

# Novell Nsure™ Identity Manager Fan-Out Driver

3.1

October 27, 2004

---

PLATFORM SERVICES ADMINISTRATION  
GUIDE FOR LINUX\* AND UNIX\*

[www.novell.com](http://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.

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

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

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 Korn and Kiem-Phong Vo.

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.

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

[www.novell.com](http://www.novell.com)

Nsure Identity Manager Fan-Out Driver 3.1 Platform Services Administration Guide for Linux and UNIX  
[October 27, 2004](#)

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

## **Novell Trademarks**

eDirectory is a trademark of Novell, Inc.

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

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

Nsure is a trademark of Novell, Inc.

## **Third-Party Trademarks**

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



# Contents

- About This Guide** **7**
  
- 1 Installing Platform Services** **9**
  - About Platform Services for UNIX . . . . . 9
  - Secure Sockets Layer Entropy Requirements . . . . . 10
  - The Platform Services Process . . . . . 10
  - The System Intercept . . . . . 10
  - The Platform Receiver . . . . . 11
  - Receiver Scripts. . . . . 11
  - Authentication Services . . . . . 12
  - Platform Services Installation Procedure . . . . . 12
  - Uninstalling Platform Services . . . . . 12
  
- 2 Configuring and Administering Platform Services** **13**
  - Platform Certificate Management. . . . . 13
  - Administering Platform Services . . . . . 14
    - PAM Configuration Notes . . . . . 14
    - UNIX Password Management . . . . . 16
    - Managing the UNIX Platform Services Process . . . . . 19
    - Managing the UNIX Platform Receiver . . . . . 20
  
- 3 Troubleshooting Platform Services** **23**
  - Obtaining Debugging Output . . . . . 23
    - Debugging the UNIX Platform Services Process and Platform Receiver . . . . . 23
  - Troubleshooting Authentication Services. . . . . 24
  - Troubleshooting Identity Provisioning . . . . . 24
  - Troubleshooting Network Issues . . . . . 24
  - Troubleshooting Platform Services Installation. . . . . 24
  
- Index** **25**



# About This Guide

This guide provides you with the information you need to install, configure, administer, and troubleshoot Novell® Nsure™ Identity Manager Fan-Out driver Platform Services on UNIX and Linux platforms. This guide assumes that you have knowledge of eDirectory™ and the platform operating system, and that you are familiar with the concepts and facilities of the driver. It is also assumed that you have reviewed the *Platform Services Planning Guide and Reference* and have completed the planning phase described therein.

This guide is divided into the following sections:

- ◆ Chapter 1, “Installing Platform Services,” on page 9
- ◆ Chapter 2, “Configuring and Administering Platform Services,” on page 13
- ◆ Chapter 3, “Troubleshooting Platform Services,” on page 23

## Additional Documentation

The following publications contain information about the Identity Manager Fan-Out driver. These publications are available at the [Identity Manager Driver Web site \(http://www.novell.com/documentation/dirxml/drivers\)](http://www.novell.com/documentation/dirxml/drivers).

*Concepts and Facilities Guide*

*Core Driver Administration Guide*

*Platform Services Planning Guide and Reference*

*Platform Services Administration Guide for Linux and UNIX*

*Platform Services Administration Guide for MVS*

*Platform Services Administration Guide for OS/400*

*NetWare Intercept and API Administration Guide*

*API Developer Guide*

*Messages Reference*

*Core Driver Quick Start Guide for Linux and Solaris*

*Core Driver Quick Start Guide for NetWare*

*Core Driver Quick Start Guide for Windows*

*Platform Services Quick Start Guide for AIX*

*Platform Services Quick Start Guide for FreeBSD, HP-UX, Linux, and Solaris*

*Platform Services Quick Start Guide for MVS CA-ACF2*

*Platform Services Quick Start Guide for MVS CA-Top Secret*

*Platform Services Quick Start Guide for MVS RACF*

*Platform Services Quick Start Guide for OS/400*

*NetWare Intercept and API Quick Start Guide*

Documentation for related products, such as Identity Manager and eDirectory, is available at the [Novell Documentation Web site \(http://www.novell.com/documentation\)](http://www.novell.com/documentation).

## **Documentation Updates**

For the most recent versions of Identity Manager Fan-Out driver documentation, see the [Identity Manager Driver Web site \(http://www.novell.com/documentation/dirxmldrivers\)](http://www.novell.com/documentation/dirxmldrivers).

## **Documentation Conventions**

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

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

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

## **User Comments**

We want to hear your comments and suggestions about this manual and the other documentation included with the driver. To contact us, send e-mail to [namdoc@novell.com](mailto:namdoc@novell.com).



# 1

## Installing Platform Services

The installation and setup of Novell® Nsure™ Identity Manager Fan-Out driver Platform Services includes tasks performed on the platform, and core driver administration tasks. This section describes the installation tasks that are performed on the platform system. For details about platform configuration and administration tasks, see [Chapter 2, “Configuring and Administering Platform Services,” on page 13](#).

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 the *Core Driver Administration Guide*.

After the planning process has been completed, installation of Platform Services for Linux and UNIX 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 are

- ♦ [“About Platform Services for UNIX” on page 9](#)
- ♦ [“Platform Services Installation Procedure” on page 12](#)
- ♦ [“Uninstalling Platform Services” on page 12](#)

### About Platform Services for UNIX

Platform Services for UNIX consists of three major components.

- ♦ **Platform Services Process:** The Platform Services Process receives requests from other processes and manages communications with one or more core drivers for Authentication Services.
- ♦ **System Intercept:** The System Intercept is implemented in most UNIX systems using a Pluggable Authentication Module (PAM). The Platform Services PAM module communicates with the Platform Services Process for password verification and password changes.
- ♦ **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.

## Secure Sockets Layer Entropy Requirements

Secure Sockets Layer (SSL), used by Platform Services for communication with core drivers, requires a source of entropy. Some UNIX implementations provide a `/dev/random` device for entropy. If your UNIX implementation does not include a `/dev/random` device, you must install an entropy daemon. You must also include an `ENTROPY` statement in your platform configuration file to specify the source of entropy. For information about the platform configuration file, see the *Platform Services Planning Guide and Reference*.

The PRNGD entropy daemon can be installed from the `/prngd` directory of the distribution media.

Solaris versions before Solaris 9 do not include a `/dev/random` device. Sun\* has released this functionality for versions 2.6 onward in Patch ID 112438-01.

## The Platform Services Process

The Platform Services Process provides an interface for the System Intercept and the AS Client API to one or more core drivers for Authentication Services.

The Platform Services Process is called whenever a user attempts to enter the system using a user ID and password or when a user attempts to change the password. Such a request is passed from the system intercept to the Platform Services Process, which then communicates with a core driver and returns a response.

The Platform Services Process performs the following tasks:

- ◆ Handles password check and password change requests from users
- ◆ Communicates with core drivers for Authentication Services
- ◆ Redirects Authentication Services requests to another core driver if a core driver is unreachable or returns an unexpected error
- ◆ Gathers and logs performance statistics

The Platform Services Process communicates with core drivers using Secure Sockets Layer (SSL).

Start the Platform Services Process during system startup and stop it during system shutdown.

The Platform Services Process reads its configuration information from `ASAM/data/asamplat.conf`, the platform configuration file. For details about the platform configuration file, see the *Platform Services Planning Guide and Reference*.

The Platform Services Process logs messages to the SYSLOG facility specified by the `SYSLOGFACILITY` statement in the platform configuration file. For details about the platform configuration file, see the *Platform Services Planning Guide and Reference*.

## The System Intercept

The Platform Services System Intercept communicates with the Platform Services Process for password verification and password changes.

The System Intercept is implemented in most UNIX systems using a Pluggable Authentication Module (PAM). Platform Services for AIX uses the Loadable Authentication Module (LAM) system provided by AIX.

## 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. You can use the CODEPAGE statement in the platform configuration file to configure the Platform Receiver to convert data using a specified code page. For details about the platform configuration file, see the *Platform Services Planning Guide and Reference*.

Run the Platform Receiver on a schedule that is appropriate for your requirements. For details about Platform Receiver operation, see the *Platform Services Planning Guide and Reference*.

The Platform Receiver reads its configuration information from ASAM/data/asamplat.conf, the platform configuration file. For details about the platform configuration file, see the *Platform Services Planning Guide and Reference*.

The Platform Receiver logs messages to the SYSLOG facility specified by the SYSLOGFACILITY statement in the platform configuration file. For details about the platform configuration file, see the *Platform Services Planning Guide and Reference*.

When the Platform Receiver successfully updates a password in the local security system or Samba password file, it logs a message to SYSLOG.

## Receiver Scripts

Receiver scripts for UNIX platforms are implemented as shell scripts. The Platform Receiver runs the scripts from ASAM/bin/PlatformServices/PlatformReceiver/scripts.

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 in environment variables as shown in the following example:

```
ENTERPRISEUSERNAME=bob
```

User names and group names are checked for validity before they are mapped to environment variables. 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 the *Platform Services Planning Guide and Reference* and the scripts themselves.

## Authentication Services

Authentication Services for UNIX redirects authentication requests to eDirectory™ and can replicate passwords from eDirectory.

When a password for a user associated with a UNIX system that uses password replication is changed in eDirectory, 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 the Maintain Password Case configuration parameter in the *Core Driver Administration Guide*.

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 the Include/Exclude lists for both the Platform Services Process and the Platform Receiver symmetrically if the platform uses password replication.

For more information about password management by Platform Services for UNIX, see “[UNIX Password Management](#)” on page 16.

## Platform Services Installation Procedure

For detailed step-by-step instructions for installing Platform Services, see the *Platform Services Quick Start Guide* for your operating system type.

- ◆ *Platform Services Quick Start Guide for AIX*
- ◆ *Platform Services Quick Start Guide for FreeBSD, HP-UX, Linux, and Solaris*

## Uninstalling Platform Services

To remove Platform Services from a UNIX system:

- 1** Stop the Platform Services Process.  
For details, see “[Starting and Stopping the Platform Services Process](#)” on page 19.
- 2** Stop the Platform Receiver.  
For details, see “[Starting and Stopping the Platform Receiver](#)” on page 20.
- 3** Remove the Platform Services Process and Platform Receiver from any system startup, shutdown, and scheduling procedures as appropriate.
- 4** In the Web interface, remove the Platform object for the UNIX system.
- 5** Remove the ASAM directory created by Platform Services installation.

# 2

## Configuring and Administering Platform Services

The topics in this section provide information about managing Novell® Nsure™ Identity Manager Fan-Out driver Platform Services.

- ♦ [“Platform Certificate Management” on page 13](#)
- ♦ [“Administering Platform Services” on page 14](#)

### 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 either the Platform Services Process or the Platform Receiver with the `-s` 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.

# Administering Platform Services

- ◆ “PAM Configuration Notes” on page 14
- ◆ “UNIX Password Management” on page 16
- ◆ “Managing the UNIX Platform Services Process” on page 19
- ◆ “Managing the UNIX Platform Receiver” on page 20

## PAM Configuration Notes

Identity Manager Fan-Out driver platforms for most UNIX implementations make use of the Pluggable Authentication Module (PAM) framework for system-entry services, such as login. PAM is defined by OSF RFC 86.0.

When a service (login, ftp, user written application, etc.) makes a call to the PAM API, the request is forwarded to the appropriate authentication module based on the specifications you have made in the PAM configuration file, normally `/etc/pam.conf`. (Some Linux implementations separate the PAM parameters for various services into files in the `/etc/pam.d/` directory.) A sample `pam.conf` file for Platform Services is included in each UNIX platform distribution.

## Stacking Multiple Schemes

The PAM architecture enables authentication by multiple authentication services through stacking. Stacking service modules can force users to authenticate to several authentication services, possibly using different passwords, or it can allow users the opportunity to authenticate using any one of several methods or some combination of methods.

It is very important to understand certain return codes returned by services in the stack, because these return codes are used in conjunction with the control flag to determine the behavior of the authentication flow within the stack.

Always test the logical flow of your configuration. *Some configurations could allow users to log in without passwords, while others could prevent login by anyone, including root.* Many service modules, including the Platform Services service module, treat root differently from other users.

## Where to Find More Detailed Information about PAM

- ◆ For detailed information about PAM, see RFC 86.0, included in each UNIX Platform Services distribution package.
- ◆ For PAM configuration file information specific to your UNIX implementation, see the man pages, typically `man pam.conf`.
- ◆ For Linux-PAM documentation on the Web, see the [Linux Kernel site \(http://www.kernel.org/pub\)](http://www.kernel.org/pub).

## Overview of pam.conf

An entry in pam.conf has the form:

```
service module_type control_flag module_path parameters
```

- ◆ **Service:** The name of a service, such as login and ftp. The specification *other* indicates the module to be used by all other applications not specified in the file.
- ◆ **Module\_Type:** The type of PAM function.
  - ◆ **auth:** User authentication
  - ◆ **account:** Account access, such as expiration and time of day restrictions
  - ◆ **session:** Session management accounting
  - ◆ **password:** Password change
- ◆ **Control\_flag:** Determines continuation or failure behavior of the module. This is especially important if stacking is used.
  - ◆ **required:** This module must return success in order to have the overall result be successful. If this module fails, stack processing continues and hides where the failure occurred from the user.
  - ◆ **requisite:** Like required, except stack processing fails immediately if this module fails. Requisite is not used in many versions of PAM.
  - ◆ **sufficient:** If this module is successful, skip the remaining modules in the stack, even if their control flags indicate they are required. If this module fails, the overall result might be determined by other modules in this stack.
  - ◆ **optional:** If this module fails, the overall result can be successful if another module in this stack returns success. If this module succeeds, the overall result might be determined by other modules in this stack. If no other modules are required, then a success by an optional module causes success for the stack.

- ◆ **Module\_Path:** The pathname of the module to be invoked for the function.

The PAM service module for Platform Services, `pam_ascauth`, checks the user ID to see if it is in the Exclude list or is the user ID root (unless the `root_nds` PAM parameter is specified). If either condition is met, then `pam_ascauth` returns `PAM_IGNORE`, which has the same effect as the Platform Services authentication service not being included in the stack.

- ◆ **Parameters:** Command line parameters to be passed to the module. The developer of a module can use these any way desired, but the PAM framework recommends that several parameters always be supported. Among these are *use\_first\_pass* (use the same password as that used by the first module that asked for one) and *try\_first\_pass* (like *use\_first\_pass*, but prompt if it is not valid).

The Platform Services PAM module supports several other parameters. For details about these parameters, see [“Platform Services PAM Module Parameters” on page 16](#).

## Example pam.conf File Fragment

The following is a fragment from the sample pam.conf file that is provided with Platform Services for Solaris.

```
login auth sufficient /usr/lib/security/pam_ascauth.so.1 stats
login auth required /usr/lib/security/pam_unix.so.1 try_first_pass
```

This fragment deals with authenticating users of the login service.

The first line specifies the Platform Services PAM module, `pam_ascauth.so.1`, passing it a parameter of `stats`, which causes it to write additional statistics records about its processing to syslog. If `pam_ascauth.so.1` returns success, the user is granted access to the system. If `pam_ascauth.so.1` returns failure, the next module is called.

The second line calls the native Solaris PAM module. It is invoked only if the Platform Services PAM module returns failure. This module first tries the password that was entered by the user and rejected by the driver. If the password is not valid, the user is prompted for the local UNIX system password. If that password is rejected, the user is not granted access to the system. Even if this module returns success, the next module in the stack, if any, is called.

**WARNING:** You must be familiar with PAM configuration for your particular UNIX implementation before attempting to create your own PAM configuration files. Take extreme care in configuring PAM on your systems. Mistakes here can result in major security exposures.

## Platform Services PAM Module Parameters

You can specify the following parameters to the Platform Services PAM module to control its operation:

<code>conf</code>	Specifies where the platform configuration file is located. The default location is <code>/usr/local/ASAM/data/asamplat.conf</code> . Example: <code>conf=/usr/local/ASAM/data/myplat.conf</code>
<code>stats</code>	Causes the PAM module to write syslog records containing authentication statistics. The records contain information on what type of request was made, the result, and the elapsed time to complete the request.
<code>debug</code>	Causes the PAM module to write debugging records to syslog.
<code>root_nds</code>	Forces the root user to be authenticated and managed by the Identity Manager Fan-Out driver. This behavior is not normally desirable. If this option is not specified, the root user is managed by the local security mechanism.

For more information about PAM module configuration, see [“Overview of pam.conf” on page 15](#).

## UNIX Password Management

Platform Services for UNIX redirects password check and password change requests to eDirectory™.

You can optionally store passwords into the local security system and the Samba password file upon a successful check or change password request. You can also replicate password change information from eDirectory onto your UNIX platform.

Given appropriate security system configuration, maintaining current password information in the local security system allows the user to log in with the eDirectory password if the driver, eDirectory, or the network is not available.

### Password Redirection

Password redirection for most UNIX systems is accomplished using the Pluggable Authentication Module (PAM) framework. Platform Services for AIX uses the Loadable Authentication Module (LAM) system provided by AIX.

For more information about PAM, see [“PAM Configuration Notes” on page 14](#).



## Storing Passwords upon Check or Change

If configured to do so, Platform Services for UNIX stores the user's password in the local security system and the Samba password file upon a successful check password or change password operation.

To configure Platform Services to store passwords in the local security system, add the UPDATEPASSWORD statement to the platform configuration file. For details about the platform configuration file, see the *Platform Services Planning Guide and Reference*.

To configure Platform Services to store passwords in the Samba password file, add the UPDATESAMBA statement to the platform configuration file. For details about the platform configuration file, see the *Platform Services Planning Guide and Reference*.

For details about configuring the local security system for failover, see [“Configuring PAM for Failover” on page 17](#) and [“Configuring AIX for Failover” on page 18](#).

## Password Replication from eDirectory

If configured to do so, the UNIX Platform Receiver stores password updates from eDirectory into the local security system and the Samba password file.

To configure the core driver to send password information from eDirectory to the platform, use the Web interface to set Permit Password Replication for the Platform object. For details, see the *Core Driver Administration Guide*. 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 Driver Administration Guide*.

To configure Platform Services to store passwords in the local security system, add the UPDATEPASSWORD statement to the platform configuration file. For details about the platform configuration file, see the *Platform Services Planning Guide and Reference*.

To configure Platform Services to store passwords in the Samba password file, add the UPDATESAMBA statement to the platform configuration file. For details about the platform configuration file, see the *Platform Services Planning Guide and Reference*.

For details about configuring the local security system for failover, see [“Configuring PAM for Failover” on page 17](#) and [“Configuring AIX for Failover” on page 18](#).

## Configuring PAM for Failover

The examples in the following sections demonstrate possible PAM configurations.

The first section in the [“Solaris 2.9 Example Pam Configuration File Fragment” on page 18](#) represents the auth configuration for service-name “other” in a generic Solaris 2.9 /etc/pam.conf file. The first module prompts for user ID and password. The second module does a keylogin (if needed). The final module does authentication based on the default repository as listed in nsswitch.conf.

The second section in this example, which would replace the first section, authenticates using the Identity Manager Fan-Out driver. If the driver authentication fails, an attempt is made to authenticate the user against the local repository, using the password from the driver prompt.

The other examples show similar PAM configurations for other platforms.

## Solaris 2.9 Example Pam Configuration File Fragment

```
/etc/pam.conf

#vendor supplied
#other auth requisite pam_authtok_get.so.1
#other auth required pam_dhkeys.so.1
#other auth required pam_unix_auth.so.1

#Identity Manager Fan-Out driver variation
other auth sufficient /usr/lib/security/pam_ascauth.so.1 stats
other auth required pam_unix_auth.so.1 try_first_pass
```

## FreeBSD 4.4 Example Pam Configuration File Fragment

The following example represents a possible auth configuration for service-name “login” on a FreeBSD\* 4.4 platform. This example is designed to authenticate against the Identity Manager Fan-Out driver. If the driver authentication fails for any reason, the same user ID and password combination is used with the standard authentication module.

```
/etc/pam.conf

login auth sufficient pam_ascauth.so stats
login auth required pam_unix.so try_first_pass
```

## Red Hat 7.2 Example Pam Configuration File Fragment

```
/etc/pam.d/login

auth sufficient /lib/security/pam_ascauth.so stats debug
auth required /lib/security/pam_stack.so service=system-auth

/etc/pam.d/system-auth

auth sufficient /lib/security/pam_unix.so likeauth nullok try_first_pass
```

## HP-UX B.11.00 Example Pam Configuration File Fragment

```
/etc/pam.conf

# Authentication Management
#
OTHER auth sufficient /usr/lib/security/libpam_ascauth.1 stats
OTHER auth required /usr/lib/security/libpam_unix.1 try_first_pass
```

## Configuring AIX for Failover

If Identity Manager Fan-Out driver authentication fails, an authentication attempt is transparently made against the local authentication mechanism for users whose SYSTEM attribute in /etc/security/user evaluates to the following:

```
DCE OR DCE[UNAVAIL] AND compat
```

# Managing the UNIX Platform Services Process

The Platform Services Process provides Authentication Services and the interface for the AS Client API. It establishes and maintains connections to core drivers and provides load balancing and failover among them.

The Platform Services Process must be running if you plan to use Authentication Services on the platform.

## Starting and Stopping the Platform Services Process

Start the Platform Service Process upon system startup and stop the Platform Service Process during system shutdown.

To start the Platform Services Process, use the following command:

```
/usr/local/ASAM/bin/PlatformServices/PlatformServicesProcess/asampsp
```

To stop the Platform Services Process, use the kill command.

**NOTE:** Optional Startup/Shutdown scripts are provided in the ASAM/data/UnixStartupScripts directory.

## Platform Services Process Command Line Parameters

Option	Argument	Explanation
-a	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.
-s	None	Obtain a security certificate for the Platform and end.  This is needed only during the initial configuration process.

## Maintaining Files Used by the Platform Services Process

### The Platform Configuration File

The Platform Services Process reads the platform configuration file to locate core drivers, to determine which users are authenticated using Authentication Services, and to find other configuration information. For details about the platform configuration file, see the *Platform Services Planning Guide and Reference*.

### Log Files

The UNIX Platform Services Process writes messages to log files in the SYSLOG facility specified by the SYSLOGFACILITY statement in the platform configuration file. Log messages are documented in the *Messages Reference*.

## Managing the UNIX Platform Receiver

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 11.

The Platform Receiver must be running if you plan to use Identity Provisioning on the platform.

### Starting and Stopping the Platform Receiver

Schedule running the Platform Receiver as appropriate for the mode of operation that you have chosen for it. If you are using Persistent Mode or Polling Mode, start the Platform Receiver during system startup and stop the Platform Receiver during system shutdown. If you are using Scheduled Mode, use cron or a similar utility to run the Platform Receiver on a schedule that is appropriate for you. For information about choosing a mode of operation, see the *Platform Services Planning Guide and Reference*.

To start the Platform Receiver, use the following command:

```
/usr/local/ASAM/bin/PlatformServices/PlatformReceiver/asamrcvr
```

To stop the Platform Receiver, use the kill command.

**NOTE:** Optional Startup/Shutdown scripts are provided in the ASAM/data/UnixStartupScripts directory.

### Platform Receiver Command Line Parameters

Option	Argument	Explanation
-a	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.
-i	None	The Platform Receiver uses Polling Mode.
-c	None	The Platform Receiver uses Check Mode.
-p	None	The Platform Receiver uses Persistent Mode.
-f	None	The Platform Receiver uses Full Sync Mode.
-r	None	The Platform Receiver uses Scheduled Mode.
-s	None	Obtain a security certificate for the Platform and end.  This is needed only during the initial configuration process.

The following options determine the mode of operation for the Platform Receiver: -i, -c, -p, -f, and -r. They are mutually exclusive. If none of them is present, 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 the *Platform Services Planning Guide and Reference*.

## Maintaining Files Used by the Platform Receiver

### 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 details about the platform configuration file, see the *Platform Services Planning Guide and Reference*.

### Receiver Scripts

Receiver scripts for UNIX platforms are implemented as shell scripts. The Platform Receiver runs the Receiver scripts from ASAM/bin/PlatformServices/PlatformReceiver/scripts. The installation process stores the base scripts in subdirectories of the scripts directory. For information about Receiver scripts, see [“Receiver Scripts” on page 11](#).

### Log Files

The UNIX Platform Receiver writes messages to log files in the SYSLOG facility specified by the SYSLOGFACILITY statement in the platform configuration file. Log messages are documented in the *Messages Reference*.



# 3

## Troubleshooting Platform Services

Novell® Nsure™ 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 UNIX Platform Services Process and Platform Receiver write log messages to the UNIX SYSLOG facility.

By its very nature, the Identity Manager 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 Identity Manager Fan-Out driver performance issues, see Planning in the *Core Driver Administration Guide*.

**IMPORTANT:** Make sure you upgrade the driver, including all of your platforms, when new versions or support packs become available.

### 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.

### Debugging the UNIX Platform Services Process and Platform Receiver

To obtain debugging output for the Platform Services Process or Platform Receiver on UNIX:

- 1 Add a DEBUGLOGFILE statement or DEBUGTOSTDOUT statement to the platform configuration file.

For details about the platform configuration file, see the *Platform Services Planning Guide and Reference*.

- 2 Specify the debugging command line parameter when you start the Platform Services Process or Platform Receiver.

To obtain full debugging output, specify `-d \*` on the command line.

To obtain debugging output limited to messages exchanged with core drivers, specify the `-d dom` parameter.

## Troubleshooting Authentication Services

If a user cannot authenticate through the driver but can log in through eDirectory, ensure that the user is present in the Census and is not marked as being inactive. If the user is not present and active in the Census, review your Census Search object specifications.

Ensure that the user name and password conform to the character set and length restrictions imposed by the platform operating system.

## 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.

## 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:

- 1 From a command prompt on MVS\*, OS/400\*, UNIX, or Windows\*, enter **ping *ipaddr***, where *ipaddr* is the IP address of the remote computer.
- 2 From a NetWare® console, enter **LOAD TPING *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.

You can use other NetWare utilities, such as MONITOR, CONFIG, INETCFG, and TCPCON to examine and change other aspects of server status that pertain to networking. Refer to your NetWare documentation for further details. The *Utilities Reference*, *Basic Protocol Configuration Guide*, and *Advanced Protocol Configuration and Management Guide* provide detailed information on using these and other NetWare utilities.

## Troubleshooting Platform Services Installation

If you receive the message, “OAP001E Error in SSL configuration. Check system for entropy,” your SSL entropy configuration might be in error, or your entropy daemon might not be properly installed. For additional information about entropy, see [“Secure Sockets Layer Entropy Requirements” on page 10](#).



# Index

## Symbols

/dev/random device 10

## A

administering Platform Services 13, 14  
AIX  
    configuring password failover 18  
    system intercept 10  
Authentication Services 12

## B

bibliography 7

## C

certificate  
    management 13  
    obtaining 13  
certs directory 13  
command line parameters  
    Platform Receiver 20  
    Platform Services Process 19  
components 9  
conf PAM parameter 16  
configuring  
    AIX password failover 18  
    PAM 14  
    PAM password failover 17  
    password replication 17  
    Platform Services 13  
    Samba password replication 17  
    Samba password storing 17

## D

debug PAM parameter 16  
debugging output 23

## E

entropy daemon, installing 10  
entropy requirement for SSL 10  
example pam.conf file 15

## I

installing  
    Platform Services 9, 12  
    PRNGD entropy daemon 10

## L

log  
    Platform Receiver 11, 21  
    Platform Services Process 10, 19  
lowercase passwords 12, 17

## M

managing  
    certificates 13  
    Platform Receiver 20  
    Platform Services 13  
    Platform Services Process 19  
manuals 7

## N

network troubleshooting 24

## P

PAM 10  
    configuration 14  
    configuring password failover 17  
    module parameters 16  
pam.conf 14  
    entries 15  
    example 15  
password  
    case 12, 17  
    failover, AIX 18  
    failover, PAM 17  
    management 16  
    redirection 12, 16  
    replication 12, 17  
    storing in local security system 17  
platform certificate  
    management 13  
    obtaining 13  
platform configuration file  
    Platform Receiver 11, 21  
    Platform Services Process 10, 19  
Platform Receiver 11, 20  
    command line parameters 20  
    log 21  
    managing 20  
    platform configuration file 21  
    SSL 11  
    starting and stopping 20

- Platform Services
  - administering 14
  - installing 9
  - PAM module parameters 16
- Platform Services Process 10, 19
  - command line parameters 19
  - log 19
  - managing 19
  - platform configuration file 19
  - SSL 10
  - starting and stopping 19
- Pluggable Authentication Module
  - See PAM
- PRNGD 10
- publications 7

## **R**

- Receiver scripts 11, 21
- related publications 7
- RFC 86.0 14
- root\_nds PAM parameter 16

## **S**

- Samba 17
- Secure Sockets Layer (SSL)
  - See SSL
- SSL
  - connection security 13
  - entropy requirement 10
  - Platform Receiver 11
  - Platform Services Process 10
- stacking PAM service modules 14
- starting
  - Platform Receiver 20
  - Platform Services Process 19
- stats PAM parameter 16
- stopping
  - Platform Receiver 20
  - Platform Services Process 19
- storing passwords locally 17
- SYSLOG 10, 11
- System Intercept 10
  - PAM 14

## **T**

- troubleshooting 23
  - Authentication Services 24
  - Identity Provisioning 24
  - network issues 24
  - obtaining debugging output 23

## **U**

- uninstalling Platform Services 12
- UTF-8 11