Novell Identity Manager Fan-Out Driver

3.5.1

September 28, 2007

www.novell.com

PLATFORM SERVICES
ADMINISTRATION GUIDE FOR
LINUX* AND UNIX*

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/) for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

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/) 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).

Novell Trademarks

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

Third-Party Materials

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

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.

Contents

About This Guide

This guide provides you with the information you need to install, configure, administer, and troubleshoot Platform Services for UNIX and Linux as part of the Novell® Identity Manager Fan-Out driver.

This guide includes the following sections:

Audience

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

It also assumes you have read the *Platform Services Planning Guide and Reference* and have completed the planning phase it describes.

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) 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).

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).

For documentation about Identity Manager, see the Identity Manager 3.5.1 Documentation Web site (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).

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.

Installing Platform Services

1

The installation and setup of Novell® Identity Manager Fan-Out driver Platform Services includes tasks performed on the platform and the core driver. This section describes the installation tasks that are performed on the platform system. For details about platform configuration and administration tasks, see Chapter 2, "Configuring and Administering Platform Services," on page 15.

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 include

1.1 About Platform Services for UNIX

Platform Services for UNIX consists of four 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.
- **Platform Services Cache Daemon:** The Platform Services Cache Daemon requests provisioning events from Event Journal Services and stores the information locally in a memory cache pool. Requests by the local system for account information, such as the Fan-Out Name Services Switch, can access this information efficiently.

1.1.1 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.

1.1.2 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*.

1.1.3 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. AIX 5.3 and later also supports PAM.

1.1.4 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.

1.1.5 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.

1.1.6 The Name Service Switch

The Name Service Switch communicates with the Platform Services Cache Daemon for account information defined by the RFC 2307 Posix Profile attributes. This library module may be installed on any Linux or UNIX system for complete account redirection, providing an alternative to storing user and group accounts and passwords locally. This information is delivered from eDirectory™ and updated live through Identitiy Management event mechanisms.

1.1.7 The Platform Services Cache Daemon

The Platform Services Cache Daemon processes provisioning events received from the Event Journal Services component of the core driver. These events are stored in local memory for quick access and the cache is updated live when new events are processed. The daemon 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 Services Cache Daemon 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 daemon on system startup. For details about the daemon's operation, see the *Platform Services Planning Guide and Reference*.

The daemon 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 daemon 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*.

1.1.8 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 19.

1.2 Platform Services Installation Procedure

For 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*

1.3 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 22.

2 Stop the Platform Receiver.

   For details, see "Starting and Stopping the Platform Receiver" on page 23.

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.

6 Carefully edit PAM configuration files to remove references to the Fan-Out driver PAM shared library. On AIX, also `edit /etc/security/user` and `/usr/lib/security/methods.cfg` to remove references to FanOut driver LAM modules.

7 Remove the PAM and Platform Client shared libraries. On AIX, also remove FanOut driver LAM modules.

8 Remove `/usr/include/ascauth.h`.

Configuring and Administering Platform Services

2

After you have installed Novell® Identity Manager Fan-Out driver Platform Services, use the information in this section to begin configuration and administration.

Topics include

2.1 Platform Certificate Management

Connections between Platform Services and core drivers use Secure Sockets Layer (SSL). SSL connections are authenticated through the use of certificates.

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

To obtain a new certificate for your platform, run 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.

2.2 Platform Provisioning Models

The model you use for provisioning will depend on your situation. Models include the following:

2.2.1 Local Provisioning, Redirected Authentication

Local Provisioning, Redirected Authentication is the traditional Fan-Out driver provisioning model and uses `asamrcvr` and the supplied scripts to locally provision users and groups into the `/etc/passwd and /etc/group` UNIX files. Authentication and password change are redirected, using PAM, back to the Identity Vault. On AIX the FanOut Driver’s DCE LAM module can be used, as an alternative to PAM, to redirect authentication and password change back to the Identity Vault. It is not recommended to use PAM on versions of AIX prior to version 5.3.

Each operating system vendor has its own set of PAM configuration files, and its default PAM configuration files usually change with each operating system release. There is no generic “one size

fits all" PAM configuration that is correct for every organization. The Local Provisioning, Redirected Authentication provisioning model requires you to properly configure all PAM module types for all PAM-enabled applications needed by your organization. When using PAM on AIX, make sure that auth_type is set to PAM_AUTH in `/etc/security/login.cfg`.

On AIX, the Fan-Out Driver DCE LAM module will be used when SYSTEM and registry are set to DCE in the default stanza in `/etc/security/user` as shown in the *Platform Services Quick Start Guide for AIX*. To configure AIX 5.3 and later to use LAM instead of PAM, make sure that auth_type is set to STD_AUTH in `/etc/security/login.cfg`.

2.2.2 Local Provisioning, Local Authentication

Local Provisioning, Local Authentication uses `asamrcvr` and the supplied scripts to locally provision users and groups to the `/etc/` files and keep the local password store synchronized with the Identity Vault password. PAM is auto-configured to cause local password changes to be reflected in the Identity Vault. This option is only recommended on AIX if you have AIX version 5.3 or higher and auth_type is set to PAM_AUTH in `/etc/security/login.cfg`.

2.2.3 Name Service Switch (Account Redirection)

The Name Service Switch (Account Redirection) option uses the ascauth Name Service Switch to virtually provision users and groups without modifying `/etc/passwd` or `/etc/group`. PAM is auto-configured to cause password changes to be reflected back to the Identity Vault. User home directories must be supplied using NFS automounting or some other method of your choice. Identity Vault user and group objects must have Posix* attributes in order to be used with the Name Service Switch, and these posix attributes must be present in the subscriber channel of the Fan-Out driver. When this option is chosen, `/usr/nsswitch.conf` is also auto-configured. The Name Service Switch is not supported for Free-BSD or AIX.

2.2.4 ASCAUTH LAM Module (Account Redirection)

To support Account Redirection on AIX, a special LAM module called ASCAUTH is provided with the Fan-Out Driver. The ASCAUTH LAM Module virtually provisions users and groups without modifying any local user or group files. To use the ASCAUTH LAM Module, change the `/etc/security/user` default settings for SYSTEM and registry to ASCAUTH, and add a stanza for `/usr/lib/security/ASCAUTH` to `/usr/lib/security/methods.cfg`, as shown in the *Platform Services Quick Start Guide for AIX*. Any local users who depend on the previous default settings for SYSTEM and registry will need to have those settings added to their own stanzas in `/etc/security/user`. Password changes are reflected back to the Identity Vault. User home directories must be supplied using NFS automounting or some other method of your choice. Identity Vault user and group objects must have posix attributes to be used with the ASCAUTH LAM module, and these posix attributes must be present in the subscriber channel of the Fan-Out driver. On AIX 5.3 and later, auth_type must be set to STD_AUTH in `/etc/security/login.cfg`.

2.3 Administering Platform Services

This section includes the following topics:

2.3.1 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).

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 19.

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:

Table 2-1 *Platform Services PAM Module Parameters*

conf	Specifies where the platform configuration file is located. The default location is `/usr/local/ASAM/data/asamplat.conf`. Example: `conf=/usr/local/ASAM/data/myplat.conf`
stats	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.
debug	Causes the PAM module to write debugging records to syslog.
root_nds	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 17.

2.3.2 Name Service Configuration Notes

Identity Manager Fan-Out driver platforms may also be configured for account redirection using the Name Service Switch and the Platform Services Cache Daemon.When a service requests account information such as uidNumber, gidNumber or homeDirectory, the Name Service Switch redirects these calls to the appropriate library configured by the Name Service Switch configuration file, `/etc/nsswitch.conf`. If configured to use the Fan-Out Platform Services Cache Daemon, information is retrieved from Event Journal Services memory cache which resides on the local Linux or UNIX system.

2.3.3 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 17.

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 21 and "Configuring AIX for Failover" on page 22.

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 21 and "Configuring AIX for Failover" on page 22.

Account Redirection from eDirectory

The UNIX Platform Services Cache Daemon can be configured to store account and password updates from eDirectory into a secure local memory cache. This information can be retrieved by local system calls to provide complete account information to your Linux or UNIX system, removing the burden of storing these accounts and other sensitive information in the passwd, shadow and group files of the local `/etc` directory.To set the core driver to send Posix Account information, be sure your driver filter is configured to allow the posixAccount and posixGroup

classes and attributes to flow from eDirectory to the target application. In addition, a Universal Password policy must be configured for the target users and the ability to retrieve the Universal Password from LDAP must be enabled.

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 21 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
```

2.3.4 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

Table 2-2 *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

This involves two types of files.

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*.

2.3.5 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

Table 2-3 *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.

Option	Argument	Explanation
-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

This involves three types of files.

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*.

2.3.6 Managing the UNIX Platform Services Cache Daemon

The Platform Services Cache Daemon provides Account information for account redirection. It establishes and maintains a connection to the core driver and synchronizes Posix profile and password information from eDirectory to a local memory cache.The Platform Services Cache Daemon must be running if you plan to use Account Redirection through the Name Service Switch on the platform.

Starting and Stopping the Platform Services Cache Daemon

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

To start the Platform Services Cache Daemon, use the following command:
`/usr/local/ASAM/bin/PlatformServices/PlatformPS/asamps`

To stop the Platform Services Cache Daemon, use the kill command.

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

Platform Services Process Command Line Parameters

Table 2-4 *Platform Services Process Command Line Parameters*

Option	Argument	Explanation
-a	Configuration File Path	Specifies the platform configuration file to use.

Maintaining Files Used by the Platform Services Process

This involves three types of files.

The Platform Configuration File

The Platform Services Cache Daemon reads the platform configuration file to locate core drivers 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 Cache Daemon 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*.

Permanent Cache File

The UNIX Platform Services Cache Daemon writes the memory cache to a protected, encrypted file on the local file system in the `/usr/local/ASAM/data/PlatformServices/certs` directory. This file is written upon shutdown and read upon startup in order to provide quick retrieval of account information without having to synchronize with eDirectory upon every startup.

Troubleshooting Platform Services

3

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

The Audit and Operational logs of core driver components are maintained in their logs directory.

The 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 the planning section 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.

3.1 Obtaining Debugging Output

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

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

3.1.1 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.

3.2 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.

3.3 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 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.

3.4 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 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 z/OS*, 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.

3.5 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 9.

3.6 Troubleshooting Account Redirection

If a user cannot access the local Linux or UNIX system through the Name Service Switch and Platform Services Cache Daemon, but can log in through eDirectory, check the following:

- The user is present in the Census and platform search object.
- The user has been extended with the posixAccount auxiliary class.
- A Universal Password policy exists and is configured to allow agents to retrieve the Universal Password.
- The driver filter is configured with the posixAccount class and attributes.