

Novell Identity Manager Fan-Out Driver

3.5.1

September 28, 2007

PLATFORM SERVICES PLANNING
GUIDE AND REFERENCE

www.novell.com



Novell[®]

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 Omnibond Systems, LLC. All Rights Reserved. Licensed to Novell, Inc. Portions Copyright © 2004, 2007 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.

The Solaris* standard IO library has kernel limitations that interfere with the operation of the Provisioning Manager. Therefore, components for Solaris use the AT&T* SFIO library. Use of this library requires the following notice:

The authors of this software are Glenn Fowler, David Born and Kim-Phone Do.

Copyright (c) 1991, 1996, 1998, 2000, 2001, 2002 by AT&T Labs - Research.

Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software.

This software is being provided as is, without any express or implied warranty. In particular, neither the authors nor AT&T Labs make any representation or warranty of any kind concerning the merchantability of this software or its fitness for any particular purpose.

Contents

About This Guide	7
1 Platform Services Overview	9
1.1 Platform Services Component Summary	9
1.2 Authentication Services	10
1.3 Identity Provisioning	11
1.4 Account Redirection	11
1.5 The Platform Services Process	12
1.6 The Platform Services Cache Daemon	12
1.7 The System Intercept	12
1.8 The Platform Receiver	13
1.8.1 Modes of Operation	13
1.8.2 Selecting a Mode of Operation	14
1.9 Receiver Scripts	15
1.10 Standard Exclude List	16
2 Planning for Platform Services	19
2.1 Basic Considerations	19
2.2 Security Planning Considerations	19
2.2.1 Users, Passwords, and Groups	19
2.2.2 Connection Security	20
2.2.3 Administrative Password Resets	20
2.2.4 Securing the AS Client API	20
2.3 Planning Considerations for Authentication Services	20
2.4 Planning Considerations for Identity Provisioning	21
2.5 Planning Considerations for Password Replication Platforms	21
2.6 Planning Considerations for Account Redirection Platforms	22
3 The Platform Configuration File	23
3.1 Platform Configuration File Location	23
3.1.1 z/OS	23
3.1.2 NetWare	23
3.1.3 OS/400	23
3.1.4 UNIX	24
3.1.5 Windows	24
3.2 Platform Configuration File Syntax	24
3.3 Configuration Statements	24
3.3.1 ACF2.DISABLE Statement	25
3.3.2 ACF2.EXPIREWARN Statement	25
3.3.3 ADMINPASSWORD Statement	26
3.3.4 ADMINUSER Statement	26
3.3.5 AM.GROUP.INCLUDE Statement / AM.GROUP.EXCLUDE Statement	26
3.3.6 AM.USER.INCLUDE Statement / AM.USER.EXCLUDE Statement	27
3.3.7 AS.USER.INCLUDE Statement / AS.USER.EXCLUDE Statement	28
3.3.8 AS.USER.NONNDS Statement	28
3.3.9 ASAMDIR Statement	29
3.3.10 AUTHENTICATION Statement	29

3.3.11	CODEPAGE Statement	30
3.3.12	DEBUGLOGFILE Statement	30
3.3.13	DEBUGTOSTDOUT Statement	31
3.3.14	DIRECTTOAUTHENTICATION Statement	31
3.3.15	ENTROPY Statement	31
3.3.16	HONORMVSDISABLE Statement	32
3.3.17	IGNORESTANDARDEXCLUDES Statement	32
3.3.18	KEY Statement	32
3.3.19	LOCALE Statement	33
3.3.20	PASSWORDPROMPT Statement	33
3.3.21	PASSWORDPROMPTCURRENT Statement	33
3.3.22	PASSWORDPROMPTCHANGE Statement	34
3.3.23	PASSWORDPROMPTCHANGEAGAIN Statement	34
3.3.24	PLATFORMNAME Statement	34
3.3.25	PASSWORDSOURCE Statement	34
3.3.26	PROVISIONING Statement	35
3.3.27	RUNMODE Statement	35
3.3.28	SECURITY Statement	36
3.3.29	SMF Statement	36
3.3.30	SYSLOGFACILITY Statement	36
3.3.31	TRACEFILE Statement	37
3.3.32	TRACETOSTDOUT Statement	37
3.3.33	UPDATEPASSWORD Statement	38
3.3.34	UPDATESAMBA Statement	38
3.3.35	USEFILEIPC Statement	38
3.4	Using Include and Exclude Configuration Statements	39
3.4.1	Mask Characters and Examples	39
3.4.2	Rules by Which Masks Are Matched Against User IDs and Groups	40
A	Obtaining a Platform Certificate During Unattended Installation	41
B	Installing and Configuring the Novell Client Password Intercept	43

About This Guide

This guide provides you with the information you need to plan for deploying Novell® Identity Manager Fan-Out driver Platform Services.

The Fan-Out driver supports multi-platform implementation of Novell Identity Manager 3.5.1, 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.

This guide also contains configuration reference information to complement the platform administration guides.

The guide includes the following sections:

- ♦ [Chapter 1, “Platform Services Overview,” on page 9](#)
- ♦ [Chapter 2, “Planning for Platform Services,” on page 19](#)
- ♦ [Chapter 3, “The Platform Configuration File,” on page 23](#)
- ♦ [Appendix A, “Obtaining a Platform Certificate During Unattended Installation,” on page 41](#)
- ♦ [Appendix B, “Installing and Configuring the Novell Client Password Intercept,” on page 43](#)

Audience

This guide is for system administrators and others who plan, install, configure, and use 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 3.5.1 Drivers Documentation Web site \(http://www.novell.com/documentation/idm35drivers\)](http://www.novell.com/documentation/idm35drivers).

Additional Documentation

For additional documentation about Identity Manager drivers, see [the Identity Manager 3.5.1 Drivers Documentation Web site \(http://www.novell.com/documentation/idm35drivers\)](http://www.novell.com/documentation/idm35drivers).

For documentation about Identity Manager, see [the Identity Manager 3.5.1 Documentation Web site \(http://www.novell.com/documentation/idm35\)](http://www.novell.com/documentation/idm35).

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.

Platform Services Overview

1

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 the *Concepts and Facilities Guide* before proceeding.

Topics in this section are

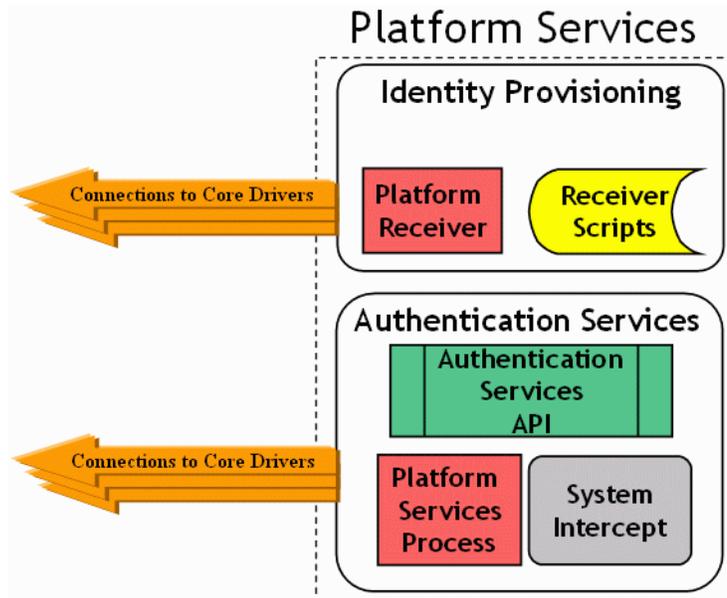
- ◆ [Section 1.1, “Platform Services Component Summary,” on page 9](#)
- ◆ [Section 1.2, “Authentication Services,” on page 10](#)
- ◆ [Section 1.3, “Identity Provisioning,” on page 11](#)
- ◆ [Section 1.4, “Account Redirection,” on page 11](#)
- ◆ [Section 1.5, “The Platform Services Process,” on page 12](#)
- ◆ [Section 1.6, “The Platform Services Cache Daemon,” on page 12](#)
- ◆ [Section 1.7, “The System Intercept,” on page 12](#)
- ◆ [Section 1.8, “The Platform Receiver,” on page 13](#)
- ◆ [Section 1.9, “Receiver Scripts,” on page 15](#)
- ◆ [Section 1.10, “Standard Exclude List,” on page 16](#)

1.1 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 the *Concepts and Facilities Guide*.

Figure 1-1 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 the *API Developer Guide*.

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.

1.2 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* and UNIX systems can redirect password verification and password changes through Authentication Services to eDirectory. An IBM* 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 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* and NetWare® 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 3, “The Platform Configuration File,” on page 23](#). For more information about the standard exclude list, see [Section 1.10, “Standard Exclude List,” on page 16](#).

1.3 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 3, “The Platform Configuration File,” on page 23](#). For more information about the standard exclude list, see [Section 1.10, “Standard Exclude List,” on page 16](#).

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

1.4 Account Redirection

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.

1.5 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 the *API Developer Guide*.

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 3, “The Platform Configuration File,” on page 23](#).

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 31](#).

1.6 The Platform Services Cache Daemon

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.

1.7 The System Intercept

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

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 *Platform Services Administration Guide for z/OS*.

System integration of Platform Services for most 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 *Platform Services Administration Guide for Linux and UNIX*.

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 higher.

Password changes from a Windows system are provided to the core driver through an extension to the Novell Client™. For additional information, see [Appendix B, “Installing and Configuring the Novell Client Password Intercept,” on page 43](#).

Password changes made by NDK applications on NetWare servers are provided to the core driver by the NetWare Password Intercept.

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 *Platform Services Administration Guide for OS/400*.

1.8 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 3, “The Platform Configuration File,” on page 23](#).

1.8.1 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.

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.

1.8.2 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 35](#).

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.

1.9 Receiver Scripts

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.

For additional information about Receiver scripts on your platform, see the administration guide for your platform-OS type.

1.10 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 3.4, “Using Include and Exclude Configuration Statements,” on page 39](#)
- ◆ [“AM.GROUP.INCLUDE Statement / AM.GROUP.EXCLUDE Statement” on page 26](#)
- ◆ [“AM.USER.INCLUDE Statement / AM.USER.EXCLUDE Statement” on page 27](#)
- ◆ [“AS.USER.INCLUDE Statement / AS.USER.EXCLUDE Statement” on page 28](#)

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
QNTF	QPEX	QPGMR
QPM400	QRJE	QSECOFR
QSNADS	QSPL	QSPLJOB
QSRV	QSRVBAS	QSVCDRCTR
QSYS	QSYSOPR	QTCM
QTCP	QTFTP	QTMHHTTP1
QTMHHTTP	QTMPLPD	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
xfx		

Planning for Platform Services

2

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 the *Concepts and Facilities Guide* before proceeding.

- ◆ [Section 2.1, “Basic Considerations,” on page 19](#)
- ◆ [Section 2.2, “Security Planning Considerations,” on page 19](#)
- ◆ [Section 2.3, “Planning Considerations for Authentication Services,” on page 20](#)
- ◆ [Section 2.4, “Planning Considerations for Identity Provisioning,” on page 21](#)
- ◆ [Section 2.5, “Planning Considerations for Password Replication Platforms,” on page 21](#)
- ◆ [Section 2.6, “Planning Considerations for Account Redirection Platforms,” on page 22](#)

2.1 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 the *Core Driver Administration Guide*, 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 3.5.1 Drivers Documentation Web site \(http://www.novell.com/documentation/idm35drivers\)](http://www.novell.com/documentation/idm35drivers). From this index page, you can select a readme file associated with the platform(s) for which you need support.

2.2 Security Planning Considerations

Topics in this section include

- ◆ [Section 2.2.1, “Users, Passwords, and Groups,” on page 19](#)
- ◆ [Section 2.2.2, “Connection Security,” on page 20](#)
- ◆ [Section 2.2.3, “Administrative Password Resets,” on page 20](#)
- ◆ [Section 2.2.4, “Securing the AS Client API,” on page 20](#)

2.2.1 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 1.3, “Identity Provisioning,” on page 11](#).

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 *Platform Services Administration Guide for z/OS*.

2.2.2 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.

2.2.3 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.

2.2.4 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.

2.3 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 31](#).
- ♦ 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 28](#).
- ♦ 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 3.3.10, “AUTHENTICATION Statement,” on page 29](#).

2.4 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 27](#) and [“AM.GROUP.INCLUDE Statement / AM.GROUP.EXCLUDE Statement” on page 26](#).
- ♦ 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 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.
- ♦ If yours is an OS/400 platform, you might need to map eDirectory attribute names to System Distribution Directory names. For more information, see the *Platform Services Administration Guide for OS/400*.
- ♦ You must specify which core drivers are used for Identity Provisioning. For details, see [Section 3.3.26, “PROVISIONING Statement,” on page 35](#).
- ♦ You must choose the mode of operation used by the Platform Receiver to obtain events. For details, see [“Modes of Operation” on page 13](#) and [“Selecting a Mode of Operation” on page 14](#).

2.5 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 the Maintain Password Case configuration parameter in the *Core Services Administration Guide*.

- ◆ 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 “Configuring Platforms” in the *Core Driver Administration Guide*.

2.6 Planning Considerations for Account Redirection Platforms

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

3

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 3.1, “Platform Configuration File Location,” on page 23](#)
- ♦ [Section 3.2, “Platform Configuration File Syntax,” on page 24](#)
- ♦ [Section 3.3, “Configuration Statements,” on page 24](#)
- ♦ [Section 3.4, “Using Include and Exclude Configuration Statements,” on page 39](#)

3.1 Platform Configuration File Location

The location of the platform configuration file depends on which operating system the platform is using.

3.1.1 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. For more information about specifying the location of the configuration file for z/OS, see the *Platform Services Administration Guide for z/OS*.

3.1.2 NetWare

For NetWare®, the default configuration file is `sys:asam\data\asamplat.conf`. You can use a command line parameter to specify a different platform configuration file.

3.1.3 OS/400

For OS/400, 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. For more information about specifying the location of the configuration file for OS/400, see the *Platform Services Administration Guide for OS/400*.

3.1.4 UNIX

For 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. For more information about specifying the location of the configuration file for UNIX, see the *Platform Services Administration Guide for Linux and UNIX*.

3.1.5 Windows

For Windows, the configuration file is `asamplat.conf` in the Windows operating system directory. To determine the location of your Windows operating system directory, enter `echo %windir%` at a command prompt.

3.2 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
```

3.3 Configuration Statements

This section describes the following platform configuration file statements:

- ◆ [Section 3.3.1, “ACF2.DISABLE Statement,” on page 25](#)
- ◆ [Section 3.3.2, “ACF2.EXPIREWARN Statement,” on page 25](#)
- ◆ [Section 3.3.3, “ADMINPASSWORD Statement,” on page 26](#)
- ◆ [Section 3.3.4, “ADMINUSER Statement,” on page 26](#)
- ◆ [Section 3.3.5, “AM.GROUP.INCLUDE Statement / AM.GROUP.EXCLUDE Statement,” on page 26](#)
- ◆ [Section 3.3.6, “AM.USER.INCLUDE Statement / AM.USER.EXCLUDE Statement,” on page 27](#)
- ◆ [Section 3.3.7, “AS.USER.INCLUDE Statement / AS.USER.EXCLUDE Statement,” on page 28](#)
- ◆ [Section 3.3.8, “AS.USER.NONNDS Statement,” on page 28](#)
- ◆ [Section 3.3.9, “ASAMDIR Statement,” on page 29](#)
- ◆ [Section 3.3.10, “AUTHENTICATION Statement,” on page 29](#)
- ◆ [Section 3.3.11, “CODEPAGE Statement,” on page 30](#)
- ◆ [Section 3.3.12, “DEBUGLOGFILE Statement,” on page 30](#)
- ◆ [Section 3.3.13, “DEBUGTOSTDOUT Statement,” on page 31](#)

- ◆ Section 3.3.14, “DIRECTTOAUTHENTICATION Statement,” on page 31
- ◆ Section 3.3.15, “ENTROPY Statement,” on page 31
- ◆ Section 3.3.16, “HONORMVSDISABLE Statement,” on page 32
- ◆ Section 3.3.17, “IGNORESTANDARDEXCLUDES Statement,” on page 32
- ◆ Section 3.3.18, “KEY Statement,” on page 32
- ◆ Section 3.3.19, “LOCALE Statement,” on page 33
- ◆ Section 3.3.20, “PASSWORDPROMPT Statement,” on page 33
- ◆ Section 3.3.21, “PASSWORDPROMPTCURRENT Statement,” on page 33
- ◆ Section 3.3.22, “PASSWORDPROMPTCHANGE Statement,” on page 34
- ◆ Section 3.3.23, “PASSWORDPROMPTCHANGEAGAIN Statement,” on page 34
- ◆ Section 3.3.24, “PLATFORMNAME Statement,” on page 34
- ◆ Section 3.3.25, “PASSWORDSOURCE Statement,” on page 34
- ◆ Section 3.3.26, “PROVISIONING Statement,” on page 35
- ◆ Section 3.3.27, “RUNMODE Statement,” on page 35
- ◆ Section 3.3.28, “SECURITY Statement,” on page 36
- ◆ Section 3.3.29, “SMF Statement,” on page 36
- ◆ Section 3.3.30, “SYSLOGFACILITY Statement,” on page 36
- ◆ Section 3.3.31, “TRACEFILE Statement,” on page 37
- ◆ Section 3.3.32, “TRACETOSTDOUT Statement,” on page 37
- ◆ Section 3.3.33, “UPDATEPASSWORD Statement,” on page 38
- ◆ Section 3.3.34, “UPDATESAMBA Statement,” on page 38
- ◆ Section 3.3.35, “USEFILEIPC Statement,” on page 38

3.3.1 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
```

3.3.2 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
```

3.3.3 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
```

3.3.4 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
```

3.3.5 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 39](#).

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 32](#).

Do not code both an `AM.GROUP.INCLUDE` statement and an `AM.GROUP.EXCLUDE` statement.

For more information, see [Section 3.4, “Using Include and Exclude Configuration Statements,” on page 39](#).

Example:

```
AM.GROUP.EXCLUDE sales, mkt*
```

3.3.6 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 39](#).

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 32](#).

Do not code both an `AM.USER.INCLUDE *` statement and an `AM.USER.EXCLUDE *` statement.

Identity Manager Fan-Out driver UNIX platforms always manage root locally.

For more information, see [Section 3.4, “Using Include and Exclude Configuration Statements,” on page 39](#).

Example:

```
AM.USER.EXCLUDE act*, billing%, sys*, sales48
```

3.3.7 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 39](#).

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 32](#).

Do not code both an AS . USER . INCLUDE * statement and an AS . USER . EXCLUDE * statement.

For more information, see [Section 3.4, “Using Include and Exclude Configuration Statements,” on page 39](#).

Example:

```
AS.USER.EXCLUDE act*, billing%, sys*, sales48
```

3.3.8 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 3-1 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
```

3.3.9 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
- ◆ **NetWare:** sys:asam
- ◆ **OS/400:** /usr/local/ASAM
- ◆ **UNIX:** /usr/local/ASAM
- ◆ **Windows:** c:\novell\asam

Example:

```
ASAMDIR c:\novell\asam
```

3.3.10 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
```

3.3.11 CODEPAGE Statement

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
```

3.3.12 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.

For information about troubleshooting, see the applicable administration guide.

Example:

```
DEBUGLOGFILE /usr/local/ASAM/debug.txt
```

3.3.13 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
```

For information about troubleshooting, see the applicable administration guide.

Example:

```
DEBUGTOSTDOUT
```

3.3.14 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
```

3.3.15 ENTROPY Statement

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
```

3.3.16 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
```

3.3.17 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 1.10, "Standard Exclude List," on page 16](#).

Syntax:

```
IGNORESTANDARDEXCLUDES
```

Example:

```
IGNORESTANDARDEXCLUDES
```

3.3.18 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 the *Core Driver Administration Guide*.

Example:

```
KEY 0f4b9692d8bca5f6
```

3.3.19 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
```

3.3.20 PASSWORDPROMPT Statement

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: "
```

3.3.21 PASSWORDPROMPTCURRENT Statement

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: "
```

3.3.22 PASSWORDPROMPTCHANGE Statement

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: "
```

3.3.23 PASSWORDPROMPTCHANGEAGAIN Statement

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: "
```

3.3.24 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
```

3.3.25 PASSWORDSOURCE Statement

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
```

3.3.26 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
```

3.3.27 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 3-2 *Values For Specified Mode*

Value	Description
PERSISTENT	The Platform Receiver uses Persistent Mode.

Value	Description
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 13](#).

Example:

```
RUNMODE polling
```

3.3.28 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
```

3.3.29 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
```

3.3.30 SYSLOGFACILITY Statement

UNIX only.

The `SYSLOGFACILITY` statement specifies the `SYSLOG` facility name to use for message logging on UNIX systems.

Syntax:

```
SYSLOGFACILITY FacilityName
```

FacilityName specifies the `SYSLOG` facility to use for logging messages. The possible values for a specific 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
```

3.3.31 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.

For information about troubleshooting, see the applicable administration guide.

Example:

```
TRACEFILE c:\novell\asam\debug.txt
```

3.3.32 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
```

For information about troubleshooting, see the applicable administration guide.

Example:

TRACETOSTDOUT

3.3.33 UPDATEPASSWORD Statement

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
```

3.3.34 UPDATESAMBA Statement

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
```

3.3.35 USEFILEIPC Statement

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
```

3.4 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 26](#)
- ◆ [“AM.USER.INCLUDE Statement / AM.USER.EXCLUDE Statement” on page 27](#)
- ◆ [“AS.USER.INCLUDE Statement / AS.USER.EXCLUDE Statement” on page 28](#)

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 32](#). For a list of the users and groups in the standard exclude list, see [Section 1.10, “Standard Exclude List,” on page 16](#).

3.4.1 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 3-3 *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 3-4 Example Masks

Mask	Matches
Z*	ZEBRA ZULU ZED ZABRZE Z9
Zn*	Z9 Z9WWW
\s\al\al\?	#BB29 #BB2A #AB9_
\aFF	AFF BFF CFF DFF EFF
*	All strings
%%%%%%%%%	All five-character strings
?????	All five-character strings
\?\?\?\?\?	All five-character strings

3.4.2 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 32](#).
- ◆ Do not code both an INCLUDE * statement and an EXCLUDE * statement of the same type.

Obtaining a Platform Certificate During Unattended Installation

A

The installation process for Platform Services of the Novell® Identity Manager Fan-Out driver requires obtaining a security certificate from the core driver. This is done by running a Platform Services SSL component with a special command line parameter.

This process normally prompts you for the common name of the Platform object, and the fully distinguished name and password of an eDirectory™ user with Read and Create object rights to the ASAM System container.

These prompts are undesirable when scripting large numbers of platform installations. You can avoid these prompts by specifying the information in the platform configuration file.

Three platform configuration file statements can be used to provide automated responses for the prompts generated when obtaining a platform security certificate. These statements are

- ◆ [Section 3.3.24, “PLATFORMNAME Statement,” on page 34](#)
- ◆ [“ADMINUSER Statement” on page 26](#)
- ◆ [“ADMINPASSWORD Statement” on page 26](#)

Installing and Configuring the Novell Client Password Intercept

B

If you use password replication in your Novell® Identity Manager Fan-Out driver configuration, 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. You must install the Novell Client™ Password Intercept on each Windows workstation that uses eDirectory.

To install and configure the Novell Client Password Intercept:

- 1 Verify that the Novell Client version is current.
 - ♦ Version 4.8 or later for Windows NT*, Windows 2000, and Windows XP
 - ♦ Version 3.3 or later for Windows 95 and Windows 98
- 2 Retrieve the Novell Client Password Intercept installation program from the distribution package `intercepts\client32` directory.
- 3 Close all running applications.
- 4 Run the installation program and respond to the prompts.

Be sure to use the correct core driver port number.

If you receive an error message stating that the `asamplat.conf` file already exists, you can usually ignore it, because a previous installation has configured this file. To verify this, look at `asamplat.conf` in your Windows directory, and ensure that all core driver configurations are correct.

- 5 Reboot to let the in-use files be copied.

IMPORTANT: If you upgrade the Novell Client, you must reinstall the password intercept.

When you install the Novell Client Password Intercept on a Windows system, the following actions are performed:

1. The following files are copied to the Windows system directory:
 - ♦ `ampm.dll`
 - ♦ `amserver.exe`
 - ♦ `ascauth.dll`
 - ♦ `audwin32.dll`
 - ♦ `calwin32.dll`
 - ♦ `clnwin32.dll`
 - ♦ `clnwinth.dll` (if the system is Windows 95 or Windows 98)
 - ♦ `clxwin32.dll`
 - ♦ `locwin32.dll`

- ♦ ncpwin32.dll
 - ♦ netwin32.dll
2. The following registry entry is created:
 - ♦ HKEY_LOCAL_MACHINE\SOFTWARE\Novell\ASAM\3.0\
NovellClientIntercept
 3. The `asamplat.conf` file is created or updated. The default location for `asamplat.conf` is in the Windows operating system directory. You can enter the command `echo %windir%` at a command prompt to determine the location of your Windows operating system directory.