSTDOUT statement.

The TRACEFILE statement is not available in z/OS.

Syntax:

```
TRACEFILE FilePath
```

FilePath specifies the location in the file system where debugging output is written.

Example:

```
TRACEFILE c:\novell\asam\debug.txt
```

10.3.34 TRACETOSTDOUT Statement

The TRACETOSTDOUT statement specifies that debugging output is generated and that it is written to the standard output channel stdout. If the TRACETOSTDOUT statement is present in the platform configuration file, full debugging output is generated even if the -d command line parameter is not present.

To obtain debugging output from the system intercepts when you use the DIRECTTOAUTHENTICATION statement, you must use either the TRACEFILE or the TRACETOSTDOUT statement.

The TRACETOSTDOUT statement is not available in z/OS.

Syntax:

```
TRACETOSTDOUT
```

Example:

```
TRACETOSTDOUT
```

10.3.35 UPDATEPASSWORD Statement

Linux/UNIX only.

The UPDATEPASSWORD statement specifies that the driver updates the local security system upon a successful check password or change password operation, or when password replication information is received from the Core Driver. This allows a user to log in using the last password that worked on the system if the driver, eDirectory, or the network is not available, and the local security system is appropriately configured.

If there is no UPDATEPASSWORD statement present in the platform configuration file, the driver does not store passwords in the local security system.

Syntax:

```
UPDATEPASSWORD
```

Example:

```
UPDATEPASSWORD
```

10.3.36 UPDATESAMBA Statement

Linux/UNIX only.

The UPDATESAMBA statement specifies that the driver updates the Samba password file upon a successful check password or change password operation, or when password replication information is received from the Core Driver.

If there is no UPDATESAMBA statement present in the platform configuration file, the driver does not store passwords in the Samba password file.

Syntax:

UPDATESAMBA *FilePath*

FilePath specifies the location of the smbpasswd program in file system.

Example:

```
UPDATESAMBA /usr/local/samba/bin/smbpasswd
```

10.3.37 USEFILEIPC Statement

Linux/UNIX only.

The USEFILEIPC statement specifies that data from the Platform Receiver should be made accessible by the Platform Receiver scripts by using file Input/Output rather than environment variables.

While environment variables are a convenient method for communicating data between the Platform Receiver and the scripts, they have size limitations, based on the operating system parameter ARG_MAX, that make it impossible to process events when they become too large. For example, a group that contains many users could exceed this size limit. The value of ARG_MAX varies on different operating systems.

Syntax:

```
USEFILEIPC
```

Example:

```
USEFILEIPC
```

10.4 Using Include and Exclude Configuration Statements

The various Include and Exclude statements can be used in the platform configuration file to determine which users are authenticated through Platform Services and which users are authenticated locally, and which users and groups are managed based on provisioning events and which users and groups are managed locally.

These statements allow the use of masking characters to specify a mask that can match more than one user ID or group.

For details about each Include and Exclude statement, see the corresponding statement description.

- ◆ [“AM.GROUP.INCLUDE Statement / AM.GROUP.EXCLUDE Statement” on page 122](#)
- ◆ [“AM.USER.INCLUDE Statement / AM.USER.EXCLUDE Statement” on page 123](#)
- ◆ [“AS.USER.INCLUDE Statement / AS.USER.EXCLUDE Statement” on page 124](#)

Certain special users and groups are always processed locally unless you specify the IGNORESTANDARDEXCLUDES statement. For more information about this statement, see [“IGNORESTANDARDEXCLUDES Statement” on page 128](#). For a list of the users and groups in the standard exclude list, see [Section 8.10, “Standard Exclude List,” on page 112](#).

10.4.1 Mask Characters and Examples

You can use masks to match more than one user ID or group in Include and Exclude statements. The following tables list mask characters and provide examples of masks.

Table 10-3 Mask Characters

Mask Character	Matches
*	Any string of zero or more characters. The asterisk (*) mask character can only be used at the end of a mask.
%	Any single character
?	Any single character
\?	Any single character
\a	A single alphabetic character
\n	A single numeric character
\x	A single alphanumeric character
\s	A single @, #, \$, or other OS-dependent non-alphanumeric special character

Table 10-4 Example Masks

Mask	Matches
Z*	ZEBRA ZULU ZED ZABRZE Z9
Zn*	Z9 Z9WWW
\s\alaln\?	#BB29 #BB2A #AB9_
\aFF	AFF BFF CFF DFF EFF
*	All strings
%%%%%%%%%	All five-character strings
?????	All five-character strings
\?\?\?\?\?	All five-character strings

10.4.2 Rules by Which Masks Are Matched Against User IDs and Groups

- ◆ The order in which INCLUDE and EXCLUDE statements are specified does not matter.
- ◆ If more than one mask matches a given user ID or group, the most specific mask is used.
- ◆ The mask is case-insensitive.
- ◆ Specifying the same mask on both an INCLUDE and an EXCLUDE statement is a syntax error.

- ◆ Unless EXCLUDE * is coded, INCLUDE * is assumed for each statement type. Certain special users and groups are excluded unless the IGNORESTANDARDEXCLUDES statement is specified. For details, see [“IGNORESTANDARDEXCLUDES Statement” on page 128](#).
- ◆ Do not code both an INCLUDE * statement and an EXCLUDE * statement of the same type.

Platform Services Administration

IV

Part IV provides you with information you need to plan for deploying the i5/OS (OS/400) Platform Services of the Novell® Identity Manager Fan-Out driver. It includes the following chapters:

- ♦ Chapter 11, “Installing Platform Services,” on page 141
- ♦ Chapter 12, “Configuring and Administering Platform Services,” on page 145
- ♦ Chapter 13, “Troubleshooting Platform Services,” on page 149

Installing Platform Services

11

The installation and setup of Novell® Identity Manager Fan-Out driver Platform Services includes tasks performed on the platform and the Core Driver. This section describes the installation tasks that are performed on the platform system. For details about platform configuration and administration tasks, see [Chapter 12, “Configuring and Administering Platform Services,” on page 145](#).

The Core Driver tasks include defining UID/GID Sets, defining Platform Sets, and defining Platform objects. These tasks must be completed before you can use Platform Services. For more information about these tasks, see [Part II, “Core Driver Administration,” on page 37](#).

After the planning process has been completed, installation of Platform Services for i5/OS by experienced system programmers familiar with the local environment and the Identity Manager Fan-Out driver should take about half an hour.

Topics in this section include

- ♦ [Section 11.1, “About Platform Services for i5/OS,” on page 141](#)
- ♦ [Section 11.2, “Platform Services Installation Procedure,” on page 143](#)
- ♦ [Section 11.3, “Uninstalling Platform Services,” on page 143](#)

11.1 About Platform Services for i5/OS

Platform Services for i5/OS consists of two major components.

- ♦ **System Intercept:** The System Intercept provides password change information to the Core Driver. The System Intercept is implemented through the Password Validation Program Exit specified with the QPWDVLDPGM system value.
- ♦ **Platform Receiver:** The Platform Receiver requests provisioning events from Event Journal Services and runs a Receiver script to carry out the appropriate action for each event as it is received.

11.1.1 The System Intercept

The driver must be informed of changes made to passwords in order to support password replication. The System Intercept for i5/OS provides information to the Core Driver about password changes made on the platform. (Information about password changes to a user in eDirectory™ are received by the i5/OS platform as provisioning events and are processed by the Platform Receiver.)

The System Intercept must be configured to connect directly to Core Drivers using the `DIRECTTOAUTHENTICATION` statement in the platform configuration file. For details about the platform configuration file, see [Chapter 10, “The Platform Configuration File,” on page 119](#).

11.1.2 The Platform Receiver

The Platform Receiver processes provisioning events received from the Event Journal Services component of the Core Driver.

The Platform Receiver communicates with Event Journal Services using Secure Sockets Layer (SSL). Data is encoded using UTF-8, which is converted to EBCDIC.

Run the Platform Receiver on a schedule that is appropriate for your requirements. For details about Platform Receiver operation, see [Part III, “Platform Services Planning,” on page 103](#).

The Platform Receiver reads its configuration information from `ASAM/data/asamplat.conf`, the platform configuration file. For details about the platform configuration file, see [Part III, “Platform Services Planning,” on page 103](#).

The i5/OS Platform Receiver uses the Attribute Name Mapping file, `/usr/local/ASAM/data/attrmap.conf`, to convert attribute names obtained from Event Journal Services to the Profile and System Distribution Directory field names for use by the Receiver scripts. For more information about the Attribute Name Mapping file, see [“The Attribute Name Mapping File” on page 146](#).

The i5/OS Platform Receiver logs messages to the standard joblog facility.

11.1.3 Receiver Scripts

Receiver scripts for i5/OS platforms are implemented as Control Language (CL) programs. The Platform Receiver runs the programs from the ASAM library.

Provisioning events are received as groupings of name-value pairs as shown in the following example:

```
enterpriseUserName bob
```

The Platform Receiver calls a Receiver script whenever it is necessary to obtain information about users or groups on the platform and whenever it is appropriate to take an action for a user or group on the platform.

Processing Summary

1. When the Platform Receiver calls a Receiver script, it maps the name-value pairs and stores them in a user space. Procedures are provided for setting and retrieving these values.

User names and group names are checked for validity before they are mapped. A utility Receiver script is called to perform the validity checking.
2. Receiver scripts are called as appropriate to determine group affiliations for user events and group membership for group events.
3. Receiver scripts are called to take the necessary actions.

For more information about Receiver scripts, see [Section 8.9, “Receiver Scripts,” on page 111](#) and the scripts themselves.

11.1.4 Authentication Services

Authentication Services for i5/OS does not redirect authentication requests to eDirectory, but instead replicates passwords between the i5/OS system and eDirectory.

When a password is changed on the i5/OS system, the System Intercept sends a change password notification to a Core Driver for processing.

When a password for a user associated with an i5/OS system is changed in

, a provisioning event is generated by the Core Driver and given to the Platform Receiver for processing. By default, the Core Driver converts passwords to lowercase before sending them to the Platform Receiver. For more information about password case, see “[Lower Password Case](#)” on [page 83](#).

Because password replication information travels in both directions, it is affected by the Include/Exclude lists of both Authentication Services and Identity Provisioning. It is important therefore, to configure both sets of Include/Exclude lists symmetrically.

11.2 Platform Services Installation Procedure

For step-by-step instructions about installing Platform Services, see the *Fan-Out Driver Installation Quick Start for Midrange Systems* at the [Identity Manager 3.6 Driver Documentation Web site \(http://www.novell.com/documentation/idm36drivers\)](http://www.novell.com/documentation/idm36drivers).

11.3 Uninstalling Platform Services

To remove Platform Services from an i5/OS system:

- 1 Stop the Platform Receiver.
For details, see “[Starting and Stopping the Platform Receiver](#)” on [page 145](#).
- 2 Remove the Platform Receiver from any system startup, shutdown, and scheduling procedures as appropriate.
- 3 In the Web interface, remove the Platform object for the i5/OS system.
- 4 Remove ASAMPWD from the QPWDVLDPGM system value.
- 5 Remove the ASAM library from your library list.
- 6 Remove the ASAM library and `/usr/local/ASAM` directory created by Platform Services installation.

Configuring and Administering Platform Services

12

After you have installed Novell® Identity Manager Fan-Out driver Platform Services, use the information in this section for configuration and administration.

- [Section 12.1, “Platform Certificate Management,” on page 145](#)
- [Section 12.2, “Administering Platform Services,” on page 145](#)

12.1 Platform Certificate Management

Connections between Platform Services and Core Drivers use Secure Sockets Layer (SSL). SSL connections are authenticated through the use of certificates.

The certificates used by the Identity Manager Fan-Out driver are minted by the Certificate Services component of the Core Driver. When you install and configure Platform Services, you obtain a certificate.

To obtain a new certificate for your platform, run the Platform Receiver with the `MODE(*SECURE)` command line parameter.

Platform certificates are stored in the `ASAM/data/platformservices/certs` directory. Ensure that access to the `certs` directory is limited to the appropriate users.

12.2 Administering Platform Services

The Platform Receiver obtains provisioning events from Event Journal Services and calls the appropriate Receiver script to process the given type of event. For more information about Receiver scripts, see [“Receiver Scripts” on page 142](#).

The Platform Receiver must be running if you plan to use Identity Provisioning on the platform.

12.2.1 Starting and Stopping the Platform Receiver

Set up routine operation of the i5/OS Platform Receiver using Persistent Mode or Polling Mode, in a subsystem as an autostart entry. For information, see [Section 8.8.1, “Modes of Operation,” on page 109](#).

To start and stop the Platform Receiver, execute `GO ASAM` to load the ASAM menu, and select the desired option. The ASAM library must be in your library list.

To start the Platform Receiver at the command line, enter `ASAMRCVR`, specifying command line parameters as appropriate. The ASAM library must be in your library list.

12.2.2 Platform Receiver Command Line Parameters

Table 12-1 Platform Receiver Command Line Parameters

Option	Argument	Explanation
CONFIG	Configuration File Path	Specifies the platform configuration file to use. If you do not specify this option, the default is /usr/local/ASAM/data/asamplat.conf.
MODE	*POLL	The Platform Receiver uses Polling Mode.
	*PERSIST	The Platform Receiver uses Persistent Mode.
	*FULL	The Platform Receiver uses Full Sync Mode.
	*SCHED	The Platform Receiver uses Scheduled Mode.
	*SECURE	Obtain a security certificate for the Platform and end. This is needed only during the initial configuration process.
CHECK	*ON	The Platform Receiver uses Check Mode.

The MODE option determines the mode of operation for the Platform Receiver. If this option is not specified, the mode of operation specified by the RUNMODE statement in the platform configuration file is used. If there is no RUNMODE statement, the Platform Receiver uses Persistent Mode.

For details about the Platform Receiver modes of operation, see [Section 8.8.1, “Modes of Operation,” on page 109](#).

12.2.3 Maintaining Files Used by the Platform Receiver

Maintenance involves four types of files.

- ◆ [“The Attribute Name Mapping File” on page 146](#)
- ◆ [“The Platform Configuration File” on page 147](#)
- ◆ [“Receiver Scripts” on page 147](#)
- ◆ [“Log Files” on page 147](#)

The Attribute Name Mapping File

The i5/OS Platform Receiver uses the Attribute Name Mapping file to convert attribute names obtained from Event Journal Services to Profile and System Distribution Directory field names for use by the Receiver scripts. The attribute names used by Event Journal Services are the names that appear in the Identity Manager Subscriber filter.

The Attribute Name Mapping file is /usr/local/ASAM/data/attrmap.conf.

Each line of the file contains the name of an attribute received from Event Journal Services, a comma, and the name to which the attribute is to be mapped for use by Receiver scripts.

Any line in the file beginning with an octothorpe (#) is a comment. Blank lines are ignored.

For more information about provisioning attributes to your platform, see [Appendix A, “OS/400 Provisioning Tips,”](#) on page 211.

The Platform Configuration File

The Platform Receiver reads the platform configuration file to locate the Core Driver, to determine which users and groups are managed using provisioning events, and to find other configuration information. For more information, see [Chapter 10, “The Platform Configuration File,”](#) on page 119.

Receiver Scripts

Receiver scripts for i5/OS platforms are implemented as Control Language (CL) programs. The Platform Receiver runs the Receiver scripts from the ASAM library. The source code for the Receiver scripts is stored in the `SCRIPTS` file in the ASAM library. Several varieties of Receiver scripts are placed in subdirectories of `asam/bin/platformservices/platformreceiver/scripts` by the installation process.

Log Files

The i5/OS Platform Receiver writes messages to standard i5/OS joblogs. Use `DSPJOBLOG` or iSeries* Navigator to view these logs. Log messages are documented in the *Messages Reference*.

Troubleshooting Platform Services

13

Novell® Identity Manager Fan-Out driver components record messages to their Audit Log, Operational Log, and their host system log. Examining these should be foremost in your troubleshooting efforts.

The Audit and Operational logs of Core Driver components are maintained in their logs directory.

The i5/OS Platform Receiver writes log messages to the standard i5/OS joblog facilities.

By its very nature, the Fan-Out driver is highly dependent upon the proper operation of your network and eDirectory™. If you are having problems with the driver, ensure that the various driver components are able to communicate with one another and that eDirectory is functioning properly.

For information pertaining to Fan-Out driver performance issues, see [Chapter 4, “Core Driver Planning,”](#) on page 39.

IMPORTANT: Make sure you upgrade the driver, including all of your platforms, when new versions or support packs become available.

13.1 Obtaining Debugging Output

Identity Manager Fan-Out driver components support the option to produce extensive debugging output. Although this output is intended primarily for use by Novell Technical Support, you might find it useful for your own troubleshooting efforts.

Because debugging mode adversely affects performance, it should not be used for routine operations.

To obtain debugging output for the Platform Receiver on i5/OS:

- 1 Add a `DEBUGLOGFILE` statement or `DEBUGTOSTDOUT` statement to the platform configuration file.

For information, see [Chapter 10, “The Platform Configuration File,”](#) on page 119.

- 2 Specify the debugging command line parameter when you start the Platform Receiver.

To obtain full debugging output, specify `DEBUG (*)` on the command line.

To obtain debugging output limited to messages exchanged with Core Drivers, specify `DEBUG (dom)`.

13.2 Troubleshooting Identity Provisioning

Ensure that user and group names conform to the character set and length restrictions imposed by the platform operating system.

Identity Provisioning information for platforms that use password replication is not normally available unless password information is available. For example, if you have just installed and configured the Identity Manager Fan-Out driver for the first time, and you run the Platform Receiver

in Full Sync Mode on a system whose Platform object specifies Permit Password Replication, no accounts are created there. You must install the password intercepts, and users must authenticate through the driver or change their passwords so that password replication information is available. Then that account information becomes available to the platform.

13.3 Troubleshooting Network Issues

Although the details of network troubleshooting are beyond the scope of this document and depend on a number of factors particular to your environment, the purpose of this section is to determine if the various Identity Manager Fan-Out driver components can communicate with one another.

To verify IP connections between driver platforms and Core Drivers using the ping command. From a command prompt on z/OS, i5/OS, UNIX, or Windows, enter `ping ipaddr`, where *ipaddr* is the IP address of the remote computer.

If your installation uses router filters to prevent the use of ping, consult with those responsible for managing your network for information on how to verify connectivity.

API Development



Part V describes the Authentication Services (AS) Client application programming interface (API) of the Novell[®] Identity Manager Fan-Out Driver. It includes the following chapters:

- ♦ [Chapter 14, “About the API,” on page 153](#)
- ♦ [Chapter 15, “C Language API Reference,” on page 157](#)
- ♦ [Chapter 16, “Java Language API Reference,” on page 193](#)
- ♦ [Chapter 17, “API Examples,” on page 207](#)

Novell® Identity Manager Fan-Out Driver platforms provide an Authentication Services (AS) application programming interface (API) that can be used by applications to access eDirectory™. This API is compatible with the AS Client API that was provided in the NDS® Authentication Services and the Account Management 3.0 products. To use this API, you must obtain and install the Identity Manager Fan-Out Driver.

The platform configuration file provides the information necessary for establishing communications with a Core Driver. For details about the platform configuration file, see [Chapter 10, “The Platform Configuration File,” on page 119](#).

In the C language environment, a call must be made to `ASC_INIT()` or `ASC_INIT_EXT()` to process the platform configuration file and initialize the environment before API calls can be made to the Core Driver. The header file `ascauth.h` provides the function prototypes for the API calls and their return value definitions.

In the Java environment, a call must be made to the `INIT()` method to process the platform configuration file and initialize the environment before API calls can be made to the Core Driver. Class `com.novell.asam.JAscAuth.JAscAuth` provides the methods used to call the API.

Details about platforms/environments with which you use the API are provided in earlier sections of this guide. Also be aware that this guide is one of three available administration guides for the Fan-Out Driver, each tailored to the range of platforms with which it can work:

- ♦ *Identity Manager Fan-Out Driver for Linux and UNIX Administration Guide*
- ♦ *Identity Manager Fan-Out Driver for Mainframes Administration Guide (z/OS)*
- ♦ *Identity Manager Fan-Out Driver for Midrange Administration Guide (OS/400, i5/OS)*

Topics in this section are

- ♦ [Section 14.1, “Using the API in the z/OS Environment,” on page 153](#)
- ♦ [Section 14.2, “Using the API in the OS/400 Environment,” on page 154](#)
- ♦ [Section 14.3, “Using the API in the Linux/UNIX Environment,” on page 154](#)
- ♦ [Section 14.4, “API Function List,” on page 154](#)

14.1 Using the API in the z/OS Environment

Access to the API in the z/OS environment uses the ASCLIENT started task. The caller must be APF authorized. The C header file is located in the z/OS Platform Services Samples Library member `ASCAUTH`. When link-editing applications that call the API, include the z/OS Platform Services Load Library in your `SYSLIB` concatenation. Calls to `ASC_INIT()` and `ASC_INIT_EXT()` must be made with the `NULL` parameter in place of the configuration file name, because ASCLIENT processes the platform configuration file for the z/OS environment.

For additional information about the z/OS platform, see the *Identity Manager Fan-Out Driver for Mainframes Administration Guide*.

14.2 Using the API in the OS/400 Environment

Access to the API using C in the OS/400 environment is provided by the `libascauth *SRVPGM` in the ASAM library. The C header file is located in `/usr/local/ASAM/bin`.

The caller must have read privileges to the `/usr/local/ASAM/data/PlatformServices/certs` directory.

For additional information about the OS/400 platform, see the *Identity Manager Fan-Out Driver for Midrange Administration Guide*.

14.3 Using the API in the Linux/UNIX Environment

Access to the API using C in the Linux/UNIX environment is through calls to the shared library. The shared library and the C header file `ascauth.h` are copied to system-specific directories during the Linux/UNIX Platform Services installation process.

Access to the API using Java is through calls to the methods of class `com.novell.asam.JAscAuth.JAscAuth`. The `jascauth.jar` file is copied to the `ASAM/bin/PlatformServices/PlatformClient/Java` directory during Platform Services installation.

The caller must have read access to the `/usr/local/ASAM/data/PlatformServices/certs` directory.

For additional information about the Linux/UNIX platform, see the *Identity Manager Fan-Out Driver for Linux and UNIX Administration Guide*.

14.4 API Function List

API routines are provided to perform the following functions:

- ◆ Initialize the environment.
 - ◆ C: “`ASC_INIT`” on page 173, “`ASC_INIT_EXT`” on page 175
 - ◆ Java: “`init`” on page 198
- ◆ Terminate the environment.
 - ◆ C: “`ASC_TERM`” on page 189
 - ◆ Java: “`destroy`” on page 196
- ◆ Validate a user ID and password combination.
 - ◆ C: “`ASC_CHKPASSWD`” on page 163
 - ◆ Java: “`checkPassword`” on page 196
- ◆ Change a user's password, given the current password.
 - ◆ C: “`ASC_CHGPASSWD`” on page 161
 - ◆ Java: “`changePassword`” on page 195

- ♦ Reset a user's password as an administrative user.
 - ♦ **C:** “ASC_ADMINRSTPASSWD” on page 158
 - ♦ **Java:** “adminResetPassword” on page 195
- ♦ Obtain the fully distinguished name for a user ID.
 - ♦ **C:** “ASC_GETCONTEXT” on page 167
 - ♦ **Java:** “getContext” on page 197
- ♦ Determine if a user has security equal to a given object.
 - ♦ **C:** “ASC_SECEQUAL” on page 186
 - ♦ **Java:** “securityEquals” on page 199
- ♦ Determine if an object has the specified effective rights to the specified attribute of another object.
 - ♦ **C:** “ASC_RIGHTS” on page 183
 - ♦ **Java:** “effectiveRights” on page 197
- ♦ Obtain a list of members of a group.
 - ♦ **C:** “ASC_GRPMEM” on page 170
 - ♦ **Java:** “groupMembers” on page 198
- ♦ Obtain a list of security equivalences for a user.
 - ♦ **C:** “ASC_LISTSEQV” on page 177
 - ♦ **Java:** “listSecurityEquivalences” on page 198
- ♦ Obtain attribute values for an object.
 - ♦ **C:** “ASC_READATTR” on page 180
 - ♦ **Java:** “readAttribute” on page 199
- ♦ Determine if a given user is in the Include/Exclude list.
 - ♦ **C:** “ASC_USER_INCLUDE_EXCLUDE” on page 191
 - ♦ **Java:** “userIncludeExclude” on page 200
- ♦ Decode API return values.
 - ♦ **C:** “ASC_STRERROR” on page 188
 - ♦ **Java:** “strError” on page 200
- ♦ Convert number of seconds to number of days.
 - ♦ **C:** “ASC_DAYS” on page 166
 - ♦ **Java:** “secondsToDays” on page 199

This section presents all AS Client API functions available in the C programming language for the Novell® Identity Manager Fan-Out Driver. Information for each function includes syntax, parameters, return values and an example of the function as applied in code.

The C functions for the Fan-Out Driver API include the following:

- ◆ [“ASC_ADMINRSTPASSWD” on page 158](#)
- ◆ [“ASC_CHGPASSWD” on page 161](#)
- ◆ [“ASC_CHKPASSWD” on page 163](#)
- ◆ [“ASC_DAYS” on page 166](#)
- ◆ [“ASC_GETCONTEXT” on page 167](#)
- ◆ [“ASC_GRPMEM” on page 170](#)
- ◆ [“ASC_INIT” on page 173](#)
- ◆ [“ASC_INIT_EXT” on page 175](#)
- ◆ [“ASC_LISTSEQV” on page 177](#)
- ◆ [“ASC_READATTR” on page 180](#)
- ◆ [“ASC_RIGHTS” on page 183](#)
- ◆ [“ASC_SECEQUAL” on page 186](#)
- ◆ [“ASC_STRERROR” on page 188](#)
- ◆ [“ASC_TERM” on page 189](#)
- ◆ [“ASC_USER_INCLUDE_EXCLUDE” on page 191](#)

ASC_ADMINRSTPASSWD

Performs an administrative reset of a user's password. The new password is marked as being expired unless it is non-expiring.

Syntax

```
#include <ascauth.h>
int ASC_ADMINRSTPASSWD(ASCENV *asce, char *adminUser, char
*adminPassword, char *user, char *newpass);
```

Parameters

asce	The environment item returned from the call to ASC_INIT() or ASC_INIT_EXT().
adminUser	The Enterprise User ID of an administrative user with rights to change the target user's password.
adminPassword	The password of the administrative user ID.
user	The Enterprise User ID whose password is to be changed.
newpass	The new password for the user.

Return Values

Returns one of the following integer values defined in `ascauth.h`:

AS_OK	Password changed
AS_NOUSER	User inactive or not found in the Census
AS_BADCLIENT	Local host not authorized to query the Core Driver
AS_NOAGENT	No Core Driver could be contacted
AS_NOAUTHENV	No environment has been established
AS_INVALIDREQ	Call rejected by the Core Driver as not valid or not supported
AS_INVALIDARGS	Invalid arguments supplied to the function
AS_KEYEXPIRED	Old key rejected by the Core Driver because the expiration date has passed
AS_INSUFFICIENTRIGHTS	Administrative user does not exist, administrative user does not have rights to change the password, or administrative user password not valid

Example

```
#include <stdio.h>
#include <stdlib.h>
#include <ascauth.h>

main(int argc, char *argv[])
{
    ASCENV *asce;
    ASCUSER ascu;
    char *adminUser, *adminPass, *user, *newpass;
    int rc;

    if (argc != 5) {
        fprintf(stderr, "usage: %s <adminUser> <adminPass> <user>
<newpass>\n",
                argv[0]);
        exit(EXIT_FAILURE);
    }

    adminUser = argv[1];
    adminPass = argv[2];
    user       = argv[3];
    newpass    = argv[4];

    /* initialize the authentication environment */
    asce = ASC_INIT(NULL);
    if (asce == NULL) {
        fprintf(stderr, "Error: cannot initialize authentication
environment\n");
        exit(EXIT_FAILURE);
    }

    /* change the user's password */
    rc = ASC_ADMINRSTPASSWD(asce, adminUser, adminPass, user, newpass);
    if (rc == AS_OK)
        printf("password has been changed\n");
    else if (rc == AS_NO)
        printf("password has not been changed\n");
    else
        printf("RC=%d, %s", rc, ASC_STRERROR(rc));

    /* now terminate the authentication environment */
    ASC_TERM(asce);
    return 0;
}
```

See Also

[“ASC_INIT” on page 173](#)

[“ASC_INIT_EXT” on page 175](#)

[“ASC_TERM” on page 189](#)

[“ASC_STRERROR” on page 188](#)

ASC_CHGPASSWD

Changes the password of a user.

Syntax

```
#include <ascauth.h>
int ASC_CHGPASSWD(ASCENV *asce, char *user, char *oldpass, char
*newpass);
```

Parameters

asce	The environment item returned from the call to ASC_INIT() or ASC_INIT_EXT().
user	The Enterprise User ID whose password is to be changed.
oldpass	The old password for the user.
newpass	The new password for the user.

Return Values

Returns one of the following integer values defined in `ascauth.h`:

AS_OK	Password changed
AS_NO	Old password is invalid
AS_NOUSER	User inactive or not found in the Census
AS_REVOKED	User's password is okay, but the user is disabled
AS_INTRUDER	Intruder lockout is enabled for this user
AS_PASSDUPLICATE	New password has been used previously
AS_PASSTOOSHORT	New password is too short
AS_BADCLIENT	Local host not authorized to query the Core Driver
AS_NOAGENT	No Core Driver could be contacted
AS_NOAUTHENV	No environment has been established
AS_INVALIDREQ	Call rejected by the Core Driver as not valid or not supported
AS_INVALIDARGS	Invalid arguments supplied to the function
AS_KEYEXPIRED	Old key rejected by the Core Driver because the expiration date has passed

Example

```
#include <stdio.h>
#include <stdlib.h>
#include <ascauth.h>

main(int argc, char *argv[])
{
    ASCENV *asce;
    ASCUSER ascu;
    char *user, *oldpass, *newpass;
    int rc;

    if (argc != 4) {
        fprintf(stderr, "usage: %s <user> <oldpass> <newpass>\n",
            argv[0]);
        exit(EXIT_FAILURE);
    }

    user    = argv[1];
    oldpass = argv[2];
    newpass = argv[3];

    /* initialize the authentication environment */
    asce = ASC_INIT(NULL);
    if (asce == NULL) {
        fprintf(stderr, "Error: cannot initialize authentication
environment\n");
        exit(EXIT_FAILURE);
    }

    /* change the user's password */
    rc = ASC_CHGPASSWD(asce, user, oldpass, newpass);
    if (rc == AS_OK)
        printf("password has been changed\n");
    else if (rc == AS_NO)
        printf("password has not been changed\n");
    else
        printf("RC=%d, %s", rc, ASC_STRERROR(rc));

    /* now terminate the authentication environment */
    ASC_TERM(asce);
    return 0;
}
```

See Also

[“ASC_INIT” on page 173](#)

[“ASC_INIT_EXT” on page 175](#)

[“ASC_TERM” on page 189](#)

[“ASC_STRERROR” on page 188](#)

ASC_CHKPASSWD

Verifies the password of a user.

Syntax

```
#include <ascauth.h>
int ASC_CHKPASSWD(ASCENV *asce, char *user, char *pass, ASCUSER
*ascu);
```

Parameters

asce	The environment item returned from the call to ASC_INIT() or ASC_INIT_EXT().
user	The Enterprise User ID to be checked.
pass	The password to be checked for the user.
ascu	The ASCUSER structure (defined in ascauth.h) to be filled in by ASC_CHKPASSWD() if the password is valid.

Return Values

Returns one of the following integer values defined in `ascauth.h`:

AS_OK	Password is okay
AS_NO	User ID/Password combination is invalid
AS_NOUSER	User inactive or not found in the Census
AS_REVOKED	User's password is okay, but the user is disabled
AS_INTRUDER	Intruder lockout enabled for this user
AS_BADCLIENT	Local host is not authorized to query the Core Driver
AS_NOAGENT	No Core Driver could be contacted
AS_NOAUTHENV	No environment has been established
AS_INVALIDREQ	Call rejected by the Core Driver as not valid or not supported
AS_INVALIDARGS	Invalid arguments supplied to the function
AS_KEYEXPIRED	Old key rejected by the Core Driver because the expiration date has passed

If an `AS_OK` return code is returned, the following fields in the `ASCUSER` structure contain additional information about the authenticated user:

<code><ascu>.pass.expire</code>	Number of seconds until the password expires (or -1 if the password does not expire)
---------------------------------------	--

<ascu>.pass.interval	Password change interval in seconds (or -1 if the password does not expire)
----------------------	---

If an AS_REVOKED code is returned, the following field in the ASCUSER structure contains additional information about the user:

<ascu>.login.disabled	User disabled flag
-----------------------	--------------------

Example

```
#include <stdio.h>
#include <stdlib.h>
#include <ascauth.h>

main(int argc, char *argv[])
{
    ASCENV *asce;
    ASCUSER ascu;
    char *user, *pass;
    int rc;

    if (argc != 3) {
        fprintf(stderr, "usage: %s <user> <password>\n", argv[0]);
        exit(EXIT_FAILURE);
    }

    user = argv[1];
    pass = argv[2];

    /* initialize the authentication environment */
    asce = ASC_INIT(NULL);
    if (asce == NULL) {
        fprintf(stderr, "Error: cannot initialize authentication
environment\n");
        exit(EXIT_FAILURE);
    }

    /* check the user's password */
    rc = ASC_CHKPASSWD(asce, user, pass, &ascu);
    if (rc == AS_OK)
        printf("password ok\n");
    else if (rc == AS_NO)
        printf("password invalid\n");
    else
        printf("RC=%d, %s", rc, ASC_STRERROR(rc));

    /* now terminate the authentication environment */
    ASC_TERM(asce);
    return 0;
}
```

See Also

[“ASC_INIT” on page 173](#)

[“ASC_INIT_EXT” on page 175](#)

[“ASC_TERM” on page 189](#)

[“ASC_STRERROR” on page 188](#)

ASC_DAYS

Converts an integer number of seconds into an integer number of days.

Syntax

```
#include <ascauth.h>
long ASC_DAYS(long secs);
```

Parameters

secs	A number of seconds.
------	----------------------

Return Values

Returns the integer number of days corresponding to the given number of seconds.

Example

```
#include <stdio.h>
#include <stdlib.h>
#include <ascauth.h>

printf("*** CHKPASWD expire days=%ld, expire interval days=%ld\n",
        ASC_DAYS(ascu.pass.expire), ASC_DAYS(ascu.pass.interval));
```

ASC_GETCONTEXT

Obtains a user's fully distinguished object name from the Census and copies it into the buffer supplied by the caller.

Syntax

```
#include <ascauth.h>
int ASC_GETCONTEXT(ASCENV *asce, char *user, char *buffer, u_int
size);
```

Parameters

asce	The environment item returned from the call to ASC_INIT() or ASC_INIT_EXT().
user	The Enterprise User ID.
buffer	The buffer that is to receive the context. The result is truncated and the call returns AS_TOOSMALL if the buffer size cannot hold the entire result.
size	The length in bytes of the buffer.

Return Values

Returns one of the following integer values defined in `ascauth.h`:

AS_OK	Context was found
AS_NOUSER	User inactive or not found in the Census
AS_BADCLIENT	Local host not authorized to query the Core Driver
AS_NOAGENT	No Core Driver could be contacted
AS_NOAUTHENV	No environment has been established
AS_INVALIDREQ	Call rejected by the Core Driver as not valid or not supported
AS_INVALIDARGS	Invalid arguments supplied to the function
AS_TOOSMALL	Size of the pre-allocated buffer is too small-result truncated
AS_KEYEXPIRED	Old key rejected by the Core Driver because the expiration date has passed

Remarks

The buffer is padded with nulls if needed.

The format of the returned login context is the simple dot form. For example: `.jondoe.j.myorg`

Example

```
#include <stdio.h>
#include <stdlib.h>
#include <ascauth.h>

#define MAX_CONTEXT 512

main(int argc, char *argv[])
{
    ASCENV *asce;
    char *user, *context;
    int rc;

    if (argc != 2) {
        fprintf(stderr, "usage: %s <user>\n", argv[0]);
        exit(EXIT_FAILURE);
    }

    user = argv[1];

    /* allocate buffer */
    context = (char *) malloc(MAX_CONTEXT);

    /* initialize the authentication environment */
    asce = ASC_INIT(NULL);
    if (asce == NULL) {
        fprintf(stderr, "Error: cannot initialize authentication
environment\n");
        exit(EXIT_FAILURE);
    }

    /* get the user's context */
    rc = ASC_GETCONTEXT(asce, user, context, MAX_CONTEXT);
    if (rc == AS_OK)
        printf("context is %s\n", context);
    else
        printf("RC=%d, %s", rc, ASC_STRERROR(rc));

    free(context);

    /* now terminate the authentication environment */
    ASC_TERM(asce);
    return 0;
}
```

See Also

[“ASC_INIT” on page 173](#)

[“ASC_INIT_EXT” on page 175](#)

[“ASC_TERM” on page 189](#)

“ASC_STRERROR” on page 188

ASC_GRPMEM

Obtains a list of all members of the given group and places it in the buffer supplied by the caller.

Syntax

```
#include <ascauth.h>
int ASC_GRPMEM(ASCENV *asce, char *object, char *buf, u_int size);
```

Parameters

asce	The environment item returned from the call to ASC_INIT() or ASC_INIT_EXT().
object	The fully distinguished group name whose membership list is to be returned.
buf	The buffer in which the membership list is to be returned. Member names are separated by a new line '\n' character. The list is truncated and the call returns AS_TOOSMALL if the buffer size cannot hold the entire list.
size	The size of the buffer you provided.

Return Values

Returns one of the following integer values defined in `ascauth.h`:

AS_OK	Member list successfully returned
AS_BADCLIENT	Local host not authorized to query the Core Driver
AS_NOAGENT	No Core Driver could be contacted
AS_NOAUTHENV	No environment has been established
AS_INVALIDREQ	Call rejected by the Core Driver as not valid or not supported
AS_INVALIDARGS	Invalid arguments supplied to the function
AS_TOOSMALL	Size of the pre-allocated buffer is too small-list truncated
AS_INVALIDOBJ	Specified object does not exist
AS_KEYEXPIRED	Old key rejected by the Core Driver because the expiration date has passed

Remarks

The list is truncated, and the call returns `AS_TOOSMALL` if the buffer size cannot hold the entire list. You can retry with a larger buffer.

You can use `ASC_SECEQUAL` to see if an individual user is a member of a given group.

The groups a given user is a member of are included in the list returned by `ASC_LISTSEQV`.

Example

```
#include <stdio.h>
#include <stdlib.h>
#include <ascauth.h>

main(int argc, char *argv[])
{
    ASCENV *asce;
    char *group, buffer[2000];
    int rc;

    if (argc != 2) {
        fprintf(stderr, "usage: %s <group>\n", argv[0]);
        exit(EXIT_FAILURE);
    }

    group = argv[1];

    /* initialize the authentication environment */
    asce = ASC_INIT(NULL);
    if (asce == NULL) {
        fprintf(stderr, "Error: cannot initialize authentication
environment\n");
        exit(EXIT_FAILURE);
    }

    /* Get group membership info */
    rc = ASC_GRPMEM(asce, group, buffer, sizeof(buffer));
    if (rc == AS_OK)
        printf("Members of group %s:\n%s\n", group, buffer);
    else if (rc == AS_TOOSMALL) {
        printf("Members of group %s:\n%s\n", group, buffer);
        printf("** list was truncated because of lack of buffer space
**\n");
    }
    else
        printf("RC=%d, %s", rc, ASC_STRERROR(rc));

    /* now terminate the authentication environment */
    ASC_TERM(asce);
    return 0;
}
```

See Also

[“ASC_INIT” on page 173](#)

[“ASC_INIT_EXT” on page 175](#)

[“ASC_LISTSEQV” on page 177](#)

“ASC_SECEQUAL” on page 186

“ASC_TERM” on page 189

“ASC_STRERROR” on page 188

ASC_INIT

Reads the platform configuration file and initializes the environment so that calls can be made to a Core Driver. This function or `ASC_INIT_EXT()` must be called before any other API function.

Syntax

```
#include <ascauth.h>
ASCENV *ASC_INIT(char *filename);
```

Parameters

filename	The name of the platform configuration file. If you call <code>ASC_INIT()</code> with a <code>NULL</code> in place of the filename parameter as in <code>ASC_INIT(NULL)</code> , the default is as follows: z/OS: Always uses the ASCLIENT started task active configuration. OS/400: <code>/usr/local/ASAM/data/asamplat.conf</code> Linux/UNIX: <code>/usr/local/ASAM/data/asamplat.conf</code>
----------	--

Return Values

Returns a pointer to the API environment item created upon success. If an error has occurred, `NULL` is returned.

Example

```
#include <stdio.h>
#include <stdlib.h>
#include <ascauth.h>

main()
{
    ASCENV *asce;

    /* initialize the authentication environment */
    asce = ASC_INIT(NULL);
    if (asce == NULL) {
        fprintf(stderr, "Error: cannot initialize authentication
environment\n");
        exit(EXIT_FAILURE);
    }

    /* now you can make additional authentication calls here */

    /* now terminate the authentication environment */
```

```
ASC_TERM(asce);  
return 0;  
}
```

See Also

[“ASC_INIT_EXT” on page 175](#)

[“ASC_TERM” on page 189](#)

ASC_INIT_EXT

Reads the platform configuration file and initializes the environment so that calls can be made to a Core Driver. This function or ASC_INIT() must be called before any other API function.

ASC_INIT_EXT() differs from ASC_INIT() in that you can provide a buffer into which the API places error messages if the API environment cannot be initialized.

Syntax

```
#include <ascauth.h>
ASCENV *ASC_INIT_EXT(char *filename, char *error_msg, size_t size);
```

Parameters

filename	The name of the platform configuration file. If you call ASC_INIT_EXT() with a NULL in place of the filename parameter as in ASC_INIT_EXT(NULL, buffer, BUFSIZE), the default is as follows: z/OS: Always uses the ASCLIENT started task active configuration. OS/400: /usr/local/ASAM/data/asamplat.conf Linux/UNIX: /usr/local/ASAM/data/asamplat.conf
error_msg	A buffer you provide into which an error message can be placed if the environment cannot be initialized.
size	The size of the error_msg buffer you have provided.

Return Values

Returns a pointer to the environment item created upon success. If an error has occurred, NULL is returned, and a descriptive error message is placed into the error_msg buffer.

Example

```
#include <stdio.h>
#include <stdlib.h>
#include <ascauth.h>

#define BUFSIZE 256

main()
{
    ASCENV *asce;

    /* initialize the authentication environment */
    /* allocate buffer */
    buffer = (char *) malloc(BUFSIZE);
    asce = ASC_INIT_EXT(NULL, buffer, BUFSIZE);
```

```
if (asce == NULL) {
    fprintf(stderr, "Error: cannot initialize authentication
environment\n");
    fprintf(stderr, "  %s \n", buffer);
    exit(EXIT_FAILURE);
}

/* now you can make additional authentication calls here */

/* now terminate the authentication environment */
ASC_TERM(asce);
return 0;
}
```

See Also

[“ASC_INIT” on page 173](#)

[“ASC_TERM” on page 189](#)

ASC_LISTSEQV

Obtains a user's Security Equals attribute list and places it in the buffer supplied by the caller.

Syntax

```
#include <ascauth.h>
int ASC_LISTSEQV(ASCENV *asce, char *user, char *buf, u_int size);
```

Parameters

asce	The environment item returned from the call to ASC_INIT() or ASC_INIT_EXT().
user	The Enterprise User ID whose Security Equals list is to be returned.
buf	The buffer in which the security equivalence list is to be returned. Object names are separated by a new line '\n' character. The list is truncated and the call returns AS_TOOSMALL if the buffer size cannot hold the entire list.
size	The size of the buffer you provided.

Return Values

Returns one of the following integer values defined in `ascauth.h`:

AS_OK	Security equivalence list successfully returned
AS_NOUSER	User inactive or not found in the Census
AS_BADCLIENT	Local host is not authorized to query the Core Driver
AS_NOAGENT	No Core Driver could be contacted
AS_NOAUTHENV	No environment has been established
AS_INVALIDREQ	Call rejected by the Core Driver as not valid or not supported
AS_INVALIDARGS	Invalid arguments supplied to the function
AS_TOOSMALL	Size of pre-allocated buffer is too small-the list is truncated
AS_INVALIDOBJ	Specified object does not exist
AS_KEYEXPIRED	Old key rejected by the Core Driver because the expiration date has passed

Remarks

The list is truncated, and the call returns `AS_TOOSMALL` if the buffer size cannot hold the entire list. You can retry with a larger buffer.

Example

```
#include <stdio.h>
#include <stdlib.h>
#include <ascauth.h>

main(int argc, char *argv[])
{
    ASCENV *asce;
    char *object, buffer[2000];
    int rc;

    if (argc != 2) {
        fprintf(stderr, "usage: %s <object>\n", argv[0]);
        exit(EXIT_FAILURE);
    }

    object = argv[1];

    /* initialize the authentication environment */
    asce = ASC_INIT(NULL);
    if (asce == NULL) {
        fprintf(stderr, "Error: cannot initialize authentication
environment\n");
        exit(EXIT_FAILURE);
    }

    /* Get security equivalence info */
    rc = ASC_LISTSEQV(asce, object, buffer, sizeof(buffer));
    if (rc == AS_OK)
        printf("Security equivalences of object %s:\n%s\n", object,
buffer);
    else if (rc == AS_TOOSMALL) {
        printf("Security equivalences of object %s:\n%s\n", object,
buffer);
        printf("*** list was truncated because of lack of buffer space
**\n");
    }
    else
        printf("RC=%d, %s", rc, ASC_STRERROR(rc));

    /* now terminate the authentication environment */
    ASC_TERM(asce);
    return 0;
}
```

See Also

[“ASC_INIT” on page 173](#)

[“ASC_INIT_EXT” on page 175](#)

[“ASC_TERM” on page 189](#)

[“ASC_STRERROR” on page 188](#)

ASC_READATTR

Returns the value of the specified single-valued attribute for the specified object.

Syntax

```
#include <ascauth.h>
int ASC_READATTR(ASCENV *asce, char *object, char *attribute,
                 char *buffer, u_int bufsize);
```

Parameters

asce	The environment item returned from the call to ASC_INIT() or ASC_INIT_EXT().
object	The Enterprise User ID or fully distinguished object name of the object whose attribute value is to be returned.
attribute	The single-valued attribute whose value is to be returned for the object. Only the Home Directory attribute of a User object is supported at this time.
buffer	The buffer in which the object's attribute value is to be returned. The results are truncated and the call returns AS_TOOSMALL if the buffer size cannot hold the entire attribute value.
bufsize	The size of the buffer you provided.

Return Values

Returns one of the following integer values defined in `ascauth.h`:

AS_OK	Attribute value has been placed in the buffer successfully
AS_BADCLIENT	Local host not authorized to query the Core Driver
AS_ATTRNOTFOUND	Attribute does not exist for the specified object
AS_NOAGENT	No Core Driver could be contacted
AS_NOAUTHENV	No environment has been established
AS_INVALIDREQ	Call rejected by the Core Driver as not valid or not supported
AS_INVALIDARGS	Invalid arguments supplied to the function
AS_TOOSMALL	Size of the pre-allocated buffer is too small-results are truncated
AS_INVALIDOBJ	Specified object does not exist
AS_KEYEXPIRED	Old key rejected by the Core Driver because the expiration date has passed

Remarks

The results are truncated, and the call returns `AS_TOOSMALL` if the buffer size cannot hold the entire attribute value. You can retry with a larger buffer.

Limitations

Only the Home Directory attribute of a User object is supported at this time.

Example

```
#include <stdio.h>
#include <stdlib.h>
#include <ascauth.h>

main(int argc, char *argv[])
{
    ASCENV *asce;
    char *user, buffer[2000];
    int rc;

    if (argc != 2) {
        fprintf(stderr, "usage: %s <UserObjectFDN>\n", argv[0]);
        exit(EXIT_FAILURE);
    }

    user = argv[1];

    /* initialize the authentication environment */
    asce = ASC_INIT(NULL);
    if (asce == NULL) {
        fprintf(stderr, "Error: cannot initialize authentication
environment\n");
        exit(EXIT_FAILURE);
    }

    /* Get User object's home directory info */
    rc = ASC_READATTR(asce, user, "HOME DIRECTORY", buffer,
sizeof(buffer));
    if (rc == AS_OK)
        printf("Home Directory for User object %s:\n%s\n", user, buffer);
    else
        printf("RC=%d, %s", rc, ASC_STRERROR(rc));

    /* now terminate the authentication environment */
    ASC_TERM(asce);
    return 0;
}
```

See Also

[“ASC_INIT” on page 173](#)

“ASC_INIT_EXT” on page 175

“ASC_TERM” on page 189

“ASC_STRERROR” on page 188

ASC_RIGHTS

Checks the specified effective rights of one object over another for a specific attribute.

Syntax

```
#include <ascauth.h>
int ASC_RIGHTS(ASCENV *asce, char *obj1, char *obj2,
               char *attribute, char *rights);
```

Parameters

asce	The environment item returned from the call to ASC_INIT() or ASC_INIT_EXT().
obj1	The Enterprise User ID or fully distinguished object name whose effective rights are to be tested.
obj2	The Enterprise User ID or fully distinguished object name for which access by obj1 is to be tested.
attribute	The name of an attribute of obj2 for which the effective rights of obj1 are requested. The special attribute names All Attributes Rights, Entry Rights, and SMS Rights can also be specified.
rights	The rights to test. The characters specified must be in the following set: [S,C,R,W,A]. These correspond to Supervisor, Compare, Read, Write, and Add Self.

Return Values

Returns one of the following integer values defined in `ascauth.h`:

AS_OK	User or object has the specified rights to the specified object attribute
AS_NO	User or object does not have the specified rights to the specified object attribute
AS_ATTRNOTFOUND	Specified attribute could not be found
AS_INVALIDOBJ	Specified user not found in the Census or the specified object does not exist
AS_INVALIDOBJLEN	Specified object exceeds maximum length
AS_BADCLIENT	Local host not authorized to query the Core Driver
AS_NOAGENT	No Core Driver could be contacted
AS_NOAUTHENV	No environment has been established
AS_INVALIDREQ	Call rejected by the Core Driver as not valid or not supported
AS_INVALIDARGS	Invalid arguments supplied to the function

AS_KEYEXPIRED

Old key rejected by the Core Driver because the expiration date has passed

Example

```
#include <stdio.h>
#include <stdlib.h>
#include <ascauth.h>

main(int argc, char *argv[])
{
    ASCENV *asce;
    char *obj1, *obj2, *attr, *rights;
    int rc;

    if (argc != 5) {
        fprintf(stderr, "usage: %s <obj1> <obj2> \
            <attribute> <rights>\n", argv[0]);
        exit(EXIT_FAILURE);
    }

    obj1 = argv[1];
    obj2 = argv[2];
    attr = argv[3];
    rights = argv[4];

    /* initialize the authentication environment */
    asce = ASC_INIT(NULL);
    if (asce == NULL) {
        fprintf(stderr, "Error: cannot initialize authentication
environment\n");
        exit(EXIT_FAILURE);
    }

    /* check for rights */
    rc = ASC_RIGHTS(asce, obj1, obj2, attr, rights);
    if (rc == AS_OK)
        printf("User has rights\n");
    else
        printf("RC=%d, %s", rc, ASC_STRERROR(rc));

    /* now terminate the authentication environment */
    ASC_TERM(asce);
    return 0;
}
```

See Also

[“ASC_INIT” on page 173](#)

[“ASC_INIT_EXT” on page 175](#)

[“ASC_TERM” on page 189](#)

“ASC_STRERROR” on page 188

ASC_SECEQUAL

Checks to see if a user has security equivalence to the specified object.

Syntax

```
#include <ascauth.h>
int ASC_SECEQUAL(ASCENV *asce, char *user, char *object);
```

Parameters

asce	The environment item returned from the call to ASC_INIT() or ASC_INIT_EXT().
user	The Enterprise User ID to be tested.
Object	The fully distinguished object name to test the user for security equivalence.

Return Values

Returns one of the following integer values defined in `ascauth.h`:

AS_OK	User has security equivalence to the specified object
AS_NO	User does not have security equivalence to the object
AS_NOUSER	User inactive or not found in the Census
AS_BADCLIENT	Local host not authorized to query the Core Driver
AS_NOAGENT	No Core Driver could be contacted
AS_NOAUTHENV	No environment has been established
AS_INVALIDREQ	Call rejected by the Core Driver as not valid or not supported
AS_INVALIDARGS	Invalid arguments supplied to the function
AS_INVALIDOBJ	Specified object does not exist
AS_KEYEXPIRED	Old key rejected by the Core Driver because the expiration date has passed

Example

```
#include <stdio.h>
#include <stdlib.h>
#include <ascauth.h>

main(int argc, char *argv[])
{
    ASCENV *asce;
    char *user, *object;
```

```

int rc;

if (argc != 3) {
    fprintf(stderr, "usage: %s <user> <object>\n", argv[0]);
    exit(EXIT_FAILURE);
}
user = argv[1];
object = argv[2];

/* initialize the authentication environment */
asce = ASC_INIT(NULL);
if (asce == NULL) {
    fprintf(stderr, "Error: cannot initialize authentication
environment\n");
    exit(EXIT_FAILURE);
}

/* check for security equivalence */
rc = ASC_SECEQUAL(asce, user, object);
if (rc == AS_OK)
    printf("User has security equivalence\n");
else
    printf("RC=%d, %s", rc, ASC_STRERROR(rc));

/* now terminate the authentication environment */
ASC_TERM(asce);
return 0;
}

```

See Also

[“ASC_INIT” on page 173](#)

[“ASC_INIT_EXT” on page 175](#)

[“ASC_TERM” on page 189](#)

[“ASC_STRERROR” on page 188](#)

ASC_STRERROR

Returns the error string for the specified ASC function error code.

Syntax

```
#include <ascauth.h>
const char *ASC_STRERROR(int errnum);
```

Parameters

errnum	The error return value from a call to an ASC_ function.
--------	---

Return Values

Returns a static character string corresponding to the integer errnum value as defined in ascauth.h for ASC function error codes.

Example

```
#include <stdio.h>
#include <stdlib.h>
#include <ascauth.h>

rc = ASC_CHKPASSWD(asce, userid, password, &ascu);
strcpy(status, ASC_STRERROR(rc));
printf("\n*** CHKPASSWD return code = %d (%s)\n", rc, status);
```

ASC_TERM

Terminates and frees the environment that was created by a call to `ASC_INIT()` or `ASC_INIT_EXT()`. After the environment is terminated, no more calls to the Core Driver can be made without first issuing another `ASC_INIT()` or `ASC_INIT_EXT()` call.

Syntax

```
#include <ascauth.h>
void ASC_TERM(ASCENV *asce);
```

Parameters

<code>asce</code>	The environment item returned from the call to <code>ASC_INIT()</code> or <code>ASC_INIT_EXT()</code> .
-------------------	---

Return Values

No value is returned from this function.

Example

```
#include <stdio.h>
#include <stdlib.h>
#include <ascauth.h>

main()
{
    ASCENV *asce;

    /* initialize the authentication environment */
    asce = ASC_INIT(NULL);
    if (asce == NULL) {
        fprintf(stderr, "Error: cannot initialize authentication
environment\n");
        exit(EXIT_FAILURE);
    }

    /* now you can make additional authentication calls here */

    /* now terminate the authentication environment */
    ASC_TERM(asce);
    return 0;
}
```

See Also

[“ASC_INIT” on page 173](#)

“ASC_INIT_EXT” on page 175

ASC_USER_INCLUDE_EXCLUDE

Determines if a given user matches an AS.USER.INCLUDE or AS.USER.EXCLUDE statement in the platform configuration file.

Syntax

```
#include <ascauth.h>
int ASC_USER_INCLUDE_EXCLUDE(ASCENV *asce, char *user);
```

Parameters

asce	The environment item returned from the call to ASC_INIT() or ASC_INIT_EXT().
user	The Enterprise User ID of the user to be checked.

Return Values

Returns one of the following integer values defined in `ascauth.h`:

AS_NOMATCH	The user does not match any INCLUDE/EXCLUDE statement. Because "AS.USER.INCLUDE *" is implicit in the absence of AS.USER.EXCLUDE *, the user is included.
AS_INCLUDED	User matches an AS.USER.INCLUDE statement.
AS_EXCLUDED	User matches an AS.USER.EXCLUDE statement or an entry in the built-in standard exclude list.
AS_NOAUTHENV	No environment has been established.

Example

```
#include <stdio.h>
#include <stdlib.h>
#include <ascauth.h>

rc = ASC_USER_INCLUDE_EXCLUDE(asce, userid);
if (rc == AS_NOMATCH)
    printf("%s does not match an Include or Exclude statement\n",
userid);
else if (rc == AS_INCLUDED)
    printf("%s matches an Include statement\n", userid);
else if (rc == AS_EXCLUDED)
    printf("%s matches an Exclude statement\n", userid);
else
    printf("RC=%d, %s", rc, ASC_STRERROR(rc));
```

See Also

[“ASC_INIT” on page 173](#)

[“ASC_INIT_EXT” on page 175](#)

This section presents the AS Client API Java implementation for the Novell® Identity Manager Fan-Out Driver.

Descriptions of Java classes and methods include the following:

- ◆ “Class `com.novell.asam.JAscAuth.JAscAuth`” on page 194
 - ◆ “`adminResetPassword`” on page 195
 - ◆ “`changePassword`” on page 195
 - ◆ “`checkPassword`” on page 196
 - ◆ “`destroy`” on page 196
 - ◆ “`effectiveRights`” on page 197
 - ◆ “`getContext`” on page 197
 - ◆ “`getLastReturnCode`” on page 198
 - ◆ “`groupMembers`” on page 198
 - ◆ “`init`” on page 198
 - ◆ “`listSecurityEquivalences`” on page 198
 - ◆ “`readAttribute`” on page 199
 - ◆ “`secondsToDays`” on page 199
 - ◆ “`securityEquals`” on page 199
 - ◆ “`strError`” on page 200
 - ◆ “`userIncludeExclude`” on page 200
- ◆ “Classes Used by `checkPassword`” on page 202
- ◆ “Exception Classes in `com.novell.asam.JAscAuth`” on page 204

To view the reference documentation in JavaDoc format, see the `asam\bin\platformservices\platformclient\java\javadoc` directory on the platform system.

For code examples, see the `asam\bin\platformservices\platformclient\java` directory on the platform system.

Class com.novell.asam.JAscAuth.JAscAuth

Provides the methods you use to access the AS Client API.

Constructor

```
public JAscAuth()
```

Fields

The following fields map the AS Client API return codes. For more information about return codes from the AS Client API, see [Appendix C, “Troubleshooting the API,” on page 215](#).

```
public static int AS_OK = 0
public static int AS_NO = 1
public static int AS_NOUSER = 2
public static int AS_NOAGENT= 3
public static int AS_NOSERVER = 3
public static int AS_BADCLIENT= 4
public static int AS_REVOKED= 5
public static int AS_INTRUDER= 6
public static int AS_INVALIDARGS= 7
public static int AS_INVALIDOBJ= 8
public static int AS_INVALIDOBJLEN= 9
public static int AS_PASSDUPLICATE= 10
public static int AS_PASSTOOSHORT= 11
public static int AS_TOOSMALL= 12
public static int AS_ATTRNOTFOUND= 13
public static int AS_WSOCKUP= 14
public static int AS_WSOCKDOWN= 15
public static int AS_NOAUTHENV= 16
public static int AS_PRODUCTEXPIRED = 17
public static int AS_INCLUDED= 18
public static int AS_EXCLUDED= 19
public static int AS_NOMATCH= 20
public static int AS_NOLICENSE= 21
public static int AS_INVALIDREQ= 22
public static int AS_KEYEXPIRED= 23
```

Methods

The following methods invoke the API functions:

- ♦ [“adminResetPassword” on page 195](#)
- ♦ [“changePassword” on page 195](#)
- ♦ [“checkPassword” on page 196](#)
- ♦ [“destroy” on page 196](#)

- ◆ [“effectiveRights” on page 197](#)
- ◆ [“getContext” on page 197](#)
- ◆ [“getLastReturnCode” on page 198](#)
- ◆ [“groupMembers” on page 198](#)
- ◆ [“init” on page 198](#)
- ◆ [“listSecurityEquivalences” on page 198](#)
- ◆ [“readAttribute” on page 199](#)
- ◆ [“secondsToDays” on page 199](#)
- ◆ [“securityEquals” on page 199](#)
- ◆ [“strError” on page 200](#)
- ◆ [“userIncludeExclude” on page 200](#)

adminResetPassword

Performs an administrative reset of a user's password. The new password is marked as being expired unless it is non-expiring.

You must call the `init` method to initialize the JAsCAuth environment before calling `adminResetPassword`. For more information about `init`, see [“init” on page 198](#).

For details about the exceptions that can be thrown, see [“Exception Classes in com.novell.asam.JAsCAuth” on page 204](#).

Syntax

```
public void adminResetPassword(
    java.lang.String adminUser,
    java.lang.String adminPass,
    java.lang.String user,
    java.lang.String pass)
```

Parameters

<code>adminUser</code>	The Enterprise User ID of an administrative user with rights to change the target user's password
<code>adminPass</code>	The password of the administrative user
<code>user</code>	The Enterprise User ID whose password is to be changed
<code>pass</code>	The new password for the user

changePassword

Changes the password of a user.

You must call the `init` method to initialize the JAsCAuth environment before calling `changePassword`. For more information about `init`, see [“init” on page 198](#).

For details about the exceptions that can be thrown, see [“Exception Classes in com.novell.asam.JAsCAuth” on page 204](#).

Syntax

```
public void changePassword(  
    String user,  
    String oldPass,  
    String newPass)
```

Parameters

user	The Enterprise User ID whose password is to be changed
oldPass	The old password for the user
newPass	The new password for the user

checkPassword

Verifies the password of a user.

You must call the `init` method to initialize the JAscAuth environment before calling `checkPassword`. For more information about `init`, see [“init” on page 198](#).

The `checkPassword` method can optionally return information about the user and password in a JAscUser object. For details about the contents of JAscUser, see [“Classes Used by checkPassword” on page 202](#).

For details about the exceptions that can be thrown, see [“Exception Classes in com.novell.asam.JAscAuth” on page 204](#).

Syntax

```
public void checkPassword(  
    String user,  
    String pass)  
public void checkPassword(  
    String user,  
    String pass,  
    JAscUser ascuser)
```

Parameters

user	The Enterprise User ID whose password is to be verified
pass	The password to be verified for the user
ascuser	A JAscUser object to be filled with information about the user and password

destroy

Destroys the JAscAuth environment and frees its underlying resources.

Syntax

```
public void destroy()
```

See Also

[“init” on page 198](#)

effectiveRights

Checks the effective rights of one object over another for a specific attribute.

You must call the `init` method to initialize the `JAscAuth` environment before calling `effectiveRights`. For more information about `init`, see [“init” on page 198](#).

For details about the exceptions that can be thrown, see [“Exception Classes in `com.novell.asam.JAscAuth`” on page 204](#).

Syntax

```
public void effectiveRights(  
    String user,  
    String object,  
    String attribute,  
    String rights)
```

Parameters

user	The Enterprise User ID or fully distinguished object name whose effective rights are to be tested
object	The Enterprise User ID or fully distinguished object name for which access by user is to be tested
attribute	The name of an attribute of object for which the effective rights of user are tested. The special attribute names All Attributes Rights, Entry Rights, and SMS Rights can also be specified.
rights	The rights to test. The characters specified must be in the following set: [S,C,R,W,A]. These correspond to Supervisor, Compare, Read, Write, and Add Self.

getContext

Returns the fully distinguished object name from the Census for a given user.

You must call the `init` method to initialize the `JAscAuth` environment before calling `getContext`. For more information about `init`, see [“init” on page 198](#).

For details about the exceptions that can be thrown, see [“Exception Classes in `com.novell.asam.JAscAuth`” on page 204](#).

Syntax

```
public String getContext(String user)
```

Parameters

user	The Enterprise User ID whose context is to be returned
------	--

getLastReturnCode

Returns the return code from the last call to the AS Client API.

For details about return codes from the AS Client API, see [Appendix C, “Troubleshooting the API,” on page 215](#).

Syntax

```
public int getLastReturnCode()
```

See Also

[“strError” on page 200](#)

groupMembers

Returns an enumeration of all members of a given Group.

You must call the `init` method to initialize the JAsCAuth environment before calling `groupMembers`. For more information about `init`, see [“init” on page 198](#).

For details about the exceptions that can be thrown, see [“Exception Classes in com.novell.asam.JAsCAuth” on page 204](#).

Syntax

```
public Enumeration groupMembers(String group)
```

Parameters

group	The Enterprise Group or fully distinguished Group object name whose members are to be returned
-------	--

init

Initializes the JAsCAuth environment using the platform configuration file.

You can optionally specify the location of the platform configuration file to be used. If you do not specify the location of the platform configuration file, the default platform configuration file is used.

Call the `destroy` method to free the JAsCAuth environment and its underlying resources when you are finished. For more information about `destroy`, see [“destroy” on page 196](#).

Syntax

```
public void init()
public void init(java.lang.String filename)
```

Parameters

filename	The path name of the platform configuration file to use
----------	---

listSecurityEquivalences

Returns an enumeration of a given user's security equivalences.

You must call the `init` method to initialize the JAsCAuth environment before calling `listSecurityEquivalences`. For more information about `init`, see [“init” on page 198](#).

For details about the exceptions that can be thrown, see [“Exception Classes in `com.novell.asam.JAsCAuth`” on page 204](#).

Syntax

```
public Enumeration listSecurityEquivalences(String user)
```

Parameters

user	The Enterprise User ID whose Security Equals attribute values are to be returned
------	--

readAttribute

Returns an enumeration of the values of a specified attribute for a given object.

You must call the `init` method to initialize the JAsCAuth environment before calling `readAttribute`. For more information about `init`, see [“init” on page 198](#).

For details about the exceptions that can be thrown, see [“Exception Classes in `com.novell.asam.JAsCAuth`” on page 204](#).

Syntax

```
public Enumeration readAttribute(  
    String object,  
    String attribute)
```

Parameters

object	The Enterprise User ID or fully distinguished object name of the object whose attribute values are to be returned
attribute	The single-valued attribute whose value is to be returned for the object. Only the Home Directory attribute of a User object is supported at this time.

secondsToDays

Returns the integer number of days for the given number of seconds.

Syntax

```
public long secondsToDays(long secs)
```

securityEquals

Checks to see if a user has security equivalence to the specified object.

You must call the `init` method to initialize the JAsCAuth environment before calling `securityEquals`. For more information about `init`, see [“init” on page 198](#).

For details about the exceptions that can be thrown, see [“Exception Classes in com.novell.asam.JAscAuth” on page 204](#).

Syntax

```
public void securityEquals(  
    String user,  
    String object)
```

Parameters

user	The Enterprise User ID to be tested
object	The fully distinguished object name for which the security equivalence of user is to be tested

strError

Returns the string representation of the given AS Client API return code.

Syntax

```
public String strError(int rc)
```

Parameters

rc	The AS Client API return code value whose string representation is to be returned
----	---

See Also

[“getLastReturnCode” on page 198](#)

userIncludeExclude

Determines if a given user matches an AS.USER.INCLUDE or AS.USER.EXCLUDE statement in the platform configuration file.

Syntax

```
public int userIncludeExclude(String user)
```

Parameters

user	The Enterprise User ID of the user to be checked
------	--

Return Values

AS_NOMATCH	The user does not match any INCLUDE/EXCLUDE statement. Because AS.USER.INCLUDE * is implicit in the absence of AS.USER.EXCLUDE *, the user is included.
AS_INCLUDED	User matches an AS.USER.INCLUDE statement.

AS_EXCLUDED

User matches an `AS .USER .EXCLUDE` statement or an entry in the built-in standard exclude list.

Classes Used by checkPassword

The following topics describe classes used by the checkPassword method of JAscAuth to return information.

Class com.novell.asam.JAscAuth.JAscUser

The checkPassword method of JAscAuth optionally returns a JAscUser object with information about the user being authenticated.

Constructor

```
public JAscUser()
```

Fields

public JAscLoginRestrict login	Contains the user's login disabled flag
public JAscPassRestrict pass	Contains the user's password expiration information

Class com.novell.asam.JAscAuth.JAscLoginRestrict

The checkPassword method of JAscAuth optionally returns a JAscUser object with information about the user being authenticated. One of the fields in JAscUser is a JAscLoginRestrict object, which contains the user's login disabled flag.

Constructor

```
public JAscLoginRestrict()
```

Fields

public int disabled	The user's login disabled flag
---------------------	--------------------------------

Methods

public int getDisabled()	Returns the user's login disabled flag
--------------------------	--

Class com.novell.asam.JAscAuth.JAscPassRestrict

The checkPassword method of JAscAuth optionally returns a JAscUser object with information about the user being authenticated. One of the fields in JAscUser is a JAscPassRestrict object, which contains password expiration information.

Constructor

```
public JAscPassRestrict()
```

Fields

public long interval	The password change interval in seconds (or -1 if the password does not expire)
public long expire	The number of seconds until the password expires (or -1 if the password does not expire)

Methods

public long getInterval()	Returns the password change interval in seconds (or -1 if the password does not expire)
public long getExpire()	Returns the number of seconds until the password expires (or -1 if the password does not expire)

Exception Classes in com.novell.asam.JAscAuth

The following exceptions, along with java/lang/NullPointerException, are the exceptions that are thrown by the methods of JAscAuth.

InvalidJAscException

Thrown when a method requires an authentication environment, but a valid authentication environment does not exist.

Most methods of com.novell.asam.JAscAuth.JAscAuth require that you call the init method before you call them. InvalidJAscException is thrown if you do not do so.

Corresponds to a return code of 16, AS_NOAUTHENV, from the AS Client API. For more information, see [Appendix C, “Troubleshooting the API,” on page 215](#).

JAscAttrNotFoundException

Thrown when the attribute specified to the readAttr method was not found for the specified object.

Corresponds to a return code of 13, AS_ATTRNOTFOUND, from the AS Client API. For more information, see [Appendix C, “Troubleshooting the API,” on page 215](#).

JAscAuthenticationException

Thrown when the password specified to the checkPassword method is not valid.

Corresponds to a return code of 1, AS_NO, from the AS Client API. For more information, see [Appendix C, “Troubleshooting the API,” on page 215](#).

JAscBadClientException

Thrown when the network address used by the platform to contact a Core Driver for a method call does not match the network address listed in the Platform Configuration object in the ASAM System container.

Corresponds to a return code of 4, AS_BADCLIENT, from the AS Client API. For more information, see [Appendix C, “Troubleshooting the API,” on page 215](#).

JAscChangePasswordException

Thrown by changePassword when the password cannot be changed.

Also thrown by changePassword if the old password given is not valid.

Corresponds to a return code of 1, AS_NO, and a return code of 4, AS_BADCLIENT, from the AS Client API. For more information, see [Appendix C, “Troubleshooting the API,” on page 215](#).

JAscException

Thrown by most method calls when an unexpected or indeterminate error condition occurs.

JAscInsufficientRightsException

Thrown by `adminResetPassword` if the administrative user does not exist, if the administrative user password specified is not valid, or if the administrative user does not have rights to change the password.

Also thrown by `adminResetPassword` if the network address used by the platform to contact a Core Driver does not match the network address listed in the Platform Configuration object in the ASAM System container.

Corresponds to a return code of 24, `AS_INSUFFICIENTRIGHTS` from the AS Client API. For more information, see [Appendix C, “Troubleshooting the API,” on page 215](#).

JAscIntruderException

Thrown by `checkPassword` and `changePassword` when the specified user is locked because of intruder detection.

Corresponds to a return code of 6, `AS_INTRUDER`, from the AS Client API. For more information, see [Appendix C, “Troubleshooting the API,” on page 215](#).

JAscInvalidArgsException

Thrown when a parameter passed to a method is null or not valid.

Corresponds to a return code of 7, `AS_INVALIDARGS`, from the AS Client API. For more information, see [Appendix C, “Troubleshooting the API,” on page 215](#).

JAscInvalidObjException

Thrown when an object passed to a method is not found or is not of the correct type.

Corresponds to a return code of 8, `AS_INVALIDOBJ`, from the AS Client API. For more information, see [Appendix C, “Troubleshooting the API,” on page 215](#).

JAscInvalidObjLenException

Thrown when an object name passed to a method is longer than the maximum allowable name.

Corresponds to a return code of 9, `AS_INVALIDOBJLEN`, from the AS Client API. For more information, see [Appendix C, “Troubleshooting the API,” on page 215](#).

JAscInvalidReqException

Thrown when a method call is not known by the Core Driver.

Corresponds to a return code of 22, `AS_INVALIDREQ`, from the AS Client API. For more information, see [Appendix C, “Troubleshooting the API,” on page 215](#).

JAscKeyExpiredException

Thrown when the DES encryption key used by a non-SSL platform has expired.

Corresponds to a return code of 23, `AS_KEYEXPIRED`, from the AS Client API. For more information, see [Appendix C, “Troubleshooting the API,” on page 215](#).

JAscNoAgentException

Thrown when no Core Driver could be contacted to process a method call.

Corresponds to a return code of 3, `AS_NOAGENT`, from the AS Client API. For more information, see [Appendix C, “Troubleshooting the API,” on page 215](#).

JAscNoUserException

Thrown when the user specified to a method call is inactive or not in the Census.

Corresponds to a return code of 2, `AS_NOUSER`, from the AS Client API. For more information, see [Appendix C, “Troubleshooting the API,” on page 215](#).

JAscPassDuplicateException

Thrown by `changePassword` when the new password has been previously used for the user object, and the user is required to use unique passwords.

Corresponds to a return code of 10, `AS_PASSDUPLICATE`, from the AS Client API. For more information, see [Appendix C, “Troubleshooting the API,” on page 215](#).

JAscPassTooShortException

Thrown by `changePassword` when the new password is shorter than the minimum password length set for the user.

Corresponds to a return code of 11, `AS_PASSTOOSHORT`, from the AS Client API. For more information, see [Appendix C, “Troubleshooting the API,” on page 215](#).

JAscProductExpiredException

Thrown when the expiration date for the platform has passed.

Corresponds to a return code of 17, `AS_PRODUCTEXPIRED`, from the AS Client API. For more information, see [Appendix C, “Troubleshooting the API,” on page 215](#).

JAscRevokedException

Thrown by `checkPassword` and `changePassword` when the specified user is disabled.

Corresponds to a return code of 5, `AS_REVOKED`, from the AS Client API. For more information, see [Appendix C, “Troubleshooting the API,” on page 215](#).

The following topics describe simple modifications to several popular products to enable them for use with the Novell® Identity Manager Fan-Out Driver.

- ♦ [Section 17.1, “Adding API Support to the Apache Web Server,” on page 207](#)
- ♦ [Section 17.2, “Adding API Support to the QUALCOMM POP Server,” on page 207](#)
- ♦ [Section 17.3, “Adding API Support to SASL,” on page 208](#)
- ♦ [Section 17.4, “Adding API Support to SSH Secure Shell,” on page 208](#)
- ♦ [Section 17.5, “Adding API Support to TACACS+,” on page 208](#)

17.1 Adding API Support to the Apache Web Server

The Apache* HTTP Web Server software is one of the most popular Web servers in use today. It is developed by the Apache Group and can be downloaded free from the [Apache Software Foundation Web site \(http://www.apache.org\)](http://www.apache.org). Apache provides the facility to configure additional modules to handle specific functions, such as user authentication and locating a user's home directory.

You can install Platform Services on your Apache server and configure Apache to authenticate users using the AS Client API. You can also configure Apache to use the AS Client API to find a user's home directory on a NetWare® file system that is mounted on a Linux Apache server. Example modules are provided in the `ASAM/bin/PlatformServices/PlatformClient/Apache` directory created by the Platform Services installation process.

17.2 Adding API Support to the QUALCOMM POP Server

QUALCOMM* Incorporated distributes freeware Linux/UNIX POP3 software known as Qpopper* in C source form. With no modifications, Qpopper can use Authentication Services for authentication through PAM. Qpopper can also be modified to use Authentication Services for authentication through the AS Client API.

You can obtain Qpopper from the [QUALCOMM Web site \(http://www.eudora.com\)](http://www.eudora.com).

You can install Platform Services on your POP server and use Qpopper to authenticate users using Authentication Services through PAM or through the API.

Directions for modifying Qpopper to use the AS Client API can be found in the `ASAM/bin/PlatformServices/PlatformClient/POP` directory created by the Platform Services installation process.

17.3 Adding API Support to SASL

SASL, the Simple Authentication and Security Layer, is a generic authentication protocol. Many connection-based protocols, such as SMTP, LDAP, IMAP, and POP3, support SASL. New authentication mechanisms that support SASL are automatically supported by those protocols. Furthermore, protocols that use SASL for authentication support Kerberos* authentication through the Generic Security Services Application Programming Interface (GSSAPI).

A common open-source SASL library is Cyrus SASL, which is available at <ftp://ftp.andrew.cmu.edu/pub/cyrus-mail>.

Directions for modifying Cyrus SASL to use the AS Client API for authentication can be found in the `ASAM/bin/PlatformServices/PlatformClient/sasl` directory created by the Platform Services installation process.

17.4 Adding API Support to SSH Secure Shell

SSH* Communications Security produces a secure login application known as SSH Secure Shell. Source is available at <ftp://ftp.ssh.com/pub/ssh>. SSH Secure Shell is a commercial product. The rules governing the commercial and non-commercial use of SSH Secure Shell can be found at the [SSH Communications Security Web Site \(http://www.ssh.com\)](http://www.ssh.com).

You can install Platform Services on your server and modify the Secure Shell daemon, `sshd`, to use the AS Client API to authenticate users.

The `sshd` using the Identity Manager Fan-Out Driver allows users to skip setting up passphrases, because the authentication stage of setting up the Secure Shell connection is achieved with the driver instead of public-private key cryptography. After you have authenticated, your Secure Shell session is securely encrypted, as normal.

The Identity Manager Fan-Out Driver provides sample instructions for modifying the Secure Shell `sshunixuser.c` module. These instructions are distributed in the `ASAM/bin/PlatformServices/PlatformClient/SSH` directory created by the Platform Services installation process.

17.5 Adding API Support to TACACS+

TACACS+ is a security protocol designed by Cisco Systems*, Inc. that is used to control dial-up access into networks. An unsupported but freely available implementation of TACACS+ is available in <ftp://peng.cisco.com> in `pub/tacacs`.

You can install Platform Services on your server and modify TACACS+ to use the AS Client API to authenticate users.

Directions for modifying TACACS+ to use the AS Client API for authentication can be found in the `ASAM/bin/PlatformServices/PlatformClient/TACACS` directory created by the Platform Services installation process.

Appendixes

VI

Part VI includes the following appendixes:

- ◆ [Appendix A, “OS/400 Provisioning Tips,” on page 211](#)
- ◆ [Appendix B, “Password Change Validation Exit,” on page 213](#)
- ◆ [Appendix C, “Troubleshooting the API,” on page 215](#)
- ◆ [Appendix D, “Messages,” on page 219](#)

OS/400 Provisioning Tips

A

You can add additional attributes to your Novell® Identity Manager Fan-Out driver configuration for your own purposes.

A.1 Provisioning Attributes

Certain attributes are configured to be made available to platforms by default. You can add additional attributes to be provisioned to platforms if necessary. For a list of the attributes configured by default, see [“Platform Attributes” on page 89](#).

To configure additional attributes, you must add them to the Identity Manager Subscriber filter. For details about adding attributes to the Subscriber filter, see the *Novell Identity Manager Administration Guide*. The attribute names that you use in the Subscriber filter must be the eDirectory™ names. The attribute names presented to the Platform Receiver are the eDirectory names.

If you are using attributes on an OS/400 platform, you may need to map attribute names to their Profile and System Distribution Directory field names. This is done using the Attribute Name Mapping file. For more information about the Attribute Name Mapping file, see [“The Attribute Name Mapping File” on page 146](#).

A.2 Synchronizing Password Expiration

To ensure that passwords expire on your OS/400 platform at the same time as they expire in eDirectory, you must make two additional attributes available to the platform.

Add the following two attribute names to the Subscriber filter:

Password Expiration Interval

Password Expiration Time

Password Change Validation Exit

B

The Core Driver of the Novell® Identity Manager Fan-Out can call a user-provided routine to enforce local password rules. This routine is called when a password change request is received from a password redirection platform.

The Password Change Validation Exit is passed the fully distinguished name of the user, the old password, the new password, and a message buffer. The exit can accept or reject the password change request and, if the request is rejected, provide an explanation in the message buffer. The explanation is written to the Core Driver Audit log and is displayed to the user.

A sample Password Change Validation Exit is provided in the ASAM directory created by the installation process in `asam\bin\coredriver\chgpaswdexit\verpass.c`.

To implement the Password Change Validation Exit:

- 1** Design, write, and build your Password Change Validation Exit. You can use the sample Password Change Validation Exit `verpass.c` as a guide.
- 2** Place a copy of the library containing your Password Change Validation Exit on each server that runs a Core Driver.
- 3** Specify the appropriate Change Password Exit Function and Change Password Exit Library configuration parameters for each Core Driver. For details, see [“Driver Object Configuration Parameters” on page 82](#).

Troubleshooting the API

C

Most calls to the Novell® Identity Manager Fan-Out Driver AS Client API return a value that describes the outcome of the call. These return code values are defined in the C language `ascauth.h` header file and are provided as fields in the `JAscAuth` class in the Java interface. The C language API function `ASC_STRError()` and the Java interface method `strError()` can be used to return a text string that corresponds to the return code. This text string is included in many of the messages that are written to the platform log file for errors involving API calls.

The Java interface uses exceptions for most non-affirmative API call outcomes.

The following table lists the return codes and their corresponding text string, and suggests actions to take for them.

Table C-1 Return Codes for Troubleshooting

Return Code	Symbol Text String	Explanation and Suggested Action
0	AS_OK Action successful	The operation returned a positive response. For calls such as check password, this corresponds to an answer of "Yes."
1	AS_NO Action not successful	The operation returned a negative response. For calls such as check password, this corresponds to an answer of "No."
2	AS_NOUSER Unknown user	The Enterprise User ID specified on the call is inactive or is not in the Census. Action: If you expect this user to be active in the Census, see Part II, "Core Driver Administration," on page 37 for additional information.
3	AS_NOAGENT No Core Drivers are available for authentication	No Core Drivers could be contacted to process the request. Action: This is generally caused by a configuration problem. <ul style="list-style-type: none">◆ Ensure that the platform configuration file specifies the correct network addresses for the Core Drivers.◆ Ensure that the Core Driver is running on the specified servers and listening on the port specified in the platform configuration file.◆ Ensure that the Platform Services Process is running or that you have specified the <code>DIRECTTOAUTHENTICATION</code> statement in your platform configuration file.◆ Ensure that you have network connectivity to a Core Driver server.◆ Ensure that driver has been activated or that the evaluation period has not expired.

For more information, see parts 2 and 3 in this guide.

Return Code	Symbol Text String	Explanation and Suggested Action
4	AS_BADCLIENT Local host is not authorized to query the Core Driver	<p>The network address used by the platform to contact a Core Driver did not match the network address listed in the Platform Configuration object in the ASAM System container.</p> <p>Action: For information about managing Platform objects with the Web interface, see .</p> <p>For an administrative password reset, this can indicate that the administrator user ID/password is not valid or that the administrator does not have rights to change the password.</p>
5	AS_REVOKED User is disabled/revoked	<p>The specified Enterprise User ID corresponds to a User object that has been disabled.</p>
6	AS_INTRUDER Intruder detection is active	<p>The specified Enterprise User ID corresponds to a User object that has been locked because of intruder detection.</p>
7	AS_INVALIDARGS Invalid arguments	<p>The arguments specified on the call are not valid.</p> <p>Action: Make certain that the arguments passed to the call are of the correct type and value. For example, an argument that specifies the name of an object cannot be blank or null, and an argument that specifies a buffer size to hold a result cannot be zero.</p>
8	AS_INVALIDOBJ Invalid object	<p>An object specified as an argument was not of the correct type or was not found.</p> <p>Action: Verify that the objects specified on arguments to the call are of the proper type. Handle the not-found condition as appropriate for your application.</p>
9	AS_INVALIDOBJLEN Invalid object length	<p>An object name specified as an argument was longer than the maximum allowable eDirectory™ object name.</p> <p>Action: Check object names that are specified as arguments to be sure that they do not exceed the maximum length for an eDirectory object name.</p>
10	AS_PASSDUPLICATE Password has been previously used	<p>The new password that was specified to the change password API function has been previously used for this User object and the User object is required to specify unique passwords.</p> <p>Action: Specify a password that has not been previously used.</p>
11	AS_PASSTOOSHORT Password does not meet password rules	<p>The new password that was specified to the change password API function is shorter than the minimum password length set for the User object.</p> <p>Action: Specify a password that meets the password rules for the User object.</p>
12	AS_TOOSMALL Buffer is too small	<p>The size specified for a buffer argument is too small to hold the result. The result is truncated.</p> <p>Action: Allocate a larger buffer, and issue the request again.</p>

Return Code	Symbol Text String	Explanation and Suggested Action
13	AS_ATTRNOTFOUND Attribute not found	The attribute specified was not found for the specified object. Action: Process this response accordingly, or specify the name of an attribute that exists for the specified object.
14	AS_WSOCKUP WINSOCK not initialized	Not used in the Identity Manager Fan-Out Driver.
15	AS_WSOCKDOWN WINSOCK not terminated	Not used in the Identity Manager Fan-Out Driver.
16	AS_NOAUTHENV No authentication environment established	The asce argument did not specify a valid environment item. C Action: Verify that a successful call to ASC_INIT() or ASC_INIT_EXT() has been made. Successful calls return a pointer to a valid environment item. Unsuccessful calls return NULL. Verify that the pointer to the environment item is correctly specified as an argument to this call. Java Action: Verify that a successful call to init() has been made.
17	AS_PRODUCTEXPIRED Ascauth client has expired	The expiration date for the platform has passed. Action: Install a current version of Platform Services.
18	AS_INCLUDED User matched an INCLUDE statement	The Enterprise User ID specified on a call to ASC_USER_INCLUDE_EXCLUDE() matched an AS.USER.INCLUDE statement in the platform configuration file.
19	AS_EXCLUDED User matched an EXCLUDE statement	The Enterprise User ID specified on a call to ASC_USER_INCLUDE_EXCLUDE() matched an AS.USER.EXCLUDE statement in the platform configuration file.
20	AS_NOMATCH User did not match any INCLUDE/EXCLUDE statement	The Enterprise User ID specified on a call to ASC_USER_INCLUDE_EXCLUDE() did not match any AS.USER.INCLUDE or AS.USER.EXCLUDE statement in the platform configuration file. The user is included because AS.USER.INCLUDE * is implicit if AS.USER.EXCLUDE * is not specified.
21	AS_NOLICENSE Client is not licensed to use the driver	Not used in the Identity Manager Fan-Out Driver.
22	AS_INVALIDREQ API request is not valid or unsupported	The AS Client API call was not recognized by the Core Driver. Action: Ensure that the version of Platform Services and the version of the Core Driver are compatible.

Return Code	Symbol Text String	Explanation and Suggested Action
23	AS_KEYEXPIRED Client is using an expired DES key	The DES encryption key used by a non-SSL version of Platform Services has expired. Action: Update the KEY statement in the platform configuration file with the same encryption key that is specified for the Platform in the Platform object in the ASAM System container. For information about managing Platform objects with the Web interface, see Part II, "Core Driver Administration," on page 37 .
24	AS_INSUFFICIENTRIGHTS Client is using an expired DES key	An administrative password reset was rejected. The administrative user does not exist, the password given for the administrative user is not valid, or the administrative user does not have rights to change the target user's password.

Messages

D

Novell® Identity Manager Fan-Out Driver components write messages to their Operational Logs, the System Log, and the Audit Log. These messages record key processing occurrences, diagnostic information, and general statistical information. The messages can be useful to you in monitoring the operation of the driver and in troubleshooting problems.

This section presents all existing messages for the Novell Identity Manager Fan-Out Driver. Each message is followed by one or more explanation(s), possible cause(s), and suggested action(s) as needed.

Each message begins with a code of 3-5 characters associated with the driver component that generated the message. Use this code to find message information quickly as follows:

- ◆ [Section D.1, “Message Format,” on page 220](#)
- ◆ [Section D.2, “Message Destination,” on page 221](#)
- ◆ [Section D.3, “AGT Messages,” on page 222](#)
- ◆ [Section D.4, “ASC Messages,” on page 226](#)
- ◆ [Section D.5, “AUDA Messages,” on page 230](#)
- ◆ [Section D.6, “AUDG Messages,” on page 233](#)
- ◆ [Section D.7, “AUDR Messages,” on page 235](#)
- ◆ [Section D.8, “AXML Messages,” on page 241](#)
- ◆ [Section D.9, “CFG Messages,” on page 243](#)
- ◆ [Section D.10, “CFGGA Messages,” on page 244](#)
- ◆ [Section D.11, “CFGP Messages,” on page 245](#)
- ◆ [Section D.12, “CRT Messages,” on page 245](#)
- ◆ [Section D.13, “DIR Messages,” on page 247](#)
- ◆ [Section D.14, “DOM Messages,” on page 258](#)
- ◆ [Section D.15, “DRVCOM Messages,” on page 259](#)
- ◆ [Section D.16, “EJS Messages,” on page 259](#)
- ◆ [Section D.17, “HES Messages,” on page 269](#)
- ◆ [Section D.18, “LWS Messages,” on page 269](#)
- ◆ [Section D.19, “NET Messages,” on page 275](#)
- ◆ [Section D.20, “OAP Messages,” on page 275](#)
- ◆ [Section D.21, “OBJ Messages,” on page 276](#)
- ◆ [Section D.22, “PLS Messages,” on page 293](#)
- ◆ [Section D.23, “PRCV Messages,” on page 293](#)
- ◆ [Section D.24, “RDXML Messages,” on page 298](#)
- ◆ [Section D.25, “W3LM Messages,” on page 300](#)

D.1 Message Format

Each message written by the driver begins with a message identifier. The text of the message follows the message identifier. A diagnostic code, meaningful to the Novell product support team, follows the message text.

Example message:

```
OBJ010I Trawl complete. aas1625
```

In this example, the message identifier is OBJ010I. The message text is Trawl complete. The diagnostic code is aas1625.

The last character of the message identifier represents one of the following possible severity codes:

Table D-1 Message Severity Codes

D	Debugging
I	Informational
W	Warning
E	Error

Each message identifier begins with a code of 3-5 characters associated with the driver component that generated the message. Message explanations in this reference are grouped according to these codes so you can find them quickly.

- ◆ [“AGT Messages” on page 222](#)
Messages beginning with AGT are issued by the Core Driver for Authentication Services.
- ◆ [“ASC Messages” on page 226](#)
Messages beginning with ASC are issued by ASCLIENT, the z/OS Platform Services Process.
- ◆ [“AUDA Messages” on page 230](#)
Messages beginning with AUDA are issued by Audit Services for Authentication Services.
- ◆ [“AUDG Messages” on page 233](#)
Messages beginning with AUDG are issued by Audit Services for general components.
- ◆ [“AUDR Messages” on page 235](#)
Messages beginning with AUDR are issued by Audit Services to report actions taken during Receiver script processing.
- ◆ [“AXML Messages” on page 241](#)
Messages beginning with AXML are issued by the Core Driver during interactions with the Identity Manager engine.
- ◆ [“CFG Messages” on page 243](#)
Messages beginning with CFG are issued by Platform Configuration file processing.
- ◆ [“CFG A Messages” on page 244](#)
Messages beginning with CFG A are issued during installation when migrating values from the asamcore.conf file to Driver object configuration parameters.

- ◆ **“CFGP Messages” on page 245**
Messages beginning with CFGP are issued by platform configuration file processing.
- ◆ **“CRT Messages” on page 245**
Messages beginning with CRT are issued by Certificate Services.
- ◆ **“DIR Messages” on page 247**
Messages beginning with DIR are issued by the Core Driver during LDAP directory access.
- ◆ **“DOM Messages” on page 258**
Messages beginning with DOM are issued by driver components as they communicate among themselves.
- ◆ **“DRVCOM Messages” on page 259**
Messages beginning with DRVCOM are issued by the include/exclude system.
- ◆ **“EJS Messages” on page 259**
Messages beginning with EJS are issued by Event Journal Services.
- ◆ **“HES Messages” on page 269**
Messages beginning with HES are issued by driver components as they use HTTP to communicate.
- ◆ **“LWS Messages” on page 269**
Messages beginning with LWS are issued by the Core Driver as it functions as an HTTP server.
- ◆ **“NET Messages” on page 275**
Messages beginning with NET are issued by driver components during verification of SSL certificates.
- ◆ **“OAP Messages” on page 275**
Messages beginning with OAP are issued by driver components when communicating among themselves.
- ◆ **“OBJ Messages” on page 276**
Messages beginning with OBJ are issued by Object Services.
- ◆ **“PLS Messages” on page 293**
Messages beginning with PLS are issued by Platform Services.
- ◆ **“PRCV Messages” on page 293**
Messages beginning with PRCV are issued by Platform Receivers.
- ◆ **“RDXML Messages” on page 298**
Messages beginning with RDXML are issued by the embedded Remote Loader.
- ◆ **“W3LM Messages” on page 300**
Messages beginning with W3LM are issued by Web Services.

D.2 Message Destination

Audit Services maintains the Operational Logs and Audit Logs for the Core Driver in the logs directory. You can use the Web interface to view these logs.

Other log messages are handled depending on the system as follows.

D.2.1 i5/OS

System messages from the i5/OS (OS/400 operating system) Platform Services Process and Platform Receiver are written to standard OS/400 job logs. You can use DSPJOBLOG or iSeries* Navigator to view these job logs.

D.2.2 z/OS

System messages from the z/OS Platform Services Process and Platform Receiver are written using the z/OS WTO service with Route Code 11. Other messages are written to the ASCLOG DD statement.

D.2.3 Linux and UNIX

System messages written by the Core Driver, and all messages written by the Linux/UNIX Platform Services Process and Platform Receiver, are written using the SYSLOG facility specified by the SYSLOGFACILITY statement of their respective configuration files.

The severity code of each message is used to determine the priority as follows.

Table D-2 *Linux/UNIX Message Destination by Severity Code*

Severity	Priority
Debugging	LOG_DEBUG
Informational	LOG_INFO
Warning	LOG_WARNING
Error	LOG_ERR

D.2.4 Windows

System messages written by the Core Driver are written to the Windows Application Log.

D.3 AGT Messages

Messages beginning with AGT are issued by the Core Driver for Authentication Services.

AGT001I Password Migration Mode is enabled.

Explanation: Password Migration Mode is enabled for this Core Driver. This mode is enabled by setting the Migration Password parameter for the Driver object.

Action: None. Informational only.

AGT002I < thread_id > Processing compatibility mode request from ipAddress on port portNumber.

Explanation: A new platform request identified by *thread_id* has been started from *ipAddress* on *portNumber*.

Action: None. Informational only.

AGT003E < *thread_id* > Error reading from socket connected to *ip_address*.

Explanation: The Core Driver was unable to read data from the socket connection. The current request is discarded.

Possible Cause: The platform might have dropped the connection.

Action: If this error occurs frequently, check for network connectivity problems between the platform and the Core Driver.

AGT004I < *thread_id* > Compatibility mode request has ended.

Explanation: The platform request identified by *thread_id* has ended.

Action: None. Informational only.

AGT005E < *thread_id* > Invalid request was received from the platform.

Explanation: The platform sent an invalid request to the Core Driver.

Possible Cause: The platform is configured with an invalid DES key.

The platform host is running a down-level version of the platform software.

Action: Ensure that the DES key in the Platform Configuration file matches the DES key for the Platform object in eDirectory™.

AGT006W < *thread_id* > Request received from an unauthorized platform *ip_address*.

Explanation: A request was received from a platform that is not known. The request is discarded.

Possible Cause: The IP address or host name of the platform does not match the IP addresses or host names listed for the Platform object in eDirectory.

Someone might be attempting to breach security.

Action: Use the Web interface to add the network address for the platform to its corresponding Platform object in eDirectory if appropriate.

AGT007E < *thread_id* > DES key has expired for Platform *ip_address*.

Explanation: The DES key being used by the platform on IP address *ip_address* has expired. The request is discarded.

Possible Cause: A new DES key was set for the platform using the Web interface, but the platform has not been changed to use the new DES key. The old DES key is expired and unusable.

Action: Update the Platform Configuration file with the new DES key.

AGT008W < *thread_id* > Response to *request_type* request from *ip_address* for *objectDN* is: *answer*.

Explanation: A request was received from *ip_address* by the Core Driver for *request_type* and was sent the response *answer*.

Action: None. Informational only.

AGT009W < thread_id> Response to request_type request from ip_address is: answer.

Explanation: A request was received by the Core Driver for *request_type* and was sent the response *answer*.

Action: None. Informational only.

AGT010E < thread_id> Error writing to socket connected to ip_address.

Explanation: The Core Driver was unable to write data to the socket connection for *ip_address*. The current request is discarded.

Possible Cause: The platform might have dropped the connection.

Action: If this error occurs frequently, check for network connectivity problems between the platform and the Core Driver.

AGT013W The password submitted for user user does not match the user's eDirectory password. Error code code (attempt attempt).

Explanation: A password submitted by the Novell Client™ Intercept did not match the user's eDirectory password. This message is a concern only if password replication is not working.

Possible Cause: If a user changes the eDirectory password on Windows, the Novell Client could do a password check on the old password. By the time this password event is processed, the eDirectory password could have been changed to the new password.

Sometimes the password event is checked against eDirectory before the eDirectory password arrives on the replica the Core Driver is using.

Another possibility is that the event came from a different tree than the one the Core Driver is using, and that tree is not listed on the External Password Sources driver parameter.

This message can indicate an intruder attempt to change a password.

Action: Examine the Core Driver's Operational Log. If soon after this message you get an ePassword successfully updated message for the user, the issue is probably transitory and no action is necessary.

If password replication is not working and you are receiving this message, check your eDirectory replication performance.

Verify that the Storage Key parameter in the driver parameters is the same for all Core Drivers.

AGT018E Password replication for user user failed with error code code.

Explanation: The Core Driver could not store its encrypted copy of the password to eDirectory. No more attempts are made to do so. The user's password is not replicated to any platforms.

Possible Cause: For ePassword operation, the most likely cause is that the LDAP server specified in the driver parameters is down or misconfigured. For more information, see message AGT023E.

Action: See message AGT023E.

AGT023E Write of ePassword for user *user* failed with error code *code*. (LDAP server: *server*: *port*).

Explanation: The Core Driver could not store the user's ePassword in eDirectory. Without an ePassword, the user cannot be replicated to platforms that require password synchronization. The Core Driver might be able to recover from this problem.

Possible Cause: The LDAP server specified in the driver configuration parameters is down or misconfigured. This generally results in error codes 80 or 81.

Action: Verify that the LDAP server specified in the driver parameters is the correct server.

Verify that the LDAP server specified in the driver parameters is running and configured properly. For information about the LDAP server, refer to your eDirectory documentation.

Verify that the computer running the Core Driver can communicate with the LDAP server using TCP/IP.

Check the eDirectory replication status.

AGT024I ePassword successfully updated for user *user*.

Explanation: The Core Driver successfully stored the ePassword for the specified user. This password is replicated to the appropriate platforms.

Possible Cause: ePasswords can be updated through:

Any password change to a user covered by a valid Universal Password policy.

eDirectory password checks and changes made on a computer with the Novell Client Intercept installed.

Password checks and changes sent to the Core Driver from a System Intercept or using the AS Client API.

Action: No action necessary.

AGT025I Password Change Validation Exit Registered using *function* from library *library*.

Explanation: A Password Change Validation Exit was registered using the indicated function and library.

Action: None

AGT026E Could not open library *library* for Password Change Validation Exit.

Explanation: The Core Driver could not open the library specified for the Password Change Validation Exit.

Action: Make sure the library exists in the location you specified.

AGT027E Could not import function *function* from library *library* for Password Change Validation Exit.

Explanation: The Core Driver could not import the specified function from the specified library for the Password Change Validation Exit.

Action: Make sure the function is exported from the library.

AGT028E The Password Change Validation Exit has rejected the password change for user *user*. Reason: *reason*.

Explanation: The registered Password Change Validation Exit has applied a user-defined set of rules to the attempted password change and determined that the new password is not valid. A reason is displayed if the exit provided one.

Action: None.

D.4 ASC Messages

Messages beginning with ASC are issued by ASCLIENT, the z/OS Platform Services Process.

ASC0000I *command response*.

Explanation: This message is issued in response to a DISPLAY command.

ASC0001I *productVersion* was started on *date* at *time*.

Explanation: This message is issued during ASCLIENT startup.

ASC0004I *n* request(s) found on existing queue.

Explanation: ASCLIENT found *n* requests awaiting processing on the in-storage queue during a warm startup.

ASC0005E Unable to load interface module *moduleName*: abend *abendCode*.

Explanation: ASCLIENT attempted to load *moduleName* into ECSA, but the load macro failed. Register 1 contains system completion code *abendCode*.

Action: Respond as appropriate for *abendCode*.

ASC0007I *productVersion* has ended.

Explanation: ASCLIENT has shut down in response to a STOP command.

ASC0010I *statistics line*.

Explanation: This message is used to write statistical information to the console.

ASC0011I Command entered was: *command_text*.

Explanation: ASCLIENT accepted a MODIFY command as valid, and attempts to execute it.

ASC0012I New parameter file applied.

Explanation: ASCLIENT has updated its configuration in response to a CONFIG *xx* command.

ASC0013I Log switched.

Explanation: The LOGSWITCH command was entered. The ASCLOG file has been closed, freed, reallocated, and reopened. The previous log is available for processing.

ASC0020E WARM and COLD parameters are mutually exclusive.

Explanation: The startup parameters WARM and COLD were both found in the PARM string in the ASCLIENT startup procedure.

Action: Remove one of the two parameters from the PARM string.

ASC0021E Module *moduleName* does not contain valid global token. Module not loaded.

Explanation: During configuration file processing, ASCMAIN was unable to set an internal token value as part of loading module ASCUSER. If this occurs during startup, ASCMAIN terminates with a return code of 8. If the message appears after a CONFIG command, the new configuration is not accepted.

Action: The copy of ASCUSER or ASCMAIN in the ASCLIENT STEPLIB could be corrupted. Reload the load library from the distribution media and compare these modules with the copies in the ASCLIENT STEPLIB. If you cannot account for any differences, re-create the load library from the distribution, and reinstall any patches you have applied.

ASC0022E Invalid keyword beginning with: *text*.

Explanation: An invalid keyword appears in the PARM string for ASCLIENT. ASCLIENT terminates.

Action: Correct the PARM string and restart ASCLIENT.

ASC0023E Invalid parameter beginning with: *text*.

Explanation: A valid command in the PARM string is followed by an invalid operand. ASCLIENT terminates.

Action: Correct the PARM string and restart ASCLIENT.

ASC0024E Name/Token Services *function* error: *description*.

Explanation: z/OS Name/Token Services returned the unexpected error described in the message.

Action: Make note of the message text, collect diagnostic information, and contact Support.

ASC0025E Parameter file error, line *n*: *description*.

Explanation: The syntax error described in the message was detected in the configuration file at line *n*. If this occurs at startup, ASCLIENT terminates. If this occurs as a result of a CONFIG command, the new configuration is not loaded, and ASCLIENT continues with its current configuration.

Action: Correct the syntax error in the configuration file, and either restart ASCLIENT, or reissue the CONFIG command.

ASC0026E Unable to process parameter file *name*.

Explanation: The ASCPARMS data set name could not be determined. The most likely cause is that ASCPARMS is not defined in the ASCLIENT JCL.

Action: Ensure that ddname ASCPARMS is defined and specifies a valid configuration file data set.

ASC0027E Unable to free ASCPARMS: dynalloc error code is *error*, *info*.

Explanation: ASCMAIN encountered an error while trying to deallocate the configuration file during configuration file processing. The dynamic allocation error and information codes appear in the message.

Dynamic allocation return codes and reason codes are documented in the IBM publication *z/OS Programming: Authorized Assembler Services Guide*.

Action: Respond as appropriate for the dynamic allocation error and information codes.

ASC0028E No *parametertype* statements in parameter file.

Explanation: ASCLIENT did not find any *parametertype* statements in the configuration file. The parameter type named in the message is a required parameter. Configuration file syntax checking continues. If this error occurs during startup, ASCLIENT terminates. If the error occurs as the result of a CONFIG command, the ASCLIENT configuration is not changed.

Action: If the missing statement type is AUTHENTICATION, add appropriate AUTHENTICATION statements to the parameter file. If the missing statement type is KEY, add one KEY statement to the parameter file. The key value must match the DES encryption key specified in the Core Driver configuration object for the platform. Then, either restart ASCLIENT (for failures during startup) or reissue the CONFIG command (for configuration changes during execution).

ASC0029E Unable to load name-token module *modulename*.

Explanation: One or more of the z/OS callable services routines (IEANTCR, IEANTRT, or IEANTDL) could not be located in LPA or the linklist.

Action: These routines are shipped with z/OS in CSSLIB. Check your SMP target zone for z/OS to see where these load modules reside. If SYS1.CSSLIB contains these modules but is not in your linklist, consider adding it. The caller-side interface module ASCUSER also requires IEANTRT and is not able to call ASCLIENT without it.

ASC0032E Unable to load ASCHOST.

Explanation: During initialization, ASCLIENT did not find module ASCHOST in STEPLIB. ASCLIENT terminates.

Action: Ensure that load module ASCHOST is available in the ASCLIENT STEPLIB concatenation.

ASC0033E Internal error occurred: *description*.

Explanation: An unexpected error occurred during normal processing. The error description provides additional information. If this message is issued by an ASCLIENT subtask, the subtask issues a User 0001 ABEND with a dump. If the message is issued by the ASCLIENT main task, ASCLIENT terminates.

Action: Make note of the message text, collect diagnostic information, and contact Support.

ASC0036E DELAY.PREFGRP value of *pg_delay* is less than DELAY.AUTHENTICATION value of *ag_delay*.

Explanation: The DELAY.PREFGRP and/or DELAY.AUTHENTICATION statements were found in the ASCLIENT configuration file. The value for DELAY.PREFGRP (either the value specified or the default of 5) is less than the value specified or defaulted to for DELAY.AUTHENTICATION. The value of DELAY.PREFGRP must equal or exceed that of DELAY.AUTHENTICATION.

Action: Correct the DELAY statement(s) in the ASCLIENT configuration file, then reload the configuration file or restart ASCLIENT.

ASC0037E Log not switched; ASCLOG was allocated by DD statement.

Explanation: The LOGSWITCH command was entered, but ASCLIENT can cause a log switch only if the ASCLOG DD statement is dynamically allocated (not specified in the ASCLIENT JCL).

Action: If you need to use the LOGSWITCH command, remove the ASCLOG DD statement from the ASCLIENT JCL and restart ASCLIENT.

ASC0050E Invalid command beginning with: *text*.

Explanation: An invalid command was entered as an ASCLIENT MODIFY command.

Action: Correct and reissue the command.

ASC0051E Invalid operand beginning with: *text*.

Explanation: A valid ASCLIENT MODIFY command was entered, but one of its operands was not recognized.

Action: Correct and reissue the command.

ASC0052E STOP/MODIFY command processing error: *description*.

Explanation: An internal QEDIT error occurred during ASCLIENT STOP/MODIFY processing.

Action: Make note of the message text, collect diagnostic information, and contact Support.

ASC0070E Name-token services error in *exitname*.

Explanation: The security system exit encountered an unexpected error while trying to find the environment.

Action: Make note of the message text, collect diagnostic information, and contact Support.

ASC0071I Userid *user* will be authenticated locally.

Explanation: No Core Drivers could be contacted to authenticate a user.

Possible Cause: The ASCLIENT started task is not running.

TCP/IP is not available.

No Core Drivers responded to a request.

Action: Review the started task log files for more information.

ASC0073I ACF2 password not changed. Use the LOGON command to change your password.

Explanation: The user attempted to change the password using the ACF command. This is disallowed. Change the password using the LOGON command, using the Novell Client, or from another system that is running Platform Services.

Action: None; informational only. This message is a ROUTCDE=11 WTO and appears on z/OS consoles that are set to display ROUTCDE=11 messages.

ASC0074I ACF2 password not changed. Unknown error in ACF2 new-password exit.

Explanation: ASCLIENT returned an unexpected response to ASCNPXIT during a password change request. The user's password is not changed in the security database. Whether or not the password is changed in eDirectory depends on the nature of the error.

Action: Examine the ASCLIENT task logs to see what messages or errors were returned to ASCLIENT from the Core Driver. Contact Support if you cannot determine the cause of the error.

D.5 AUDA Messages

Messages beginning with AUDA are issued by Audit Services for Authentication Services.

AUDA001I Administrative Password Reset by *driver_name* for Platform *platform_name* IP address *platform_ip_address*: eUser *eUser*, Return Value *rc*, Elapsed Time *seconds*.

Explanation: The Core Driver identified by *driver_name* processed an Administrative Password Reset request for the platform identified by *platform_name* and *platform_ip_address*. The eUser whose password was reset is *eUser*. The return code from the Core Driver to the platform was *rc*. The Core Driver took *seconds* seconds to process the request.

Action: None. Informational only.

AUDA002W Connection Rejected by *driver_name* for Platform *platform_name* IP address *platform_IP_address*: Reason *reason*.

Explanation: The Core Driver identified by *driver_name* rejected a connection attempt from the platform identified by *platform_name* and *platform_IP_address*. If the request was from a platform that does not have a configuration object in the ASAM System container, *platform_name* is empty. The reason the connection attempt was rejected is given by *reason*.

Action: Correct the cause of the error based on the reason given by *reason*.

AUDA003I Check Password by *driver_name* for Platform *platform_name* IP address *platform_ip_address*: eUser *eUser*, Return Value *rc*, Elapsed Time *seconds*.

Explanation: The Core Driver identified by *driver_name* processed a Check Password request for the platform identified by *platform_name* and *platform_ip_address*. If the request was from a platform that does not have a configuration object in the ASAM System container, *platform_name* is empty. The eUser whose password was checked is *eUser*. The return code from the Core Driver to the platform was *rc*. The Core Driver took *seconds* seconds to process the request.

Action: None. Informational only.

AUDA004I Change Password by *driver_name* for Platform *platform_name* IP address *platform_ip_address*: eUser *eUser*, Return Value *rc*, Elapsed Time *seconds*.

Explanation: The Core Driver identified by *driver_name* processed a Change Password request for the platform identified by *platform_name* and *platform_ip_address*. If the request was from a platform that does not have a configuration object in the ASAM System container, *platform_name* is empty. The eUser whose password was to be changed is *eUser*. The return code from the Core Driver to the platform was *rc*. The Core Driver took *seconds* seconds to process the request.

Action: None. Informational only.

AUDA005I Get Context by *driver_name* for Platform *platform_name* IP address *platform_ip_address*: eUser *eUser*, Return Value *rc*, Elapsed Time *seconds*.

Explanation: The Core Driver identified by *driver_name* processed a Get Context request for the platform identified by *platform_name* and *platform_ip_address*. If the request was from a platform that does not have a configuration object in the

ASAM System container, *platform_name* is empty. The eUser whose context was to be obtained is *eUser*. The return code from the Core Driver to the platform was *rc*. The Core Driver took *seconds* seconds to process the request.

Action: None. Informational only.

AUDA006I Get Security Equivalents by *driver_name* for Platform *platform_name* IP address *platform_ip_address*: eUser *eUser*, Return Value *rc*, Elapsed Time *seconds*.

Explanation: The Core Driver identified by *driver_name* processed a Get Security Equivalents request for the platform identified by *platform_name* and *platform_ip_address*. The eUser whose security equivalences list was to be obtained is *eUser*. The return code from the Core Driver to the platform was *rc*. The Core Driver took *seconds* seconds to process the request.

Action: None. Informational only.

AUDA007I Get Group Members by *driver_name* for Platform *platform_name* IP address *platform_ip_address*: Group *group*, Return Value *rc*, Elapsed Time *seconds*.

Explanation: The Core Driver identified by *driver_name* processed a Get Group Members request for the platform identified by *platform_name* and *platform_ip_address*. The group whose member list was to be obtained is *group*. The return code from the Core Driver to the platform was *rc*. The Core Driver took *seconds* seconds to process the request.

Action: None. Informational only.

AUDA008I Check Security Equivalence by *driver_name* for Platform *platform_name* IP address *platform_ip_address*: eUser *eUser* to object *object*, Return Value *rc*, Elapsed Time *seconds*.

Explanation: The Core Driver identified by *driver_name* processed a Check Security Equivalence request for the platform identified by *platform_name* and *platform_ip_address*. The eUser *eUser* was checked for security equivalence to the object *object*. The return code from the Core Driver to the platform was *rc*. The Core Driver took *seconds* seconds to process the request.

Action: None. Informational only.

AUDA009I Check Rights to Attribute by *driver_name* for Platform *platform_name* IP address *platform_ip_address*: Object1 *object1*, Rights [*rights*], Attribute *attribute_name*, Object2 *object2*, Return Value *rc*, Elapsed Time *seconds*.

Explanation: The Core Driver identified by *driver_name* processed a Check Rights to Attribute request for the platform identified by *platform_name* and *platform_ip_address*. The object *object1* was checked for the rights *rights* to the attribute *attribute_name* of object *object2*. The return code from the Core Driver to the platform was *rc*. The Core Driver took *seconds* seconds to process the request.

Action: None. Informational only.

AUDA010I Get Attribute by *driver_name* for Platform *platform_name* IP address *platform_ip_address*: Object *object*, Attribute *attribute_name*, Return Value *rc*, Elapsed Time *seconds*.

Explanation: The Core Driver identified by *driver_name* processed a Get Attribute request for the platform identified by *platform_name* and *platform_ip_address*. The value of the attribute *attribute_name* for object *object* was to be obtained. The return code from the Core Driver to the platform was *rc*. The Core Driver took *seconds* seconds to process the request.

Action: None. Informational only.

AUDA011I Password Notify on *driver* for Platform *system address*: User *user*, Tree *tree*, Return Value *rc*.

Explanation: An eDirectory password verify, change, or administrative reset was intercepted for the specified user on the specified platform. A return value of 0 indicates successful processing. A value of -1 indicates an error.

Possible Cause: The user performed an action on the platform to verify or change the password, or an administrator reset a user's password.

If the return value is -1, the event might have failed because the system is low on memory or the event was submitted from a different eDirectory tree than the one the Core Driver is operating on.

Action: If success (0) is returned, no action is necessary, unless you do not want passwords intercepted on the platform.

If you do not want passwords intercepted on the specified platform system, use the Windows Add/Remove Programs utility to remove the Novell Client Password Intercept from that platform.

If events are failing, verify that your system has enough memory.

If events are coming from other eDirectory trees, you can ignore them. If you want those events to be processed, specify the tree name on the External Password Sources driver configuration parameter.

D.6 AUDG Messages

Messages beginning with AUDG are issued by Audit Services for general components.

AUDG001I *component_object* started: Version *version* ID= *code_id_string*, Tree *tree_name*, ASAM System Container *system_container*, ASAM Master User *master_user*, Command Line *command_line*.

Explanation: The component identified by *component_object* has started. It is version *version* with code identification *code_id_string*. The directory tree used is *tree_name*. The system container in use is *system_container*. The Master User is *master_user*. The command line used to start the component was *command_line*.

Action: None. Informational only.

AUDG002I *component_object* ended. Start time was *time_stamp*.

Explanation: The component identified by *component_object* has ended. It was started at *time_stamp*.

Action: None. Informational only.

AUDG003I *component_object* Interval Start Time: *interval_start_time*: *name* = *value*.

Explanation: The component identified by *component_object* is reporting periodic statistical information. The measurement interval began at *interval_start_time*. The statistic name is *name*. The statistic value is *value*.

Action: None. Informational only.

AUDG004I *component_object* Interval Start Time: *interval_start_time*: Platform: *platform_object name* = *value*.

Explanation: The Core Driver identified by *component_object* is reporting periodic statistical information for services to the platform identified by *platform_object*. The measurement interval began at *interval_start_time*. The statistic name is *name*. The statistic value is *value*.

Action: None. Informational only.

AUDG007E Unable to write to log file because of insufficient memory.

Explanation: Insufficient memory was available to write a message to the log file. An attempt is made to write the message to the system log.

Possible Cause: Insufficient memory.

Action: Determine and correct the cause of the memory problem.

AUDG008E Unable to open log file *filename*.

Explanation: Audit Services could not open *filename* in order to write a log message. An attempt is made to write the message to the system log.

Possible Cause: The ASAM Directory driver configuration parameter is incorrect.

The Core Driver does not have the necessary file system rights.

Action: Examine the system log. Determine and correct the cause of the problem.

AUDG009E Unable to write to *logtype* log file. Failed with errno *errno*.

Explanation: Audit Services could not write a message to the *logtype* log. An attempt is made to write the message to the system log.

Action: Examine the system log. Determine and correct the cause of the problem.

AUDG010E Unable to write to *logtype* log file index. Failed with errno *errno*.

Explanation: Audit Services could not write a message to the *logtype* log because of a problem writing to the log index. An attempt is made to write the message to the system log.

Action: Examine the system log. Determine and correct the cause of the problem.

AUDG011E Error logging message to log file. Internal error *interr symbolicname*.

Explanation: Audit Services could not write a message to the log. The message is identified by *symbolicname*. An attempt is made to write the message to the system log.

Action: Examine the system log. Determine and correct the cause of the problem.

D.7 AUDR Messages

Messages beginning with AUDR are issued by Audit Services to report actions taken during Receiver script processing.

AUDR001I Add User on Platform *platform_object*: eUser *eUser*, UID *uid*, Platform Association *platform_association*.

Explanation: An Add User was processed by the platform identified by *platform_object* for eUser *eUser*. The association *platform_association* was returned for the user. The Linux/UNIX UID number for the user is *uid*.

Action: None. Informational only.

AUDR002I Modify User on Platform *platform_object*: eUser *eUser*, UID *uid*, Platform Association *platform_association*.

Explanation: A Modify User was processed by the platform identified by *platform_object* for eUser *eUser*. The association for the user is *platform_association*. The Linux/UNIX UID number for the user is *uid*.

Action: None. Informational only.

AUDR003I Delete User on Platform *platform_object*: eUser *eUser*, Platform Association *platform_association*.

Explanation: A Delete User was processed by the platform identified by *platform_object* for eUser *eUser*. The association for the user was *platform_association*.

Action: None. Informational only.

AUDR004I Enable User on Platform *platform_object*: eUser *eUser*, Platform Association *platform_association*.

Explanation: An Enable User was processed by the platform identified by *platform_object* for eUser *eUser*. The association for the user is *platform_association*.

Action: None. Informational only.

AUDR005I Disable User on Platform *platform_object*: eUser *eUser*, Platform Association *platform_association*.

Explanation: A Disable User was processed by the platform identified by *platform_object* for eUser *eUser*. The association for the user is *platform_association*.

Action: None. Informational only.

AUDR006I Rename User on Platform *platform_object*: eUser *eUser*, Old Platform Association *old_platform_association*, New Platform Association *new_platform_association*.

Explanation: A Rename User was processed by the platform identified by *platform_object* for eUser *eUser*. The old association for the user was *old_platform_association*. The new association *new_platform_association* was returned for the user.

Action: None. Informational only.

AUDR007I Move User on Platform *platform_object*: eUser *eUser*, Old Platform Association *old_platform_association*, New Platform Association *new_platform_association*.

Explanation: A Move User was processed by the platform identified by *platform_object* for eUser *eUser*. The old association for the user was *old_platform_association*. The new association *new_platform_association* was returned for the user.

Action: None. Informational only.

AUDR008I Add User to Group on Platform *platform_object*: eUser *eUser*, eUser Platform Association *eUser_platform_association*, eGroup *eGroup*, eGroup Platform Association *eGroup_platform_association*.

Explanation: An Add User to Group was processed by the platform identified by *platform_object* for eUser *eUser*. The Group is *eGroup*. The association for the user is *eUser_platform_association*. The association for the group is *eGroup_platform_association*.

Action: None. Informational only.

AUDR009I Remove User from Group on Platform *platform_object*: eUser *eUser*, eUser Platform Association *eUser_platform_association*, eGroup *eGroup*, eGroup Platform Association *eGroup_platform_association*.

Explanation: A Remove User from Group was processed by the platform identified by *platform_object* for eUser *eUser*. The Group is *eGroup*. The association for the user is *eUser_platform_association*. The association for the group is *eGroup_platform_association*.

Action: None. Informational only.

AUDR010I Add Group on Platform *platform_object*: eGroup *eGroup*, GID *gid*, Platform Association *platform_association*.

Explanation: An Add Group was processed by the platform identified by *platform_object* for eGroup *eGroup*. The association *platform_association* was returned for the group. The Linux/UNIX GID number for the group is *gid*.

Action: None. Informational only.

AUDR011I Modify Group on Platform *platform_object*: eGroup *eGroup*, GID *gid*, Platform Association *platform_association*.

Explanation: A Modify Group was processed by the platform identified by *platform_object* for eGroup *eGroup*. The association for the group is *platform_association*. The Linux/UNIX GID number for the group is *gid*.

Action: None. Informational only.

AUDR012I Delete Group on Platform *platform_object*: eGroup *eGroup*, Platform Association *platform_association*.

Explanation: A Delete Group was processed by the platform identified by *platform_object* for eGroup *eGroup*. The association for the group was *platform_association*.

Action: None. Informational only.

AUDR013I Rename Group on Platform *platform_object*: eGroup *eGroup*, Old Platform Association *old_platform_association*, New Platform Association *new_platform_association*.

Explanation: A Rename Group was processed by the platform identified by *platform_object* for eGroup *eGroup*. The old association for the group was *old_platform_association*. The new association *new_platform_association* was returned for the group.

Action: None. Informational only.

AUDR014I Move Group on Platform *platform_object*: eGroup *eGroup*, Old Platform Association *old_platform_association*, New Platform Association *new_platform_association*.

Explanation: A Move Group was processed by the platform identified by *platform_object* for eGroup *eGroup*. The old association for the group was *old_platform_association*. The new association *new_platform_association* was returned for the group.

Action: None. Informational only.

AUDR015I Replicate Password on Platform *platform_object*: eUser *eUser*.

Explanation: A Replicate Password was processed by the platform identified by *platform_object* for eUser *eUser*.

Action: None. Informational only.

AUDR016E Add User failed on Platform *platform_object*: eUser *eUser*, UID *uid*.

Explanation: An Add User failed on the platform identified by *platform_object* for eUser *eUser*. The Linux/UNIX UID number for the user is *uid*.

Action: Examine the log on the platform to determine the cause of the failure, and take action as appropriate.

AUDR017E Modify User failed on Platform *platform_object*: eUser *eUser*, UID *uid*, Platform Association *platform_association*.

Explanation: A Modify User failed on the platform identified by *platform_object* for eUser *eUser*. The association for the user is *platform_association*. The Linux/UNIX UID number for the user is *uid*.

Action: Examine the log on the platform to determine the cause of the failure, and take action as appropriate.

AUDR018E Delete User failed on Platform *platform_object*: eUser *eUser*, Platform Association *platform_association*.

Explanation: A Delete User failed on the platform identified by *platform_object* for eUser *eUser*. The association for the user is *platform_association*.

Action: Examine the log on the platform to determine the cause of the failure, and take action as appropriate.

AUDR019E Enable User failed on Platform *platform_object*: eUser *eUser*, Platform Association *platform_association*.

Explanation: An Enable User failed on the platform identified by *platform_object* for eUser *eUser*. The association for the user is *platform_association*.

Action: Examine the log on the platform to determine the cause of the failure, and take action as appropriate.

AUDR020E Disable User failed on Platform *platform_object*: eUser *eUser*, Platform Association *platform_association*.

Explanation: A Disable User failed on the platform identified by *platform_object* for eUser *eUser*. The association for the user is *platform_association*.

Action: Examine the log on the platform to determine the cause of the failure, and take action as appropriate.

AUDR021E Rename User failed on Platform *platform_object*: eUser *eUser*, Old Platform Association *platform_association*.

Explanation: A Rename User failed on the platform identified by *platform_object* for eUser *eUser*. The association for the user is *platform_association*.

Action: Examine the log on the platform to determine the cause of the failure, and take action as appropriate.

AUDR022E Move User failed on Platform *platform_object*: eUser *eUser*, Old Platform Association *platform_association*.

Explanation: A Move User failed on the platform identified by *platform_object* for eUser *eUser*. The association for the user is *platform_association*.

Action: Examine the log on the platform to determine the cause of the failure, and take action as appropriate.

AUDR023E Add User to Group failed on Platform *platform_object*: eUser *eUser*, eUser Platform Association *eUser_platform_association*, eGroup *eGroup*, eGroup Platform Association *eGroup_platform_association*.

Explanation: An Add User to Group failed on the platform identified by *platform_object* for eUser *eUser*. The Group is *eGroup*. The association for the user is *eUser_platform_association*. The association for the group is *eGroup_platform_association*.

Action: Examine the log on the platform to determine the cause of the failure, and take action as appropriate.

AUDR024E Remove User from Group failed on Platform *platform_object*: eUser *eUser*, eUser Platform Association *eUser_platform_association*, eGroup *eGroup*, eGroup Platform Association *eGroup_platform_association*.

Explanation: A Remove User from Group failed on the platform identified by *platform_object* for eUser *eUser*. The Group is *eGroup*. The association for the user is *eUser_platform_association*. The association for the group is *eGroup_platform_association*.

Action: Examine the log on the platform to determine the cause of the failure, and take action as appropriate.

AUDR025E Add Group failed on Platform *platform_object*: eGroup *eGroup*, GID *gid*.

Explanation: An Add Group failed on the platform identified by *platform_object* for eGroup *eGroup*. The Linux/UNIX GID number for the group is *gid*.

Action: Examine the log on the platform to determine the cause of the failure, and take action as appropriate.

AUDR026E Modify Group failed on Platform *platform_object*: eGroup *eGroup*, GID *gid*, Platform Association *platform_association*.

Explanation: A Modify Group failed on the platform identified by *platform_object* for eGroup *eGroup*. The association for the group is *platform_association*. The Linux/UNIX GID number for the group is *gid*.

Action: Examine the log on the platform to determine the cause of the failure, and take action as appropriate.

AUDR027E Delete Group failed on Platform *platform_object*: eGroup *eGroup*, Platform Association *platform_association*.

Explanation: A Delete Group failed on the platform identified by *platform_object* for eGroup *eGroup*. The association for the group is *platform_association*.

Action: Examine the log on the platform to determine the cause of the failure, and take action as appropriate.

AUDR028E Rename Group failed on Platform *platform_object*: eGroup *eGroup*, Old Platform Association *platform_association*.

Explanation: A Rename Group failed on the platform identified by *platform_object* for eGroup *eGroup*. The association for the group is *platform_association*.

Action: Examine the log on the platform to determine the cause of the failure, and take action as appropriate.

AUDR029E Move Group failed on Platform *platform_object*: eGroup *eGroup*, Old Platform Association *platform_association*.

Explanation: A Move Group failed on the platform identified by *platform_object* for eGroup *eGroup*. The association for the group is *platform_association*.

Action: Examine the log on the platform to determine the cause of the failure, and take action as appropriate.

AUDR030E Replicate Password failed on Platform *platform_object*: eUser *eUser*.

Explanation: A Replicate Password failed on the platform identified by *platform_object* for eUser *eUser*.

Action: Examine the log on the platform to determine the cause of the failure, and take action as appropriate.

AUDR031I Pending Delete User on Platform *platform_object*: eUser *eUser*, Platform Association *platform_association*.

Explanation: A Pending Delete User was processed by the platform identified by *platform_object* for eUser *eUser*. The association for the user is *platform_association*.

Action: None. Informational only.

AUDR032I Pending Delete Group on Platform *platform_object*: eGroup *eGroup*, Platform Association *platform_association*.

Explanation: A Pending Delete Group was processed by the platform identified by *platform_object* for eGroup *eGroup*. The association for the group is *platform_association*.

Action: None. Informational only.

AUDR033E Pending Delete User failed on Platform *platform_object*: eUser *eUser*, Platform Association *platform_association*.

Explanation: A Pending Delete User failed on the platform identified by *platform_object* for eUser *eUser*. The association for the user is *platform_association*.

Action: Examine the log on the platform to determine the cause of the failure, and take action as appropriate.

AUDR034E Pending Delete Group failed on Platform *platform_object*: eGroup *eGroup*, Platform Association *platform_association*.

Explanation: A Pending Delete Group failed on the platform identified by *platform_object* for eGroup *eGroup*. The association for the group is *platform_association*.

Action: Examine the log on the platform to determine the cause of the failure, and take action as appropriate.

AUDR035I User user authentication result is returnCode (reasonString) [elapsedTime elapsed seconds].

Explanation: This message displays the result of an authentication attempt.

Possible Cause: This message is the result of an authentication request.

Action: None.

D.8 AXML Messages

Messages beginning with AXML are issued by the Core Driver during interactions with the Identity Manager engine.

AXML0000I Success.

Explanation: The action succeeded.

Action: No action is required.

AXML0006E The event could not be processed. The driver will retry the event.

Explanation: The event could not be processed because an error occurred during processing. The nature of the error could be transitory, so the event is retried.

Possible Cause: This error can be caused by configuration problems with the Core Driver.

Action: Examine the Core Driver logs to see if errors are being generated by the event. Use the Core Driver documentation to determine the cause of the error and how to correct it. After you correct the problem, the event should succeed when the Core Driver retries the event.

AXML0008W The driver is in discard-events mode and will not process events.

Explanation: Discard-Events mode is used when you do not want directory events to be processed by the Core Driver. This can be useful if you have a large backlog of directory events. The driver discards directory events.

Possible Cause: The Discard Events driver parameter of the Driver object is set to true.

Action: To return to normal mode, open the configuration parameters of the Driver driver object in iManager, and select the Driver Parameters. Change the Discard Events parameter to false.

AXML0012W Some initialization parameters could not located; default values are being used.

Explanation: Default values for either the ASAM root directory or the Locale, or both are being used. The default directory depends on your operating system. The default locale is en for English. This could cause problems if these values are not valid for your installation.

Possible Cause: The ASAM root directory or Locale values were left blank in the Driver object parameters.

Action: If you do not want to use default values, open the Core Driver Driver object in iManager, click the Driver Parameters tab, and change the parameters to the desired values.

AXML0013E The event for object *dn* failed with error code *code*. The event has been discarded.

Explanation: The event could not be processed because an error occurred in the Core Driver. This error could not be corrected, so the Core Driver has discarded the event.

Possible Cause: This error can be caused by configuration problems with the Core Driver.

Action: The error code usually corresponds to an LDAP error. Some more common LDAP errors and suggested actions follow.

3, 85 - Time-out. Increase the LDAP time-out value in the Web interface.

16 - No such attribute. The system attempted to access an attribute that was not present on an eDirectory object.

17 - Type not found. The schema might not have been correctly updated.

32 - Object not found. The system attempted to access an eDirectory object that was not present.

49 - Invalid credentials. Check the username and password in the Driver object parameters.

51, 52 - Busy/unavailable. Check the health of your LDAP server using DSTrace.

81 - Server down. Restart your LDAP server, or check network connectivity between the Core Driver server and the LDAP server.

For a full list of eDirectory errors, see your eDirectory documentation.

An error of -1 is an internal error. In this case, and for all errors, examine your log files for more information about the error.

AXML0014E No GUID could be found for the event.

Explanation: The GUID attribute for the event was not present in the XML document sent by the Identity Manager engine.

Possible Cause: The GUID attribute might not be enabled in the driver Subscriber filter.

Action: Make sure the GUID attribute is enabled for Aliases, Users, Groups, and Organizational Roles in the driver Subscriber filter.

AXML0015E Could not retrieve the LDAP attribute map. The ASAM Master User and Password driver parameters might be invalid, or the specified user does not have sufficient rights.

Explanation: The Core Driver attempts to load a mapping of LDAP attribute names to eDirectory attribute names. This message is issued when the driver cannot load this mapping. The driver cannot start if it cannot read the attribute map.

Possible Cause: The LDAP Host and Port driver parameter might be invalid.

The ASAM Master User and Password driver parameters might be invalid. By default, a user named ASAMMaster is used to log in to eDirectory with an installation-generated password.

eDirectory or LDAP on the specified server might be down or in an error state.

Action: Check the LDAP Host and Port Driver object parameters, and verify that LDAP is running on the specified host and port. If a DNS name is specified, verify that DNS is working on the Core Driver host server.

Check the ASAM Master User and Password parameters to make sure a valid user and password are specified. Also, make sure the user has sufficient rights.

Verify that eDirectory and LDAP are healthy on the specified server.

D.9 CFG Messages

Messages beginning with CFG are issued by Platform Configuration file processing.

CFG001E Could not open configuration file *filename*.

Explanation: Could not open the configuration file.

Possible Cause: There are several possible causes for this error.

The file does not exist. The default location for the file is in the ASAM\data directory. The file path can be specified by using the -a command line option.

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 typed wrong 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 parse 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.

D.10 CFGA Messages

Messages beginning with CFGA are issued during installation when migrating values from the `asamcore.conf` file to Driver object configuration parameters.

CFGA001E Invalid ASAM System Container configuration.

Explanation: The ASAM System Container Driver object configuration parameter is not valid.

Action: Correct the parameter.

CFGA002E Invalid Entropy configuration.

Explanation: The Entropy Driver object configuration parameter is not valid.

Action: Correct the parameter.

CFGA003E Invalid ASAM Master User configuration.

Explanation: The ASAM Master User Driver object configuration parameter is not valid.

Action: Correct the parameter.

CFGA004E Invalid ASAM Master User Password configuration.

Explanation: The ASAM Master User Password Driver object configuration parameter is not valid.

Action: Correct the parameter.

CFGA005E Invalid LDAP Host and Port configuration.

Explanation: The LDAP Host and Port Driver object configuration parameter is not valid.

Action: Correct the parameter.

CFGA006E Invalid Locale configuration.

Explanation: The Locale Driver object configuration parameter is not valid.

Action: Correct the parameter.

CFGA007E Invalid ASAM Directory configuration.

Explanation: The ASAM Directory Driver object configuration parameter is not valid.

Action: Correct the parameter.

CFGA008E Invalid Debug Log File configuration.

Explanation: The Debug Log File Driver object configuration parameter is not valid.

Action: Correct the parameter.

CFGA009E Invalid Syslog Facility configuration.

Explanation: The Syslog Facility Driver object configuration parameter is not valid.

Action: Correct the parameter.

CFGA010E Invalid Storage Key configuration.

Explanation: The Storage Key Driver object configuration parameter is not valid.

Action: Correct the parameter.

D.11 CFGP Messages

Messages beginning with CFGP are issued by platform configuration file processing.

CFGP001E Invalid *statement_name* statement.

Explanation: The *statement_name* statement is not valid.

Action: Correct the statement.

CFGP002I There are no Core Drivers configured for provisioning. If you want to provision to this platform, specify a PROVISIONING statement.

Explanation: No PROVISIONING statement was found in the platform configuration file.

Possible Cause: None was coded.

Action: If you want to provision users and groups to this platform, add a PROVISIONING statement to the platform configuration file.

CFGP003I There are no Core Drivers configured for authentication. If you want to use authentication redirection or APIs on this platform, specify an AUTHENTICATION statement.

Explanation: No AUTHENTICATION statement was found in the platform configuration file.

Possible Cause: None was coded.

Action: If you want to allow authentication redirection for this platform, add an AUTHENTICATION statement to the platform configuration file.

D.12 CRT Messages

Messages beginning with CRT are issued by Certificate Services.

CRT001E Error: Certificate Authority not found.

Explanation: The certificate authority could not be found.

Possible Cause: The Core Driver was not properly installed, or the certificate authority is damaged, missing, or in the wrong location.

Action: Verify that the Core Driver is properly installed and that its files are not damaged.

CRT002E Error: Could not contact directory. Check username and password.

Explanation: The username/password provided for basic authentication failed.

Possible Cause: The username and password specified in response to a prompt are incorrect.

The ASAM Master User and ASAM Master User Password are not correct.

Action: Ensure that the fully distinguished username and password are specified correctly.

Ensure that the ASAM Master User and ASAM Master User Password are specified correctly.

CRT003E Error: Certificate Services not properly configured.

Explanation: The Certificate Services configuration object and its attributes were not found.

Possible Cause: The Core Driver installation did not complete properly.

The Core Driver configuration specifies the wrong ASAM System OU.

Action: Verify that the Core Driver installation completed normally.

Verify that the ASAM System Container Core Driver parameter is correct.

CRT004E Error: *component_name* not properly configured.

Explanation: Configuration information for *component_name* is missing or incomplete.

Possible Cause: The administrator did not create and complete the proper component configuration using the Web interface.

Action: Examine the configuration object for the component with the Web interface. Provide any missing information, such as network address.

CRT005E Error: Internal Server Error.

Explanation: The Core Driver encountered an unknown error, such as out of memory or memory allocation failure.

Action: Ensure that sufficient memory is available.

CRT006E Error: Insufficient rights to create *component_name* configuration object.

Explanation: You do not have sufficient rights to create the component configuration object.

Action: Obtain sufficient rights to the ASAM System container.

CRT007E Error: Insufficient rights to modify *component_name* configuration object.

Explanation: You do not have sufficient rights to modify the component configuration object.

Action: Obtain sufficient rights to the ASAM System container.

CRT008I All certificate and host information has been checked and verified successfully.

Explanation: The certificate autocheck procedure has determined that all certificates for this particular driver have been located and include the correct host information.

Action: None. Informational only.

CRT009I Certificates have been updated with new host information.

Explanation: The certificate autocheck procedure has determined that the certificates for this driver are not current with the host information provided by the Fan-Out system. Therefore, new certificates have been created to include the correct host information.

Possible Cause: This driver might have been moved to another server, the server might have had a network configuration change, or the administrator might have added new host address information for this host.

Action: Use the Web interface to ensure that the correct host information is specified.

CRT010I New driver certificates were created.

Explanation: The certificate autocheck procedure was unable to locate an existing certificate for this driver. A new certificate authority was generated, along with a new certificate containing host information provided by the Fan-Out system.

Possible Cause: This can be caused by a new installation or upgrade.

Action: If this is not the expected behavior, check the file system under ASAM/CoreDriver/certs/ for an existing certificate authority and driver certificates. Make sure that the driver has appropriate access to these files.

CRT011I The certificate authority was retrieved successfully from the primary Core Driver.

Explanation: The certificate autocheck procedure was unable to locate a certificate authority and requested the information from the primary Core Driver. Upon retrieving the data successfully, new certificates were created for this driver with appropriate host information.

Possible Cause: This can result from a new installation or upgrade of a secondary Core Driver.

Action: If this behavior is not expected, check ASAM/CoreDriver/certs/ for existing certificates, and make sure that the driver is configured properly as a primary or secondary driver.

D.13 DIR Messages

Messages beginning with DIR are issued by the Core Driver during LDAP directory access.

DIR001E Attribute Not Supported.

Explanation: A call was made to the API routine to read the value of an attribute for an object, but the attribute specified is not supported. Only the Home Directory attribute is supported.

Action: Correct the API call in the application program.

DIR002E Request Build Error.

Explanation: The directory interface routine was unable to create a request to perform a directory action. This is an internal error.

Action: Examine the log for related messages.

DIR003D Error.

Explanation: This is a general error indication. This message is accompanied by other messages that provide additional details.

Action: Examine the log for related messages.

DIR004D Success.

Explanation: A directory operation was successful.

Action: No action is required.

DIR005D Operations Error.

Explanation: An LDAP operation returned LDAP_OPERATIONS_ERROR. This indicates an internal error. The server is unable to respond with a more specific error and is also unable to properly respond to a request. It does not indicate that the client has sent an erroneous message.

Action: Examine the log for related messages.

DIR006D Protocol Error.

Explanation: An LDAP operation returned LDAP_PROTOCOL_ERROR. This indicates that the server has received an invalid or malformed request from the client.

Action: Examine the log for related messages.

DIR007D Time Limit Exceeded.

Explanation: An LDAP operation returned LDAP_TIMELIMIT_EXCEEDED. This indicates that the operation's time limit specified by either the client or the server has been exceeded. On search operations, incomplete results are returned.

Action: Examine the log for related messages. Check the health of the server hosting LDAP.

DIR008D Size Limit Exceeded.

Explanation: An LDAP operation returned LDAP_SIZELIMIT_EXCEEDED. This indicates that in a search operation, the size limit specified by the client or the server has been exceeded. Incomplete results are returned.

Action: Examine the log for related messages.

DIR009D Compare False.

Explanation: An LDAP operation returned LDAP_COMPARE_FALSE. This does not indicate an error condition. It indicates that the results of a compare operation are false.

Action: No action is required.

DIR010D Compare True.

Explanation: An LDAP operation returned LDAP_COMPARE_TRUE. This does not indicate an error condition. It indicates that the results of a compare operation are true.

Action: No action is required.

DIR011D Authentication Method Not Supported.

Explanation: An LDAP operation returned LDAP_AUTH_METHOD_NOT_SUPPORTED. This indicates that during a bind operation the client requested an authentication method not supported by the LDAP server.

Action: Examine the log for related messages. Make sure your LDAP server is running the most current version.

DIR012D Strong Authentication Required.

Explanation: An LDAP operation returned LDAP_STRONG_AUTH_REQUIRED. This indicates one of the following:

In bind requests, the LDAP server accepts only strong authentication.

In a client request, the client requested an operation, such as delete, that requires strong authentication.

In an unsolicited notice of disconnection, the LDAP server discovers the security protecting the communication between the client and server has unexpectedly failed or been compromised.

Possible Cause: LDAPHOST port set to the unencrypted port 289 instead of the default of 636.

Action: Examine the log for related messages. Make sure your LDAP server is running the most current version.

DIR013D Partial Results.

Explanation: An LDAP operation returned LDAP_PARTIAL_RESULTS. This should not occur. The server should return LDAP_REFERRAL instead.

Action: Examine the log for related messages.

DIR014D Referral.

Explanation: An LDAP operation returned LDAP_REFERRAL. This does not indicate an error condition. In LDAPv3, it indicates that the server does not hold the target entry of the request, but that the servers in the referral field might hold the target.

Action: No action is required.

DIR015D Admin Limit Exceeded.

Explanation: An LDAP operation returned LDAP_ADMINLIMIT_EXCEEDED. This indicates that an LDAP server limit set by an administrative authority has been exceeded.

Action: Examine the log for related messages. Check the health of the server hosting LDAP.

DIR016D Unavailable Critical Extension.

Explanation: An LDAP operation returned LDAP_UNAVAILABLE_CRITICAL_EXTENSION. This indicates that the LDAP server was unable to satisfy a request because one or more critical extensions were not available. Either the server does not support the control or the control is not appropriate for the operation type.

Action: Examine the log for related messages. Make sure your LDAP server is running the most current version.

DIR017D Confidentiality Required.

Explanation: An LDAP operation returned LDAP_CONFIDENTIALITY_REQUIRED. This indicates that the session is not protected by a protocol such as Transport Layer Security (TLS), which provides session confidentiality.

Action: Examine the log for related messages. Make sure your LDAP server is running the most current version.

DIR018D SASL Bind in Progress.

Explanation: An LDAP operation returned LDAP_SASL_BIND_IN_PROGRESS. This does not indicate an error condition, but indicates that the server is ready for the next step in the process. The client must send the server the same SASL mechanism to continue the process.

Action: No action is required.

DIR019D No Such Attribute.

Explanation: An LDAP operation returned LDAP_NO_SUCH_ATTRIBUTE. This indicates that the attribute specified in the modify or compare operation does not exist in the entry.

Action: Examine the log for related messages. Many times this requires no action.

DIR020D Undefined Type.

Explanation: An LDAP operation returned LDAP_UNDEFINED_TYPE. This indicates that the attribute specified in the modify or add operation does not exist in the LDAP server's schema.

Action: Make sure the schema has been properly extended.

DIR021D Inappropriate Matching.

Explanation: An LDAP operation returned LDAP_INAPPROPRIATE_MATCHING. This indicates that the matching rule specified in the search filter does not match a rule defined for the attribute's syntax.

Action: Examine the log for related messages.

DIR022D Constraint Violation.

Explanation: An LDAP operation returned LDAP_CONSTRAINT_VIOLATION. This indicates that the attribute value specified in a modify, add, or modify DN operation violates constraints placed on the attribute. The constraint can be one of size or content (string only, no binary).

Possible Cause: Password rules, such as uniqueness and length, are violated.

Action: Examine the log for related messages.

DIR023D Type or Value Exists.

Explanation: An LDAP operation returned LDAP_TYPE_OR_VALUE_EXISTS. This indicates that the attribute value specified in a modify or add operation already exists as a value for that attribute.

Action: Examine the log for related messages. This might not require any action.

DIR024D Invalid Syntax.

Explanation: An LDAP operation returned LDAP_INVALID_SYNTAX. This indicates that the attribute value specified in an add, compare, or modify operation is an unrecognized or invalid syntax for the attribute.

Action: Examine the log for related messages.

DIR025D No Such Object.

Explanation: An LDAP operation returned LDAP_NO_SUCH_OBJECT. This indicates the target object cannot be found. This code is not returned on the following operations:

Search operations that find the search base but cannot find any entries that match the search filter.

Bind operations.

Action: Examine the log for related messages. Make sure the application is installed and configured correctly.

DIR026D Alias Problem.

Explanation: An LDAP operation returned LDAP_ALIAS_PROBLEM. This indicates that an error occurred when an alias was dereferenced.

Action: Examine the log for related messages. Check the server health of the LDAP host.

DIR027D Invalid DN Syntax.

Explanation: An LDAP operation returned LDAP_INVALID_DN_SYNTAX. This indicates that the syntax of the DN is incorrect. (If the DN syntax is correct, but the LDAP server's structure rules do not permit the operation, the server returns LDAP_UNWILLING_TO_PERFORM.)

Action: Examine the log for related messages.

DIR028D Is Leaf.

Explanation: An LDAP operation returned LDAP_IS_LEAF. This indicates that the specified operation cannot be performed on a leaf entry.

Action: Examine the log for related messages.

DIR029D Alias Dereference Problem.

Explanation: An LDAP operation returned LDAP_ALIAS_DEREF_PROBLEM. This indicates that during a search operation, either the client does not have access rights to read the aliased object's name or dereferencing is not allowed.

Action: Examine the log for related messages. Check the health of the LDAP host.

DIR030D Inappropriate Authentication.

Explanation: An LDAP operation returned LDAP_INAPPROPRIATE_AUTH. This indicates that during a bind operation, the client is attempting to use an authentication method that the client cannot use correctly. For example, the following can cause this error:

The client returns simple credentials when strong credentials are required.

The client returns a DN and a password for a simple bind when the entry does not have a password defined.

Action: Examine the log for related messages.

DIR031D Invalid Credentials.

Explanation: An LDAP operation returned LDAP_INVALID_CREDENTIALS. This indicates that during a bind operation one of the following occurred:

The client passed either an incorrect DN or password.

The password is incorrect because it has expired, intruder detection has locked the account, or some other similar reason.

Action: Examine the log for related messages.

DIR032D Insufficient Access.

Explanation: An LDAP operation returned LDAP_INSUFFICIENT_ACCESS. This indicates that the caller does not have sufficient rights to perform the requested operation.

Action: Examine the log for related messages.

DIR033D Busy.

Explanation: An LDAP operation returned LDAP_BUSY. This indicates that the LDAP server is too busy to process the client request at this time, but if the client waits and resubmits the request, the server might be able to process it later.

Action: Examine the log for related messages. Check the health of the LDAP server.

DIR034D Unavailable.

Explanation: An LDAP operation returned LDAP_UNAVAILABLE. This indicates that the LDAP server cannot process the client's bind request, usually because it is shutting down.

Action: Examine the log for related messages. Check the LDAP server's health.

DIR035D Unwilling to Perform.

Explanation: An LDAP operation returned LDAP_UNWILLING_TO_PERFORM. This indicates that the LDAP server cannot process the request because of server-defined restrictions. This error is returned for the following reasons:

The add entry request violates the server's structure rules.

The modify attribute request specifies attributes that users cannot modify.

Password restrictions prevent the action.

Connection restrictions prevent the action.

Action: Examine the log for related messages.

DIR036D Loop Detected.

Explanation: An LDAP operation returned LDAP_LOOP_DETECT. This indicates that the client discovered an alias or referral loop, and is thus unable to complete this request.

Action: Examine the log for related messages.

DIR037D Naming Violation.

Explanation: An LDAP operation returned LDAP_NAMING_VIOLATION. This indicates that the add or modify DN operation violates the schema's structure rules. For example:

The request places the entry subordinate to an alias.

The request places the entry subordinate to a container that is forbidden by the containment rules.

The RDN for the entry uses a forbidden attribute type.

Action: Examine the log for related messages.

DIR038D Object Class Violation.

Explanation: An LDAP operation returned LDAP_OBJECT_CLASS_VIOLATION. This indicates that the add, modify, or modify DN operation violates the object class rules for the entry. For example, the following types of request return this error:

The add or modify operation tries to add an entry without a value for a required attribute.

The add or modify operation tries to add an entry with a value for an attribute that the class definition does not contain.

The modify operation tries to remove a required attribute without removing the auxiliary class that defines the attribute as required.

Action: Examine the log for related messages.

DIR039D Not Allowed on Non Leaf Object.

Explanation: An LDAP operation returned LDAP_NOT_ALLOWED_ON_NONLEAF. This indicates that the requested operation is permitted only on leaf entries. For example, the following types of requests return this error:

The client requests a delete operation on a parent entry.

The client requests a modify DN operation on a parent entry.

Action: Examine the log for related messages.

DIR040D Not Allowed on RDN (Relative Distinguished Name).

Explanation: An LDAP operation returned LDAP_NOT_ALLOWED_ON_RDN. This indicates that the modify operation attempted to remove an attribute value that forms the entry's relative distinguished name.

Action: Examine the log for related messages.

DIR041D Already Exists.

Explanation: An LDAP operation returned LDAP_ALREADY_EXISTS. This indicates that the add operation attempted to add an entry that already exists, or that the modify operation attempted to rename an entry to the name of an entry that already exists.

Action: Examine the log for related messages. This message might not require any action.

DIR042D No Object Class Modifications.

Explanation: An LDAP operation returned LDAP_NO_OBJECT_CLASS_MODS. This indicates that the modify operation attempted to modify the structure rules of an object class.

Action: Examine the log for related messages.

DIR043D Results Too Large.

Explanation: An LDAP operation returned LDAP_RESULTS_TOO_LARGE. This indicates that the results of the request are too large.

Action: Examine the log for related messages.

DIR044D Affects Multiple DSAS.

Explanation: An LDAP operation returned LDAP_AFFECTS_MULTIPLE_DSAS. This indicates that the modify DN operation moves the entry from one LDAP server to another and thus requires more than one LDAP server.

Action: Examine the log for related messages.

DIR045D Other.

Explanation: An LDAP operation returned LDAP_OTHER. This indicates an unknown error condition. This is the default value for error codes that do not map to other LDAP error codes.

Action: Examine the log for related messages.

Use DSTRACE to gather more specific error information.

DIR046D Server Down.

Explanation: An LDAP operation returned LDAP_SERVER_DOWN. This indicates that the LDAP libraries cannot establish an initial connection with the LDAP server. Either the LDAP server is down or the specified host name or port number is incorrect.

Action: Examine the log for related messages. Check LDAP server health.

DIR047D Local Error.

Explanation: An LDAP operation returned LDAP_LOCAL_ERROR. This indicates that the LDAP client has an error. This is usually a failed dynamic memory allocation error.

Action: Examine the log for related messages. Check LDAP server health.

DIR048D Encoding Error.

Explanation: An LDAP operation returned LDAP_ENCODING_ERROR. This indicates that the LDAP client encountered errors when encoding an LDAP request intended for the LDAP server.

Action: Examine the log for related messages. Check LDAP server health.

DIR049D Decoding Error.

Explanation: An LDAP operation returned LDAP_DECODING_ERROR. This indicates that the LDAP client encountered errors when decoding an LDAP response from the LDAP server.

Action: Examine the log for related messages.

DIR050D Time Out.

Explanation: An LDAP operation returned LDAP_TIMEOUT. This indicates that the time limit of the LDAP client was exceeded while waiting for a result.

Action: Examine the log for related messages. Check LDAP server health.

DIR051D Authentication Unknown.

Explanation: An LDAP operation returned LDAP_AUTH_UNKNOWN. This indicates that the ldap_bind or ldap_bind_s function was called with an unknown authentication method.

Action: Examine the log for related messages.

DIR052D Filter Error.

Explanation: An LDAP operation returned LDAP_FILTER_ERROR. This indicates that the ldap_search function was called with an invalid search filter.

Action: Examine the log for related messages.

DIR053D User Cancelled.

Explanation: An LDAP operation returned LDAP_USER_CANCELLED. This indicates that the user cancelled the LDAP operation.

Action: Examine the log for related messages.

DIR054D Parameter Error.

Explanation: An LDAP operation returned LDAP_PARAM_ERROR. This indicates that an LDAP function was called with an invalid parameter value (for example, the ld parameter is NULL).

Action: Examine the log for related messages.

DIR055D No Memory.

Explanation: An LDAP operation returned LDAP_NO_MEMORY. This indicates that a dynamic memory allocation function failed when calling an LDAP function.

Action: Examine the log for related messages. Check LDAP server health.

DIR056D Connect Error.

Explanation: An LDAP operation returned LDAP_CONNECT_ERROR. This indicates that the LDAP client has either lost its connection or cannot establish a connection to the LDAP server.

Action: Examine the log for related messages.

DIR057D Not Supported.

Explanation: An LDAP operation returned LDAP_NOT_SUPPORTED. This indicates that the requested functionality is not supported by the client. For example, if the LDAP client is established as an LDAPv2 client, the libraries return this error code when the client requests LDAPv3 functionality.

Action: Examine the log for related messages.

DIR058D Control Not Found.

Explanation: An LDAP operation returned LDAP_CONTROL_NOT_FOUND. This indicates that the client requested a control that the libraries cannot find in the list of supported controls sent by the LDAP server.

Action: Examine the log for related messages.

DIR059D No Results Returned.

Explanation: An LDAP operation returned LDAP_NO_RESULTS_RETURNED. This indicates that the LDAP server sent no results. When the ldap_parse_result function is called, no result code is included in the server's response.

Action: Examine the log for related messages.

DIR060D More Results to Return.

Explanation: An LDAP operation returned LDAP_MORE_RESULTS_TO_RETURN. This indicates that more results are chained in the result message. The libraries return this code when the call to the ldap_parse_result function reveals that additional result codes are available.

Action: Examine the log for related messages.

DIR061D Client Loop.

Explanation: An LDAP operation returned LDAP_CLIENT_LOOP. This indicates the LDAP libraries detected a loop. Usually this happens when following referrals.

Action: Examine the log for related messages.

DIR062D Referral Limit Exceeded.

Explanation: An LDAP operation returned LDAP_REFERRAL_LIMIT_EXCEEDED. This indicates that the referral exceeds the hop limit. The hop limit determines how many servers the client can hop through to retrieve data. For example, assume the following conditions:

The hop limit is two.

The referral is to server D, which can be contacted only through server B (1 hop) which contacts server C (2 hops) which contacts server D (3 hops)

With these conditions, the hop limit is exceeded and the LDAP libraries return this code.

Action: Examine the log for related messages.

DIR063D No Such Object.

Explanation: A call was made to the API routine to determine if a user has security equivalence to an object, but the object does not exist.

Action: This can be normal. The application should handle this as appropriate.

DIR064D Invalid Argument.

Explanation: An argument to a directory routine was not valid.

Action: Examine the log for related messages.

DIR065D Revoked.

Explanation: In a directory operation involving a User object, the user was found to have the login disabled flag set.

Action: Examine the log for related messages and handle the event as appropriate.

DIR066W Unable to connect to LDAP. Component will retry connection periodically.

Explanation: An attempt to connect to the configured LDAP server failed. The component issuing this message periodically retries the connection. When the connection is successful, the component continues processing.

Possible Cause: The configured LDAP server is not started or is unreachable.

Action: Make sure that an LDAP server is running at the configured LDAP host and port.

DIR067W Directory Services returned *rc*.

Explanation: An LDAP error occurred. The LDAP return code is given by *rc*.

Action: Check LDAP server health.

DIR068E LDAP Server *server* is not responding correctly. RC = *rc*.

Explanation: The LDAP server specified by the LDAP Host and Port Driver object configuration parameter is not responding to a search request on the ASAM System container.

Action: Restart the LDAP server and make sure LDAP services are available.

DIR069I LDAP Server is now responding to requests.

Explanation: The LDAP server specified by the LDAP Host and Port Driver object configuration parameter is now up and responding correctly to requests.

Action: None.

D.14 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 trying to parse 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.

D.15 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.

D.16 EJS Messages

Messages beginning with EJS are issued by Event Journal Services.

EJS0001E No Platform object FDN was provided with the Platform Receiver request.

Explanation: The Platform object FDN was missing from the Platform Receiver request. The Platform object FDN is required for every Platform Receiver request and is used to identify the corresponding Platform object in eDirectory.

Possible Cause: The security certificate was not found by the Platform Receiver, or an invalid security certificate is installed.

Action: Install a security certificate on the platform.

EJS0002E Unable to create an instance of the string handler.

Explanation: An instance of the string handler object could not be created.

Possible Cause: There might not be enough free memory available on the system.

Action: Ensure that there is adequate free memory available.

EJS0003E Unable to create an instance of the memory manager.

Explanation: An instance of the memory manager object could not be created.

Possible Cause: There might not be enough free memory available on the system.

Action: Ensure that there is adequate free memory available.

EJS0004E Unable to create an instance of the ASAM directory interface, direrr=*DirectoryError (DirectoryErrorText)*.

Explanation: An instance of the ASAM directory interface object could not be created.

Possible Cause: There are several possible causes of this error:

An invalid ASAM Master User or ASAM Master User Password is specified in the Driver object parameters.

An invalid DNS name or IP address, or port number is specified for the LDAP Host and Port in the Driver object parameters.

The LDAP host is down or not responding to requests.

Action: Ensure that the correct ASAM Master User Password is specified in the Driver object parameters.

Ensure that the correct network address and port is specified for the LDAP Host and Port in the Driver object parameters.

Ensure that the host running the LDAP server is functioning correctly.

EJS0005E Directory Search object *DirectorySearchObjectFDN* not found with scope= *DirectorySearchScopeLevel*.

Explanation: The object was not found using the specified directory search scope. The scope can be one of the following values: *DirectoryScopeBase*, *DirectoryScopeOneLevel*, *DirectoryScopeSubtree*.

This message is accompanied by messages EJS0007W and EJS0008W.

Possible Cause: The search did not find any results that matched the search criteria.

Action: No action is required.

EJS0006E Directory search error for object *DirectorySearchObjectFDN*, direrr= *DirectoryError (DirectoryErrorText)*, numRows= *DirectoryEntriesReturned*, scope= *DirectorySearchScopeLevel*.

Explanation: An error occurred while searching for the specified object. This message is accompanied by messages EJS0007W and EJS0008W.

Possible Cause: See the direrr value to determine the cause of the error.

Action: Correct the cause of the error and retry the Platform Receiver request.

EJS0007W Directory search requested attributes=*DirectoryAttributes*.

Explanation: This message shows the attributes that were requested for the search. This message is accompanied by messages EJS0005E, or EJS0006E, and EJS0008W.

Action: No action is required.

EJS0008W Directory search for values= *DirectorySearchValues*.

Explanation: This message shows the matching criteria for the search request.

Action: No action is required.

EJS0009E Directory modification error for object *DirectoryObjectFDN*, direrr=*DirectoryError* (*DirectoryErrorText*), actions= *ActionsToPerform*.

Explanation: An error occurred while trying to modify an attribute value for the specified object. The displayed actions are the actions and attributes that were to be modified.

Possible Cause: See the direrr value to determine the cause of the error.

Action: Correct the cause of the error and retry the Platform Receiver request.

EJS0010E Directory modification error for object *DirectoryObjectFDN*, direrr=*DirectoryError* (*DirectoryErrorText*).

Explanation: An error occurred while trying to modify an attribute value for the specified object.

Possible Cause: See the direrr value to determine the cause of the error.

Action: Correct the cause of the error and retry the Platform Receiver request.

EJS0011E Unable to create or obtain the Event Journal Services Platform item.

Explanation: An instance of the Event Journal Service item could not be created.

Possible Cause: There are several possible causes of this error:

There might not be enough free memory available on the system.

A string handler interface could not be created (look for message EJS0002E).

A memory manager interface could not be created (look for message EJS0003E).

An ASAM directory interface object could not be created (look for message EJS004E).

The Platform FDN provided by the Platform Receiver is invalid (look for message EJS0031E).

Action: Ensure that there is adequate free memory available. Perform the actions for any additional messages that were issued.

EJS0012W Event *EventType* for object *DirectoryObjectFDN* could not be processed.

Explanation: An event for the specified Platform FDN could not be processed.

Possible Cause: Required information needed to process the event was not found. If this was a change password event, the Platform object does not have the Permit Password Replication attribute enabled.

Action: Look for other messages beginning with the EJS prefix to determine what information that is needed to process this event is missing.

EJS0013W Unable to obtain UID/GID information for object *DirectoryObjectFDN*.

Explanation: No UID or GID information exists for the specified object.

Possible Cause: The specified object has no corresponding UID/GID object in the UID/GID Set for the platform, or the UID/GID object contains no value for the UIDGIDNumber attribute.

Action: Ensure that a UID/GID Set is defined for the Platform Set. Run a Trawl to create the appropriate UID/GID objects.

EJS0014E Unable to create a directory search request.

Explanation: A directory search request object could not be created.

Possible Cause: There might not be enough free memory available on the system.

Action: Ensure that there is adequate free memory available.

EJS0015E Unable to delete attribute *DirectoryAttribute* for object *DirectoryObjectFDN*.

Explanation: The attribute could not be deleted for the specified object.

Possible Cause: There might not be enough free memory available on the system.

Action: Ensure that there is adequate free memory available.

EJS0016E Unable to add attribute *DirectoryAttribute* with value *AttributeValue* for object *DirectoryObjectFDN*.

Explanation: The attribute could not be added for the specified object.

Possible Cause: There might not be enough free memory available on the system.

Action: Ensure that there is adequate free memory available.

EJS0017E Unable to create a directory modify attributes request.

Explanation: A directory modification request object could not be created.

Possible Cause: There might not be enough free memory available on the system.

Action: Ensure that there is adequate free memory available.

EJS0018E Unable to delete value *AttributeValue* for attribute *DirectoryAttribute* for object *DirectoryObjectFDN*.

Explanation: The attribute value could not be deleted for the specified object.

Possible Cause: There might not be enough free memory available on the system.

Action: Ensure that there is adequate free memory available.

EJS0019E Unable to replace attribute *DirectoryAttribute* value *OldAttributeValue* with new value *NewAttributeValue* for object *DirectoryObjectFDN*.

Explanation: The attribute value could not be replaced by the new value for the specified object.

Possible Cause: There might not be enough free memory available on the system.

Action: Ensure that there is adequate free memory available.

EJS0020E Unable to obtain the CN of the Platform object *DirectoryObjectFDN*.

Explanation: No common name attribute exists for the specified object.

Possible Cause: This situation should not occur under normal circumstances.

Action: Collect diagnostic information and contact Support.

EJS0021E No Census object was found for the Platform object *DirectoryObjectFDN*.

Explanation: No corresponding Census object was found for the specified object.

Possible Cause: This situation should not occur under normal circumstances.

Action: Collect diagnostic information and contact Support.

EJS0022E Unable to parse the journal value for object *DirectoryObjectFDN*.

Explanation: The events could not be parsed for the specified object.

Possible Cause: This situation should not occur under normal circumstances.

Action: Collect diagnostic information and contact Support.

EJS0023I No UID/GID number was found for object *DirectoryObjectFDN*.

Explanation: The UID/GID number attribute was not found for the specified object.

Possible Cause: The Platform Set that contains the associated user or group object might not have a UID/GID Set defined.

Action: Ensure that the Platform Set that contains the associated User or Group object has a UID/GID Set defined.

EJS0024W No Platform Receiver attribute list was loaded for object class *DirectoryObjectClass*.

Explanation: The Platform Receiver attribute list was not loaded for the specified object class.

Possible Cause: No attributes that are to be sent to the Platform Receivers are defined for the specified object type.

The LDAP server is down or not responding properly.

Action: Add the appropriate attributes to the Subscriber filter.

EJS0026W The password could not be retrieved for object *DirectoryObjectFDN*.

Explanation: The object's password could not be retrieved.

Possible Cause: The object might not have the old password set in ePassword.

Action: None. Normal processing continues.

EJS0029E *ElementTagName* SOAP element could not be created in the Platform response document.

Explanation: The specified SOAP structure element tag name could not be created.

Possible Cause: There might not be enough free memory available on the system.

Action: Ensure that there is adequate free memory available.

EJS0031E Invalid Platform FDN *platformFDN* was specified by the Platform Receiver.

Explanation: The Platform FDN provided in the Platform Receiver request was invalid.

Possible Cause: The object referenced by the FDN does not exist in eDirectory.

Action: Ensure that the correct security certificate has been installed on the platform.
Also ensure that the Platform object has been created in eDirectory.

EJS0032E Unable to search for pending events for Platform *platformFDN*.

Explanation: The search criteria was empty for the search request during a Polling or Persistent Mode request. The search for events is not performed.

Possible Cause: This situation should not occur under normal circumstances.

Action: Collect diagnostic information and contact Support.

EJS0033I Platform *PlatformName* returned *ReturnCode* for event *EventType* for object *DirectoryObjectFDN*.

Explanation: The Platform Receiver running on the specified platform returned the return code after processing the event for the specified object. The possible return codes values are:

prrcSuccess - The event was successfully processed by the Platform Receiver.

prrcIgnored - The event was ignored by the Platform Receiver.

prrcExcluded - The event was excluded by the Platform Receiver.

prrcWarning - The event was processed by the Platform Receiver, but all necessary actions were not completed.

prrcError - The event was not processed successfully by the Platform Receiver.

Action: No action is required.

EJS0034I Processed event *EventType* for Platform *PlatformName* was removed for object *DirectoryObjectFDN*.

Explanation: The event was successfully processed by the specified platform. The event is now being removed for this platform.

Action: No action is required.

EJS0035I Platform *PlatformName* added association *PlatformAssociation* for object *DirectoryObjectFDN*.

Explanation: The Platform Receiver assigned the specified association name to the directory object.

Action: No action is required.

EJS0037I Platform *PlatformName* has *NumberOfEvents* events pending.

Explanation: The platform has the specified number of pending events that are waiting to be processed by the Platform Receiver.

Action: No action is required.

EJS0038W A Platform Receiver is already active for Platform *platformName*.

Explanation: A Platform Receiver made a request to the Event Journal Services component of the Core Driver, but a Platform Receiver is already active for the specified platform. Only one Platform Receiver can be active at a time for a Platform object.

Possible Cause: Multiple Platform Receivers are attempting requests for the same Platform object.

It is also possible that a Platform Receiver has abended and left its connection token active with Event Journal Services.

Action: Run only one instance of the Platform Receiver at a time for each Platform object.

If a Platform Receiver abended, and you are trying to start a new one, allow several minutes for Event Journal Services to release control to a new instance of the Platform Receiver.

EJS0041I Searching for objects with pending events for Platform *platformName*.

Explanation: The Event Journal Services component of the Core Driver is searching for events that are pending for the specified platform.

Possible Cause: This message is in response to a get next event request from the Platform Receiver.

Action: None.

EJS0042I Pending event search for Platform *platformName* returned *numObjects* objects.

Explanation: A search for pending events for the specified platform returned the displayed number of user or group objects that have one or more pending events.

Possible Cause: This message is in response to a get next event request from the Platform Receiver.

Action: None

EJS0043I Ready to send events to Platform *platformName*.

Explanation: The Event Journal Services component of the Core Driver has finished processing the list of objects with pending events. Event Journal Services now begins sending these events to the Platform Receiver running on the specified platform.

Possible Cause: This message is in response to a get next event request from the Platform Receiver.

Action: None.

EJS0044I Removing all error events for Platform *platformName*.

Explanation: All error events for the specified platform are being removed.

Possible Cause: A Full Sync operation being performed by a Platform Receiver
An administrator is clearing the events using the Web interface.

Action: None.

EJS0045I Re-sending all error events for Platform *platformName*.

Explanation: All error events for the specified Platform are being re-sent to the Platform Receiver for retry.

Possible Cause: This action is the result of an administrator using the Web interface to specify that all error events for the platform be re-sent to the Platform Receiver.

Action: None.

EJS0046I Removing event *eventType* for object *objectCN*.

Explanation: An error event is being removed for the specified object.

Possible Cause: This can be in response to a request to remove all errors for a platform, or a request to remove the error for the individual object that is specified.

Action: None.

EJS0047I Re-sending error event *eventType* for object *objectCN*.

Explanation: An error event is being re-sent to the Platform Receiver for the specified object.

Possible Cause: This can be in response to a request to re-send all errors for a platform or a request to re-send the error for the individual object that is specified.

Action: None.

EJS0048I Platform Receiver *platformName* version is *version* build level *build*.

Explanation: The Platform Receiver is running the specified version and build level code.

Action: None.

EJS0049E Event *event* for object *objectCN* was changed to an error state.

Explanation: The event for the specified object could not be processed. The event has been set to an internal error state so that it will not be processed again until an administrator re-sends the error events for the platform.

Possible Cause: The Platform Receiver could not process the event and returned an error to the Core Driver.

The Core Driver was trying to process the event, but it could not obtain the object's password from ePassword.

Action: An Administrator can use the Web interface to re-send all error events to the platform.

EJS0050E Unable to open temporary file *fileName* for event processing (error=*errno*, reason= *reason*).

Explanation: The Event Journal Services component of the Core Driver could not open the specified temporary file that is needed for processing of queued events.

Possible Cause: The path might be invalid.

The Core Driver might not have the proper permissions to the file system.

The file system might be full.

Action: Make sure the file path exists.

Make sure the Core Driver has read/write permission to the path.

Make sure enough space exists on the volume.

EJS0051E Unable to obtain a directory connection.

Explanation: A connection could not be established to eDirectory.

Possible Cause: The replica could be down or not responding.

Action: Try the action again.

EJS0052E Unable to create temporary work files.

Explanation: The Event Journal Services component of the Core Driver could not create any temporary work files.

Possible Cause: The path might be invalid.

The Core Driver might not have the proper permissions to the file system.

The file system might be full.

Action: Make sure the file path exists.

Make sure the Core Driver has read/write permission to the path.

Make sure enough space exists on the volume.

EJS0053I Now attempting to process event *eventType* for object *objectDN*.

Explanation: The Event Journal Services component of the Core Driver is processing the event for the specified object.

Action: None.

EJS0054E Unable to add attribute *attributeName* value *attributeValue* for object *objectCN*.

Explanation: The attribute value could not be added for the specified object.

Possible Cause: There might not be enough free memory available on the system.

Action: Ensure that there is adequate free memory available.

EJS0055E Populate event was generated for object *objectCN* on platform *platformName*.

Explanation: A populate event was generated for the specified object for the single platform.

Possible Cause: The generation of the event is usually in response to a Web request to repopulate the user on the desired platforms.

Action: None.

EJS0056I Updated event timestamps for platform *PlatformName*.

Explanation: One or more attributes used for tracking event processing have been updated for the platform object.

Action: None.

EJS0057I Removing error event *eventType* for object *objectCN*.

Explanation: An error event is being removed for the specified object.

Possible Cause: This can be in response to a request to remove all errors for a platform, or a request to remove the error for the individual object that is specified.

Action: None.

EJS0058E Unable to create ASAM Directory Interface item.

Explanation: The Event Journal Services component of the Core Driver could not create a directory interface object.

Possible Cause: The LDAP server is down.

The server is low on memory.

Action: Retry the attempted operation.

EJS0059E Ignoring event for *objectDN* because of error status.

Explanation: The pending event for the specified object is ignored because an error state currently exists for the object.

Possible Cause: There are several possible causes for this error.

The platform returned an error while attempting to process the event.

If the event was for a User object, the eUser password was not available, and the platform's permit password replication setting is YES, an error state is returned for that User object.

Invalid event data.

Action: Check the platform logs to see if script errors are being reported for that object.

Check for platform errors using the Web interface, and re-send the error events to the platform.

Use the Web interface to clear error events for the platform if needed.

**EJS0060W Unable to obtain the alternate name of the Platform object
*DirectoryObjectFDN.***

Explanation: No alternate name attribute exists for the specified object.

Possible Cause: An alternate naming attribute was specified for the platform set, but no alternate name has been specified for this user/group

Action: Specify an alternate name for the object.

D.17 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 trying to connect to its desired server.

Action: None.

HES003W Core Driver has an incorrect certificate. rc = *rc*.

Explanation: The security certificate for a Core Driver could not be verified. Message HES002I precedes this message and identifies the Core Driver involved.

Possible Cause: The certificate files for the Core Driver might be missing or invalid.

Action: Obtain a new certificate for the Core Driver.

D.18 LWS Messages

Messages beginning with LWS are issued by the Core Driver as it functions as an 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 message 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 message 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 was unable to start. The server terminates processing.

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

LWS0008I Stopping all services.

Explanation: The server was requested to stop by an operator STOP command. 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 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 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 EXEC PARMs. The server terminates with a minimum return code of 8.

Action: Collect diagnostic information and contact 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 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 software support to determine the maintenance level of the server being executed.

Action: Normally, no action is required. However, if a problem with the server is called in to 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 an alternate 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 an alternate 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 address space is not active or is unavailable. The server retries every two minutes to reestablish communication with the TCP/IP address space.

Action: Ensure that the TCP/IP address space 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's 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's TCP/IP services are up and 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 was unable to 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 was unable to 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 message 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 only allows 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 through the Web interface.

Possible Cause: The ASAM Master User does not have file system rights to delete old log files.

Action: Verify that the ASAM Master User has the appropriate rights.

Examine the current logs for related messages.

LWS0048I Log file *filename* successfully deleted.

Explanation: The log file has been deleted as specified through the Web interface.

Action: None. Informational only.

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

Explanation: The connection manager was unable to connect to the directory as user *fdn*.
The error was *error*.

Possible Cause: The Driver object configuration parameters do not contain the correct password for the ASAM Master User object.

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

Verify that the ASAM Master User has the appropriate rights.

Verify that the password given for the ASAM Master User object in the configuration parameters is correct.

LWS0050E Server application initialization failure was detected.

Explanation: During server initialization, an error was detected while trying to initialize the server's application object.

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 operating system specific initialization error.

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

D.19 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.

The security certificate has expired.

The component's CERTS directory has been corrupted.

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

D.20 OAP Messages

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

OAP001E Error in SSL configuration. Check system for 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 Core Driver dn is missing from the driver XML.

Action: If you cannot resolve the error, collect diagnostic information and call Support.

OAP003E Error in SSL connect. Check address and port.

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

Possible Cause: The server is not running.

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

Action: Verify that the server is running properly.

Correct the configuration.

OAP004E HTTP Error: *cause*.

Explanation: The username/password provided for basic authentication failed.

Possible Cause: The username or password was incorrect.

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

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.

Ensure that the LDAP Host and Port Driver object configuration parameter is specified correctly.

D.21 OBJ Messages

Messages beginning with OBJ are issued by Object Services.

OBJ001I Processing Users In *search_object*.

Explanation: The Trawl is detecting all users specified by *search_object* and checking those users to determine if updates are needed in the Census.

Action: None. Informational only.

OBJ002I Checking for deleted users.

Explanation: The Trawl is looking for Enterprise Users that were not found during the processing of users specified by the Search objects. Any Enterprise Users whose corresponding User object was not found are removed from the Census.

Action: None. Informational only.

OBJ004I Processing groups in *search_object*.

Explanation: The Trawl is detecting all groups specified by *search_object* and checking those groups to determine if updates are needed in the Census.

Action: None. Informational only.

OBJ005I Checking for deleted groups.

Explanation: The Trawl is looking for Enterprise Groups that were not found during the processing of groups specified by the Search objects. Any Enterprise Group whose corresponding group object was not found is removed from the Census.

Action: None. Informational only.

OBJ007I Starting Trawl.

Explanation: A Census Trawl is starting.

Action: None. Informational only.

OBJ008I Phase *phase_number*: Processing Users.

Explanation: The Census Trawl is verifying information in the Census pertaining to users.

Action: None. Informational only.

OBJ009I Phase *phase_number*: Processing Groups.

Explanation: The Census Trawl is verifying information in the Census pertaining to groups.

Action: None. Informational only.

OBJ010I Trawl complete.

Explanation: A Census Trawl is ending.

Action: None. Informational only.

OBJ013W No valid Search objects found for *Census_or_Platform_set*.

Explanation: *Census_or_Platform_set* has no Search objects defined.

Possible Cause: Configuration of the product might not be complete.

Action: Define Search objects for the identified component.

OBJ014W No Platforms found in *Platform_set*.

Explanation: Platform Set *Platform_set* has no platforms defined for it.

Possible Cause: Configuration of the Platform Set might not be completed.

Action: Add desired platforms to the Platform Set.

OBJ015I No UID/GID Sets found.

Explanation: No UID/GID Sets were found.

Possible Cause: No UID/GID Set has been created.

Action: If Linux/UNIX Platforms are to be controlled, define needed UID/GID Sets.

OBJ016W Search object *search_object_name* does not have a value for *attribute_name*. It is ignored.

Explanation: A Search object must have a value for *attribute_name* in order to be processed. *search_object_name* does not have this value.

Possible Cause: The Search object might have been edited manually.

Action: Determine the intended values for the Search object and set the values.

OBJ017E UID/GID Set *UID_GID_set_name*, specified for Platform Set *Platform_set_name*, was not found.

Explanation: The UID/GID Set named *UID_GID_set_name* could not be found. It is referenced by Platform Set *Platform_set_name*. Identity Provisioning cannot function properly on any Linux/UNIX platforms defined for the Platform Set named *Platform_set_name*.

Possible Cause: The UID/GID Set container named *UID_GID_set_name* was manually removed from eDirectory.

Action: Restore the UID/GID container named *UID_GID_set_name* from backup.

OBJ018W No Platform Sets found.

Explanation: No Platform Sets were found. Account information cannot be exported to any platforms.

Possible Cause: Configuration of the product might not have been completed.

Action: Define Platform Sets as needed for your installation.

OBJ019I UID/GID *number* assigned to user in UID/GID Set *uidgid_set_name*.

Explanation: UID/GID number *number* has been assigned to user *user* in UID/GID Set *uidgid_set_name*. This is the ID that is used for Linux/UNIX platforms in Platform Sets that use UID/GID Set *uidgid_set_name*.

Action: None. Informational only.

OBJ020I Exception resolved for *exception_object*.

Explanation: The condition that caused the creation of Exception object *exception_object* has been corrected. The Exception object has been removed.

Action: None. Informational only.

OBJ021I Added *user_or_group_name* to Platform Set *Platform_set_name*.

Explanation: A user or group named *user_or_group_name* has been added to the Platform Set specified by *Platform_set_name*.

Action: None. Informational only.

OBJ022I Enterprise object *object_name* removed from Census.

Explanation: The Enterprise object named *object_name* was removed from the Census.

Possible Cause: The user, group, or alias represented by the Enterprise object named *object_name* was deleted from the directory, is disabled, or is no longer included by the Search objects.

Action: None. Informational only.

OBJ023I Enterprise object *object_name* renamed to *new_object_name*.

Explanation: The Enterprise object named *object_name* was renamed to *new_object_name*.

Possible Cause: The user, group, or alias represented by *object_name* was renamed to *new_object_name*.

Action: None. Informational only.

OBJ024I Created Exception object for *object_dn*.

Explanation: A group or user could not be processed.

Possible Cause: The cn of the Group or User object is not unique among all the users and groups that are represented in the Census.

Two or more objects in the directory have the same GUID.

Action: Examine the contents of the Exception object to determine the reason it was created.

If the Exception object is because of a create problem, a naming conflict has occurred. Rename the user or group so its name is unique.

If the Exception object is because of a duplicate GUID, look in the operational log for a listing of the objects that use the same GUID, and see TID 10064771 for information on resolving GUID conflicts.

OBJ025I User *user_name*, attribute(s) *attribute_list* modified in Census.

Explanation: Information for user *user_name* was updated in the Census.

Action: None. Informational only.

OBJ026I Group *group_name*, attribute(s) *attribute_list* modified in Census.

Explanation: Information for group *group_name* was updated in the Census.

Action: None. Informational only.

OBJ027I User *user_name* added to Census.

Explanation: User *user_name* was detected and added to the Census.

Possible Cause: A user was added to eDirectory, or Search objects were expanded to include a user that was not previously in the Census.

Action: None. Informational only.

OBJ028I Group *group_name* added to Census.

Explanation: Group *group_name* was detected and added to the Census.

Possible Cause: A group was added to eDirectory, or Search objects were expanded to include a group that was not previously in the Census.

Action: None. Informational only.

OBJ030E Error *error_id* authenticating to eDirectory as *username*.

Explanation: The Core Driver is unable to authenticate to eDirectory.

Possible Cause: Incorrect settings for LDAP Host and Port, ASAM Master User, or ASAM Master User Password in the Driver object configuration parameters.

Action: Check the configuration parameters.

OBJ031E Error *error_id* renaming object *dn* to *cn*.

Explanation: The eDirectory error *error_id* occurred while trying to rename object *dn* to *cn*.

Action: See the eDirectory documentation for error *error_id*.

OBJ032E Out of memory.

Explanation: The Core Driver ran out of memory.

Possible Cause: The machine on which the Core Driver runs does not have enough memory to allow operation, or the swap space is not large enough.

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

OBJ033E Error *error_id* retrieving from *dn*.

Explanation: The eDirectory error *error_id* occurred while trying to retrieve from *dn*.

Action: See the eDirectory documentation for error *error_id*.

OBJ034E Error *error_id* retrieving attributes for *object*.

Explanation: The eDirectory error *error_id* occurred while retrieving attributes for object *object*.

Action: See the eDirectory documentation for error *error_id*.

OBJ035E Error *error_id* modifying attributes for *object*.

Explanation: The eDirectory error *error_id* occurred while trying to modify *object*.

Possible Cause: Insufficient rights to the object.

Action: See the eDirectory documentation for error *error_id*.

OBJ036E Error *error_id* searching for object *object*.

Explanation: The eDirectory error *error_id* occurred while trying to determine if *object* exists.

Action: See the eDirectory documentation for error *error_id*.

OBJ037E Error *error_id* creating object *object*.

Explanation: The eDirectory error *error_id* occurred while trying to create *object*.

Possible Cause: Incorrect ASAM System Container setting in the Driver object configuration, or insufficient rights to this container.

Action: See the eDirectory documentation for error *error_id*.

OBJ038E Error *error_id* removing object *object*.

Explanation: The eDirectory error *error_id* occurred while trying to remove *object*.

Action: See the eDirectory documentation for error *error_id*.

OBJ039E Unexpected error processing information retrieved from the directory in function *function_name*.

Explanation: An unexpected error has occurred during processing.

Possible Cause: Unknown.

Action: Turn on debugging information using the command line parameter -d *asam_objectserv,dom*, and forward the resulting log to Support.

OBJ040E Unable to load request document.

Explanation: An eDirectory event could not be processed.

Possible Cause: Internal error.

Action: Turn on debugging information using the command line parameter -d *asam_objectserv,dom*, and forward the resulting log to Support.

OBJ041E Unable to determine DN for the ASAM System Container.

Explanation: The ASAM System container cannot be identified.

Possible Cause: The Driver object configuration parameters do not contain a valid value for the ASAM System Container parameter.

Action: Correct the ASAM System Container parameter.

OBJ042E Unable to process some users in *search_object*.

Explanation: Appropriate actions for some of the users in *search_object* might not have been taken because of errors that occurred.

Action: See other errors reported during the processing of *search_object* for specific troubleshooting information.

OBJ043E Unable to process some groups in *search_object*.

Explanation: Appropriate actions for some of the groups in *search_object* might not have been taken because of errors that occurred.

Action: See other errors reported during the processing of *search_object* for specific troubleshooting information.

OBJ044E Unable to process some aliases in *search_object*.

Explanation: The Core Driver was unable to process an Alias object.

Action: See the log for more information about the specific error.

OBJ046I Updated attribute *attribute_name* in object *object*.

Explanation: An out-of-date attribute of an Enterprise User or Group object was detected. The attribute was updated.

Possible Cause: A Core Driver might not be running or might not be functioning properly.

A new user was added to the Census, and a group to which it belongs was updated accordingly.

A new group was added to the Census, and a user in that group was updated accordingly.

Action: Ensure proper operation of all Core Drivers.

OBJ047I Removed *object_cn* from Platform Set *Platform_set*.

Explanation: *object_cn* was removed from Platform Set *Platform_set*.

Possible Cause: The user or group is no longer included in the Search objects defined for the Platform Set.

Action: None. Informational only.

OBJ051E Duplicate GUID found among the listed objects: *dn_list*.

Explanation: Multiple objects exist in the tree with the same GUID.

A list of the objects having duplicate GUIDs is produced in the log.

Action: As described in TID 10064771, duplicate GUIDs can only be fixed by deleting all but one of the objects and re-creating them. An eDirectory patch is available to prevent multiple GUIDs from being generated in the future. For a complete explanation, see TID 10064771.

OBJ052E Duplicate ASAM-inputGUID found among the listed objects: *dn_list*.

Explanation: Multiple objects exist in the tree with the same GUID.

A list of the objects having duplicate GUIDs is produced in the log.

Action: As described in TID 10064771, duplicate GUIDs can only be fixed by deleting all but one of the objects and re-creating them. An eDirectory patch is available to prevent multiple GUIDs from being generated in the future. For a complete explanation, see TID 10064771.

OBJ053I Created events of type *event_type* for *object*.

Explanation: A change in the User or Group object was detected. Affected platforms are notified.

Action: None. Informational only.

OBJ055E UID/GID Set *set_name* was not found.

Explanation: When assigning a UID/GID for an eUser or eGroup, the requested UID/GID Set could not be found.

Possible Cause: A UID/GID Set container was manually removed from eDirectory.

Action: Restore the UID/GID container from backup.

OBJ056E Unable to retrieve object *object_dn* referenced by alias *alias_dn*.

Explanation: The object referenced by an alias could not be found.

Possible Cause: An Alias object refers to a user or group to which the ASAM Master User has insufficient rights.

Action: Grant necessary rights to the ASAM Master User.

OBJ057E Unable to retrieve attribute *attribute_name* from *object_dn*.

Explanation: An attribute needed for processing could not be retrieved.

Possible Cause: The ASAM Master User does not have sufficient rights.

Action: Ensure that the ASAM Master User has the necessary rights.

OBJ058E Duplicate UID/GID number *uidgid_number* found in both *object1* and *object2*.

Explanation: Duplicate UID/GID numbers have been discovered. A UID/GID number is used on Linux/UNIX systems to uniquely identify an account or a group. Duplicate UID/GID numbers can indicate that an unintended user has access to Linux/UNIX resources, such as files.

Possible Cause: Partial restoration of the ASAM System container could result in duplicate UID/GID numbers.

Action: Determine which user or group should correspond to the associated UID/GID. Manually remove the ASAM-uidgidAssociation value for any other users or groups that are assigned that same number. A new UID/GID will be assigned during the next Trawl for those that have been deleted.

OBJ059E Cannot remove Platform Set *Platform_set_name*. It has associated Platform objects.

Explanation: A Platform Set has been marked for removal, but it cannot be removed. All platforms must be removed from it first.

Possible Cause: Platforms were added to a Platform Set that had been marked for removal.

Action: Remove all platforms from the Platform Set.

OBJ060I Removed Platform Set *Platform_set*.

Explanation: The Platform Set named *Platform_set* was removed.

Possible Cause: The Platform Set was marked for deletion using the Web interface.

Action: None. Informational only.

OBJ061E Cannot remove UID/GID Set *uidgid_set_name*. It is used by a Platform Set.

Explanation: A UID/GID Set has been marked for removal, but it cannot be removed. All Platform Sets using the UID/GID Set must be removed first.

Action: Remove all Platform Sets that use the UID/GID Set.

OBJ062I Removed UID/GID Set *uidgid_set*.

Explanation: The UID/GID Set named *uidgid_set* was removed.

Possible Cause: The UID/GID Set was marked for deletion using the Web interface.

Action: None. Informational only.

OBJ064W Error *error_id* setting LDAP time-out.

Explanation: An error occurred while trying to use the LDAP Time-Out value.

Action: See the eDirectory documentation for error *error_id*.

OBJ065E Platform Set *set_name* not found in directory.

Explanation: An error occurred while looking up information about the Platform Set named *set_name*.

Action: Gather diagnostic information and contact Support.

OBJ066E Unable to recognize object type of Search object *search_object_name*.

Explanation: The Search object has as its input reference the dn of an unsupported object type.

Possible Cause: An incorrect object is specified as the input reference for a Search object.

Action: Remove the invalid Search object and recreate it using the correct input reference.

OBJ069E Skipping checks for deleted users because of errors during processing of users.

Explanation: Deleted users are detected during a Trawl when processing of all users has completed. If an error prevents the recognition of all users that should be in the Census, then no users are deleted.

Possible Cause: Time-outs prevented the detection of all users defined by the Search objects, or a Search object was invalid.

Action: Check the operational log for errors and determine the actions required to resolve those errors.

OBJ070E Skipping checks for deleted groups because of errors during processing of groups.

Explanation: Deleted groups are detected during a Trawl when processing of all groups has completed. If an error prevents the recognition of all groups that should be in the Census, then no groups are deleted.

Possible Cause: Time-outs prevented the detection of all groups defined by the Search objects, or a Search object was invalid.

Action: Check the operational log for errors and determine the actions required to resolve those errors.

OBJ072E Unrecognized object class for object *dn* in *function_name*.

Explanation: The Core Driver was unable to determine the object class for *dn*.

Possible Cause: The object denoted by *dn* is an object whose class is not supported.

Action: Ensure that *dn* exists and is spelled correctly. Inspect the object denoted by *dn* to determine whether its object class is supported. If so, contact Support. If not, you cannot manage this object.

OBJ073E Cannot handle object class *internal_objectclass_identifier* for object *dn* in *function_name*.

Explanation: The Core Driver was unable to process the object class denoted by *internal_objectclass_identifier* for the object given by *dn*. The problem occurred in the function named *function_name*.

Possible Cause: The object denoted by *internal_objectclass_identifier* has an object class that is not supported for the attempted purpose.

Action: Ensure that *internal_objectclass_identifier* exists and is spelled correctly. Inspect the object denoted by *internal_objectclass_identifier* to determine whether its object class is supported. If so, contact Support. If not, you cannot manage this object.

OBJ074E Cannot determine Platform Set for *dn*.

Explanation: The *dn* of the Platform object *dn* could not be parsed to determine the Platform Set name.

Possible Cause: Internal error.

Action: Gather diagnostic information and contact Support.

OBJ075I Trawl aborted because of user request.

Explanation: The Trawl was aborted because of a user request for it to stop.

Possible Cause: An administrator used the Web interface to stop the Trawl.

The Core Driver was shut down.

Action: None. Informational only.

OBJ076I Deleting Platform Set *set_name*.

Explanation: The container for Platform Set *set_name* and all references to it are being removed. This operation can take some time, depending on the number of users and groups that are managed.

Possible Cause: The Platform Set *set_name* was marked for deletion using the Web interface.

Action: None.

OBJ077I Deleting UID/GID Set *set_name*.

Explanation: The container for UID/GID Set *set_name* and all references to it are being removed. This operation can take some time, depending on the number of users and groups that are managed.

Possible Cause: The UID/GID Set *set_name* was marked for deletion using the Web interface.

Action: None.

OBJ079E Unable to convert dn *dn* to required format.

Explanation: The dn *dn* could not be converted to the format required for processing.

Possible Cause: No memory was available.

Action: Ensure that the process has enough memory to complete.

OBJ080E Unable to create file *file_name*. Error = *errno*.

Explanation: An attempt to create the file *file_name* failed.

Possible Cause: The directory is write-protected, or there is not enough disk space available.

Action: Ensure that the ASAM Master User has permission to write to the specified directory. Ensure that disk space is available on the volume.

OBJ081E Unable to write to file *file_name*. Error = *errno*.

Explanation: An attempt to write to the file *file_name* failed.

Possible Cause: There is not enough disk space available.

Action: Ensure that disk space is available on the volume.

OBJ082E Unable to delete file *file_name*. Error = *errno*.

Explanation: An attempt to delete the file *file_name* failed.

Possible Cause: Permissions do not allow the file to be deleted.

Action: Ensure that the ASAM Master User has permission to delete the specified directory.

OBJ084I Checking UID/GID Set *UIDGID_set*.

Explanation: The Census Trawl is verifying the contents of UID/GID Set *UIDGID_set*.

Action: None. Informational Only.

OBJ086W Unable to start Trawl because a Trawl is already running.

Explanation: A Trawl could not start because a Trawl is already in progress.

Possible Cause: The specified scheduled Trawl times are not sufficiently spaced to allow completion of the previous Trawl.

A manual Trawl was started and it had not completed before the scheduled Trawl time arrived.

Action: Wait until the currently running Trawl has completed, or stop the Trawl and restart it manually.

OBJ087E Cleanup of resources from the previous Trawl failed.

Explanation: An error occurred while trying to free resources used by the previously run Trawl.

Action: Wait for the Trawl to complete. Use the Trawl Status screen in the Web interface to confirm that no Trawl is running. If you are still unable to start a Trawl, restart the primary Core Driver.

OBJ088E Unable to allocate resources for starting a Trawl.

Explanation: A task could not be created for performing a Trawl.

Possible Cause: The system is low on memory.

Action: Restart the primary Core Driver. If the problem persists, look for other processes that are consuming excessive memory.

OBJ089E Unable to start the Trawl task.

Explanation: A task could not be started because of system limitations. The implementation of a task is operating system dependent. For example, a task might be implemented as a thread. In this case, a thread could not be created.

Possible Cause: The system is low on resources.

Action: Determine and correct the cause of limited system resources.

OBJ090E Unable to read from file *file_name*. Error = *errno*.

Explanation: An attempt to read from the file *file_name* failed.

Possible Cause: Internal error.

Action: Turn on debugging information using the command line parameter: -d asam_objectserv,dom, and forward the resulting log to Support.

OBJ091W Object type of *object_dn* is not recognized.

Explanation: The object class for the object was not recognized.

Possible Cause: The given object does not have an object class that can be processed.

Action: Examine the object named by *object_dn* to determine why it cannot be processed.

OBJ092E Unable to determine value of attribute *attribute_name* for object *object_name*.

Explanation: An attempt to read the value for attribute *attribute_name* failed.

Possible Cause: System memory is low.

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

OBJ093E Unable to create directory search request.

Explanation: An attempt to read information from the directory failed.

Possible Cause: System memory is low.

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

OBJ094E Unable to create request to modify attributes in the directory.

Explanation: An attempt to modify information in the directory failed.

Possible Cause: System memory is low.

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

OBJ095E Unable to initialize mutex.

Explanation: A mutex could not be initialized.

Possible Cause: The system is low on available resources.

Action: Ensure adequate resources for the process.

OBJ096E Unable to find object *dn* during repair of links in Census because of error *error_id*.

Explanation: When attempting to repair Census information for the previously deleted object *dn*, the reinstated object could not be found.

Possible Cause: The object has not yet been re-created.

Action: Re-create or restore the object *dn*.

OBJ097I ASAM-inputGUID updated in object *dn*.

Explanation: Information has been repaired in object *dn*.

Possible Cause: Census information is being repaired for the user.

Action: None.

OBJ098I Processed *processed_count* of users *in_search_object* users.

Explanation: Indicates progress in processing the users specified by a Search object.

Action: None.

OBJ099I Processed *processed_count* of groups *in_search_object* groups.

Explanation: Indicates progress in processing the groups specified by a Search object.

Action: None.

OBJ100I Processed *processed_count* of *aliases_in_search_object* aliases.

Explanation: Indicates progress in processing the aliases specified by a Search object.

Action: None.

OBJ102I Processed *processed_count* UIDGID objects.

Explanation: Indicates progress in processing the UID/GID objects in a UID/GID Set.

Action: None.

OBJ105I Dispatching new event notification to Platform *platformName*.

Explanation: Object Services is dispatching a notification to Event Journal Services that a new event is ready to be processed for the specified platform.

Only Platform Receivers that are running in Persistent mode are notified of new events that are pending. Platform Receivers running in other modes discover the new events the next time they poll or connect to Event Journal Services.

Possible Cause: A new object event has been detected by the Event Subsystem or a Trawl process.

Action: The Event Journal Services component processes the event and sends it to the Persistent mode Receiver that is running on the specified platform.

OBJ106I Phase *phase_number*: Processing Password Updates.

Explanation: The Census Trawl is updating ePasswords that Core Drivers were previously unable to store.

Action: None. Informational only.

OBJ107E Attempt to process an event with no DN was aborted.

Explanation: An event was detected for an eDirectory object, but the dn of that object was unavailable. The event could not be processed.

Possible Cause: Running a down-level version of the Core Driver.

Action: Update the Core Driver.

OBJ108I Updated password for user *object_dn*.

Explanation: The password stored for object *object_dn* was updated.

Possible Cause: The password for the object has changed.

Action: None.

OBJ109E Error *error_id* updating password for user *object_dn*.

Explanation: The password for *object_dn* could not be updated because of error *error_id*.

Action: Change the password for the given user.

OBJ111I Removed password from temporary storage for user *user_dn*.

Explanation: A password that was held in temporary storage pending processing by the Core Driver was removed.

Possible Cause: The password was successfully stored, or the user is not managed.

Action: None.

OBJ112I Error *error_id* removing password for user *user_dn* from temporary storage.

Explanation: A password that was held in temporary storage pending processing by the Core Driver could not be removed.

Action: None.

OBJ113I *user_or_group_name* updated for driver storage format.

Explanation: The user or group has been updated for use with the driver. It will no longer function correctly with Account Management 3.0.

Action: None.

OBJ114I Removed *object_cn* from UID/GID Set *UIDGID_set*.

Explanation: *object_cn* was removed from UID/GID Set *UIDGID_set*.

Possible Cause: The UID or GID number has been migrated to a new storage format.

Action: None. Informational only.

OBJ115I Migrating *user_or_group_name* to driver storage format.

Explanation: Data for the user or group is being converted to the storage format used by the driver.

Possible Cause: Software version has been updated.

Action: None. Informational only.

OBJ116I Updating inclusion in Platform Set *platform_set* for *user_or_group*.

Explanation: Platform Set information for the user or group is being migrated to a new storage format.

Action: None. Informational only.

OBJ117I Updating association to platform *platform* for *user_or_group*.

Explanation: Platform Association information for the user or group is being migrated to a new storage format.

Action: None. Informational only.

OBJ118I Updating UID/GID in set *uidgid_set* for *user_or_group*.

Explanation: UID/GID information for the user or group is being migrated to a new storage format.

Action: None. Informational only.

OBJ119I Removed object *object_dn*.

Explanation: Object *object_dn* was removed during data migration to a new storage format.

Action: None. Informational only.

OBJ120I Object Services received an event for *object_dn*.

Explanation: The Event Subsystem notified Object Services of an event.

Possible Cause: An object was added, changed, or deleted in eDirectory.

Action: None. Informational only.

OBJ121I Object Services received an event for object with unidentified dn.

Explanation: The Event Subsystem notified Object Services of an event.

Possible Cause: An object was added, changed, or deleted in eDirectory.

Action: None. Informational only.

OBJ122I Processing a pseudo-event for *object_dn*.

Explanation: The object is being processed as if an event occurred.

Possible Cause: The object was re-populated.

Action: None. Informational only.

OBJ123E Delete action for *object_cn* aborted because of invalid Search object.

Explanation: One or more Search objects did not contain a valid inputReference.

Possible Cause: A Search object exists for which the object specified by the inputReference has been deleted, or an error occurred while trying to retrieve information from the object specified by the inputReference.

Action: Determine which Search object is not valid and correct it.

OBJ124I Obsolete object *dn* successfully removed.

Explanation: The information represented by object *dn* has been updated to a new storage format. The obsolete object has been cleaned up.

After removal of a large number of objects, it can be desirable to use directory maintenance techniques to reduce the size of the directory on disk.

Possible Cause: A new version of the Fan-Out Driver software has been installed.

Action: None.

OBJ125I Migration status changed to *migration_status*.

Explanation: Stages of data conversion are Migration (to new data format), Cleanup (removal of obsolete objects), and Complete.

Each user or group is migrated to the new data format the first time it is processed by the Core Driver.

After all users and groups have migrated to the new data format, cleanup of obsolete objects begins.

The status is reported as Complete after all users or groups have been migrated, and all obsolete objects have been cleaned up. The size of the eDirectory database can be reduced by using standard eDirectory maintenance practices when this stage has been reached.

Possible Cause: A new version of the Fan-Out Driver software has been installed.

Action: None.

OBJ126I Phase *phase_number*: Migration Cleanup.

Explanation: The Census Trawl is removing obsolete data that has been migrated to a new storage format.

Action: None. Informational only.

OBJ127E Alternate name attribute *alternate_name* must have single value or form *<platform set name>:<alt name>*.

Explanation: Attributes used for specifying alternate names must have only a single value, or all values must be of the form *<platform set name>:<alternate name>*.

Action: Modify the alternate naming attribute to either have one value, or have values of the form *<platform set name>:<alternate name>*.

OBJ128I Created Census entry for alternate name *alternate_name*.

Explanation: An entry was created in the Census to represent the alternate name for the object.

Action: None.

OBJ129E Could not add alternate name *alternate_name* to the Census. Name already exists.

Explanation: Another object already exists in the Census with the specified name.

Possible Cause: Another user or group has the same name or alternate name.

Action: Resolve as you would any naming exception.

OBJ130I Removed alternate name *alternate_name* from Census.

Explanation: An alternate name was removed from the Census.

Action: None.

OBJ131I Name is on the census exclude list.

Explanation: The user or group has been designated as one to exclude from the Census.

Possible Cause: The user or group has been manually added to the Census exclude list.

Action: The user or group may be removed from the Census exclude list from the Provisioning Configuration screen.

OBJ132I Removed obsolete Platform Association association from *object_cn*.

Explanation: The obsolete Platform Association association was removed from Census object *object_cn*.

Possible Cause: The obsolete platform was removed from the Fan-Out Configuration.

Action: None. Informational only.

D.22 PLS Messages

Messages beginning with PLS are issued by Platform Services.

PLS001I *core_driver* is not responding correctly. rc = rc.

Explanation: The specified Core Driver is not answering requests correctly. Requests are not directed to this Core Driver again until it begins responding correctly.

Action: None.

PLS002I *core_driver* is now responding to requests.

Explanation: The specified Core Driver has returned to a usable state.

Action: None.

D.23 PRCV Messages

Messages beginning with PRCV are issued by Platform Receivers.

PRCV001E Unable to create the platform parameter item.

Explanation: The Platform Receiver was unable to create a platform parms item, which is used for parsing the Platform Configuration file.

Possible Cause: There might not be enough free memory available on the system.

Action: Ensure that there is adequate free memory available.

PRCV002E Unable to create a string handler item.

Explanation: An instance of the string handler object could not be created.

Possible Cause: There might not be enough free memory available on the system.

Action: Ensure that there is adequate free memory available.

PRCV003E Unknown command line option or error: option= *ShortOptionValue*, long option= *LongOptionValue*.

Explanation: An unknown command line option was discovered while processing the command line options.

Possible Cause: You entered an invalid command line option.

Action: See the administration documentation for the list of valid command line options.

PRCV004E Mutually exclusive command line parameters were specified.

Explanation: One or more command line options are mutually exclusive.

Possible Cause: You entered conflicting Platform Receiver run modes on the command line.

Action: Determine the desired Platform Receiver run mode and enter the corresponding option on the command line.

PRCV005I You can specify only one of the following options: -i, -c, -p, -f, or -r.

Explanation: This message describes the valid run modes that are available for the Platform Receiver.

Action: No action is required.

PRCV006E Platform Configuration file parsing has failed because of a syntax error.

Explanation: The parsing of the Platform Configuration file has ended with a syntax error.

Possible Cause: A syntax error exists in the Platform Configuration file.

Action: Ensure that the parameters in the Platform Configuration file are valid.

PRCV007E Unable to create a configuration parameter item.

Explanation: An instance of the configuration parameter item could not be created.

Possible Cause: There might not be enough free memory available on the system.

Action: Ensure that there is adequate free memory available.

PRCV008E Unable to load the string resource file *StringResourceFileName*.

Explanation: The specified string resource file could not be loaded.

Possible Cause: The Platform Receiver attempted to load the string resource file in response to the LOCALE statement in the Platform Configuration file.

Action: Ensure that the specified string resource file exists.

Ensure that the ASAMDIR statement is correct.

PRCV009E Unable to establish a connection with host *ipAddress* port *portNumber*.

Explanation: The Platform Receiver is unable to make a socket connection to the Core Driver.

Possible Cause: Network connectivity between the Platform Receiver and the Core Driver server is lost.

The host that runs the Core Driver is down.

Event Journal Services has failed.

The Core Driver is running on a different port than is expected by the Platform Receiver.

Action: Ensure that network connectivity exists between the Platform Receiver and the Core Driver server.

Ensure that the Core Driver host is up.

Ensure that the Core Driver is running.

Ensure that the Core Driver is listening on the port number expected by the Platform Receiver.

PRCV010I Connection established with host *ipAddress* port *portNumber*.

Explanation: Socket connectivity to the Core Driver has been reestablished.

Possible Cause: The connection to the Core Driver that was previously interrupted has been reestablished.

Action: No action is required.

PRCV011E Unable to begin a session with host *ipAddress* port *portNumber*, reason=*reasonString*.

Explanation: The Platform Receiver was unable to establish a session with the Core Driver.

Possible Cause: There are several possible causes for this error.

The Core Driver has terminated the connection.

The request was rejected by the Core Driver because of an invalid certificate or internal server error.

An instance of the DOM interface could not be created.

An instance of the SOAP request document could not be created.

Action: See the reason string for additional details on the cause of the error. Also verify the following items:

Ensure that the Core Driver is running.

Ensure that the platform host has network connectivity to the host running the Core Driver.

Ensure that the correct security certificate is installed on the system.

PRCV012W *MessageFromManager*.

Explanation: This error message is generated by the Event Journal Services component of the Core Driver and is reported by the Platform Receiver.

Possible Cause: The Event Journal Services component of the Core Driver discovered an error condition. The Platform Receiver is reporting the error to the local host system.

Action: Take action as appropriate for the message text.

PRCV013E Unable to complete the get next platform event request.

Explanation: The Platform Receiver is unable to get events from the Core Driver.

Possible Cause: Connectivity to the Core Driver has been interrupted and the Platform Receiver has exceeded the retry attempt limit for reestablishing the connection to the Core Driver.

Action: Determine why the connection to the Core Driver was interrupted.

PRCV014I The driver running on host *ipaddress* on port *port* is shutting down.

Explanation: The Core Driver running on the specified network address and port number is shutting down. If the Platform Receiver is running in Persistent Mode or Polling Mode, the Platform Receiver tries to reestablish a connection to the Core Driver.

Action: No action is required.

PRCV015E The security certificate could not be loaded.

Explanation: The security certificate was not loaded or is not valid.

Possible Cause: There are several possible causes for this error.

No security certificate has been created for this platform.

The security certificate is invalid.

The security certificate could not be found, possibly because of an incorrect ASAMDIR statement in the Platform Configuration file.

Action: Ensure that the security certificate was created and installed on the platform. A security certificate can be obtained by running the Platform Receiver with the `-s` command line parameter.

Ensure that the ASAMDIR statement is correct.

PRCV016I The Platform Receiver is shutting down because of a stop request.

Explanation: An administrator has requested that the Platform Receiver stop processing and end.

Action: The Platform Receiver ends as soon as it completes any required tasks.

PRCV017I SSL Certificate Local FDN is *SSLLocal/FDN*.

Explanation: The message logs the Platform Receiver FDN found in its security certificate.

Action: If this message is not issued, the certificate is either missing or corrupt. Obtain a new security certificate for the Platform Receiver by starting it with the `-s` command line parameter.

If this message is issued, the FDN should be verified to be the correct object in eDirectory.

PRCV018I The Platform Receiver for *platformName* is running in *runMode* mode.

Explanation: The Platform Receiver is running in the specified mode.

Possible Cause: The Platform Receiver is running in the mode specified by the RUNMODE configuration statement value or the value of command line parameters.

Action: None.

PRCV019I An event was received for object *objectCN*.

Explanation: An event was received from the Core Driver for the specified user or group object.

Action: No action is required.

PRCV020I The event for object *objectCN* was excluded.

Explanation: The event for the specified object was excluded because of the use of an AM.USER.EXCLUDE or AM.GROUP.EXCLUDE statement in the Platform Configuration file, or the object is on the standard excludes list.

Action: No action is required.

PRCV021I Connection established with host *ipAddress* port *portNumber* version *version* build level *build*.

Explanation: The Platform Receiver is communicating with a Core Driver running the specified version and build level code.

Action: No action is required.

PRCV022I Platform Receiver version is *version* build level *build*.

Explanation: The Platform Receiver is running the specified version and build level code.

Action: No action is required.

PRCV023I Event summary for Platform *platformName*: received= *numReceived*, processed= *numProcessed*, excluded= *numExcluded*, ignored= *numIgnored*, errors= *numErrors*.

Explanation: This message displays the total number of events that were received from the Core Driver, the number of events that were processed successfully, the number of events that were excluded by the platform, and the number of events that were not processed because of errors.

This message is displayed when the Platform Receiver terminates.

Action: None.

PRCV024I *objectType* event totals for Platform *platformName*: received= *numReceived*, processed= *numProcessed*, excluded= *numExcluded*, ignored= *numIgnored*, errors= *numErrors*.

Explanation: This message displays the total number of events for the specified object type that were received from the Core Driver, the number of events that were processed successfully, the number of events that were excluded by the platform, and the number of events that were not processed because of errors.

This message is displayed when the Platform Receiver terminates.

Action: None.

PRCV025I Platform Receiver executed for *days* days, *hours* hours, *minutes* minutes, and *seconds* seconds.

Explanation: This message displays the execution time for the Platform Receiver.

This message is displayed when the Platform Receiver terminates.

Action: None.

D.24 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.

D.25 W3LM Messages

Messages beginning with W3LM are issued by Web Services.

W3LM001I Object *driverDN* created by *webUserDN*.

Explanation: A Core Driver was created by the specified user through the Web interface.

Action: None. Informational only.

W3LM002I Object *driverDN* deleted by *webUserDN*.

Explanation: A Core Driver was deleted by the specified user through the Web interface.

Action: None. Informational only.

W3LM003I Event Listener *eventListenerDN* deleted by *webUserDN*.

Explanation: The Event Listener was deleted by the specified user through the Web interface.

Action: None. Informational only.

W3LM004I Trawl Initiated by *webUserDN*.

Explanation: A Trawl was started by the specified user through the Web interface.

Action: None. Informational only.

W3LM007I Platform *platformDN* deleted by *webUserDN*.

Explanation: A Platform object was deleted by the specified user through the Web interface.

Action: None. Informational only.

W3LM008I Platform *platformDN* created by *webUserDN*.

Explanation: A Platform object was created by the specified user through the Web interface.

Action: None. Informational only.

W3LM009I Platform Set *platformSetDN* marked for deletion by *webUserDN*.

Explanation: The specified Platform Set was marked for deletion by the specified user through the Web interface.

Action: None. Informational only.

W3LM010I Platform Set *platformSetDN* created by *webUserDN*.

Explanation: The specified Platform Set was created by the specified user through the Web interface.

Action: None. Informational only.

W3LM011I UID/GID Set *UIDGIDSetDN* marked for deletion by *webUserDN*.

Explanation: The UID/GID Set was deleted by the specified user through the Web interface.

Action: None. Informational only.

W3LM012I UID/GID Set *UIDGIDSetDN* created by *webUserDN*.

Explanation: The specified UID/GID set was created by the specified user through the Web interface.

Action: None. Informational only.

W3LM013I SearchObject *searchObjectDN* created by *webUserDN*.

Explanation: The Search object was created by the specified user through the Web interface.

Action: None. Informational only.

W3LM014I SearchObject *searchObjectDN* deleted by *webUserDN*.

Explanation: The specified Search object was deleted by the specified user through the Web interface.

Action: None. Informational only.

W3LM015I Object *objectDN* modified by *webUserDN*.

Explanation: The specified object was modified by the specified user through the Web interface.

Action: None. Informational only.

W3LM016I Connection (default) *netAddress* attribute on object *objectDN* modified by *webUserDN*.

Explanation: Connection (default) *netAddress* attribute on the specified object was modified by the specified user through the Web interface.

Action: None. Informational only.

W3LM017I *netAddress* attribute on object *objectDN* modified by *webUserDN*.

Explanation: The *netAddress* attribute of the specified object was modified by the specified user through the Web interface.

Action: None. Informational only.

W3LM018W Web Interface login Failure *loginDN*.

Explanation: An attempt to authenticate to the Web interface by *loginDN* failed.

Possible Cause: Invalid login ID, password, or insufficient rights.

Action: Log in with sufficient rights.

W3LM019I Successful Web Interface login by *loginID*.

Explanation: The user successfully logged in to the Web interface.

Action: None. Informational only.

W3LM020W Web Interface login attempt with invalid credentials.

Explanation: An attempt to log in to the Web interface failed because of invalid credentials.

Possible Cause: The user attempting to log in has invalid credentials

Action: Check user credentials.

W3LM021W Web Interface login attempt with invalid DN Syntax.

Explanation: An attempt to log in to the Web interface was made with invalid DN syntax.

Possible Cause: DN syntax was invalid.

Action: Correct DN syntax and try logging in again.

W3LM022W Web Interface login attempt for an unknown user.

Explanation: The user attempting to log in to the Web Interface is invalid because a Census entry for the user was not found.

Possible Cause: The user is not in Census.

Action: Make sure the user is in the Census.

W3LM023W Web Interface login attempt failure with an unknown error.

Explanation: An attempt to log in to the Web interface failed with an unknown error.

Action: Examine the log for related messages.

W3LM024E Check the Trawl Time-Out value and re-enter.

Explanation: The Trawl Time-Out value is invalid.

Possible Cause: An invalid Trawl Time-Out value was specified.

Action: Correct the Trawl Time-Out value.

Glossary

Account Redirection

The process of ensuring that users and groups are the same across all platforms by redirecting account information requests to a User or Group object in eDirectory™.

AS Client API

The Authentication Services application programming interface (API). The AS Client API can be used by applications to perform functions, such as user ID/password verification, password changes, and obtaining information from eDirectory.

ASAM Directory

The file system directory that contains the binaries, configuration information, and other related files used by Identity Manager Fan-Out Driver components.

ASAM Master User Object

The User object that Core Driver components use for LDAP Bind operations.

ASAM System Container Object

The container object in eDirectory that holds component configuration and user and group management objects.

Audit Log

The log of occurrences of interest for auditing purposes. The Audit Log is maintained by the Audit Services component of each Core Driver.

Audit Services

The Core Driver component that performs logging.

Authentication Services

The set of services that provides access to information from eDirectory for authentication purposes. The principal components of Authentication Services are the Core Driver Authentication Services component, the Platform Services Process, the AS Client API, and the System Intercept.

Census

The collection of Enterprise User and Enterprise Group objects that represent users and groups from eDirectory that can be associated with a Platform Set. Object Services maintains the Census using provisioning events. Object Services on the primary Core Driver initially builds and periodically verifies the Census through the use of Trawls.

Census Search Object

An eDirectory object used to specify users and groups to be included in the Census.

Certificate

A digital object used to authenticate and secure SSL communications.

Certificate Services

The Core Driver component that issues certificates for other components.

Context

The location of an object within the eDirectory tree.

Core Driver

The components that provide Identity Provisioning and Authentication Services to platforms, and provide for the management of the Identity Manager Fan-Out Driver.

DES

Data Encryption Standard, approved by the U.S. government.

Enterprise Group (eGroup)

An object that represents a group of users that can be defined on a platform. Enterprise Group objects reside in the Census container.

Enterprise User (eUser)

An object that represents a user that can be defined on a platform. It is used by Authentication Services to locate the corresponding User object in eDirectory. Enterprise User objects reside in the Census container.

Entropy Daemon

A process that collects and provides cryptographically strong random data.

Event Driven Objects

A container in the ASAM System container that holds objects affected by provisioning events.

Event Journal Services

The Core Driver component that manages event information and provides provisioning events to Platform Receivers.

Event Subsystem

The Core Driver component that receives provisioning events from eDirectory and provides them to Object Services.

Identity Provisioning

The automatic provisioning of account related information from eDirectory to a target platform. The principal components of Identity Provisioning are the Event Subsystem, Object Services, Event Journal Services, Platform Receivers, and Receiver scripts.

Name Service Switch

A library for Linux and UNIX operating systems that implements a set of system functions used by programs to retrieve user and group account information. The Fan-Out Driver provides a Name Service Switch that allows a Linux or UNIX system to redirect account information from eDirectory.

Naming Exception

A conflict detected by Object Services between multiple User or Group objects having the same common name.

Object Services

The Core Driver component that maintains the Census.

Operational Log

A log of occurrences pertaining to the processing of a component. Audit Services maintains the Operational Log for the Core Driver.

PAM

Pluggable Authentication Module. PAM is a standard framework for UNIX defined by OSF RFC 86.0 that provides for authentication of users by facilities external to the original UNIX operating system.

Password Redirection

The process of ensuring that users' passwords are the same across all platforms by redirecting authentication requests to a User object in eDirectory.

Password Replication

The process of ensuring that users' passwords are the same across all platforms by replicating password information between the platforms and eDirectory.

Platform

A system that uses the Core Driver for Identity Provisioning, Authentication Services, or both.

Platform Configuration File

The file that contains configuration information for Platform Services. It identifies users to include or exclude from processing, and contains information used to locate the Core Driver servers.

Platform Object

The object in the ASAM System container that contains information about a platform.

Platform Receiver

The Platform Services component that obtains provisioning events from Event Journal Services and runs Receiver scripts to process them as appropriate for the platform.

Platform Services

The Identity Manager Fan-Out Driver components that run on a platform. These include the System Intercept, the Platform Services Process, the AS Client API, the Platform Receiver, and Receiver scripts.

Platform Services Cache Daemon

The process that runs on a platform and communicates with the Core Driver for Posix account information. Along with the Name Service Switch, the Platform Service Cache Daemon provides complete account redirection.

Platform Services Process

The process that runs on a platform and communicates with the Core Driver for Authentication Services. The Platform Services Process provides Core Driver server connection management, load balancing, and failover capability.

Platform Set

A group of platforms that share a common set of users and groups.

Platform Set Search Object

An eDirectory object used to specify users and groups to be included in a Platform Set.

Primary Core Driver

The Core Driver that serves the Web interface, provides environmental information during the installation of other Core Drivers, performs Census Trawls, and listens for events from eDirectory.

Provisioning Event

An event, such as an add, modify, or delete, originating from eDirectory, that pertains to a user account or group. The Event Subsystem subscribes to events from eDirectory and passes them to Object Services. Object Services records provisioning events in eUser and eGroup objects. Event Journal Services passes the events to Platform Receivers. Platform Receivers run Receiver scripts to process provisioning events as appropriate for the platform.

Provisioning Manager

The Core Driver component that comprises Object Services, Audit Services, Certificate Services, Event Journal Services, and Web Services. Platforms access the Provisioning Manager to obtain a security certificate and to obtain provisioning events.

Receiver Script

A script invoked by the Platform Receiver to process provisioning events. A fully functional set of base scripts, written in the customary scripting language for the platform, is provided. You can extend these scripts as appropriate for your needs.

Secondary Core Driver

Any Core Driver other than the primary Core Driver.

Secure Sockets Layer (SSL)

The communications protocol used for communication between components. SSL is a standard security protocol that provides communications privacy. The protocol allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, and message forgery.

System Intercept

A vendor-provided control point into the system that is used to interface with Authentication Services for a platform.

System Log

The operating system log of information that is of system-wide interest.

Trawl

The process used by Object Services to collect information from eDirectory to initially build and periodically ensure the validity of the Census.

Universal Time

By international agreement, the world-wide standard for systematic time keeping. Universal Time is based on the mean solar time at zero degrees longitude. Formerly known as GMT, Universal Time is abbreviated as Z or as UT.

User and Group Subtree

The high level container object that you specify during installation of the Core Driver that holds users and groups that can be included in the Census. The ASAM Master User is granted Supervisor rights to this container.

Web Application

The Web-based application that is used to administer and monitor the Identity Manager Fan-Out Driver. The application is accessed as a plug-in to the iManager interface.

Web Services

The Core Driver component that provides the Web interface.