

Implementation Guide

July 23, 2008

Novell® Identity Manager Driver for Linux* and UNIX* Settings

3.6

www.novell.com



Legal Notices

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

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

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

Copyright © 2006-2008 Omnibond Systems, LLC. All rights reserved. Licensed to Novell, Inc. Portions Copyright © 2006-2008 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

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

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

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

Novell Trademarks

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

Third-Party Materials

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

Contents

About This Guide	7
1 Overview	9
1.1 Basic Information Flow	9
1.2 RFC 2307 Attributes	9
2 Planning for the Linux and UNIX Settings Driver	11
3 Installing the Linux and UNIX Settings Driver	13
3.1 Installation Procedure	13
3.2 Required Knowledge and Skills	13
3.3 Prerequisites	14
3.4 Installing the Jar File	14
3.5 Installing the Import File	14
3.6 Importing the Driver	14
4 Configuring the Linux and UNIX Settings Driver	15
5 Customizing the Linux and UNIX Settings Driver	17
5.1 Customizing RFC 2307 Attributes Set by the Driver	17
5.2 Customizing UID/GID Ranges	17
5.2.1 Editing UID/GID Ranges	17
5.2.2 NxSettings Style Sheet Details	18
5.3 Accommodating Multiple Sets of RFC 2307 Attributes	19
A Troubleshooting	21
A.1 Attributes Not Added for New Users	21
A.2 Performance	22
B Technical Details	23
B.1 LUM Automation	23
B.2 Samba Automation	24
B.3 Retrieving Values with a Policy	24
C Messages	27

About This Guide

This guide describes implementation of the Identity Manager 3.6 driver for Linux and UNIX Settings, which automates the process of setting and configuring Linux and UNIX attributes in the Identity Vault of the Novell® Identity Manager driver for Linux and UNIX.

Identity Manager is the comprehensive identity management suite that allows organizations to manage the full user life cycle, from initial hire, through ongoing changes, to ultimate retirement of the user relationship.

This guide includes the following sections:

- ♦ Chapter 1, “Overview,” on page 9
- ♦ Chapter 2, “Planning for the Linux and UNIX Settings Driver,” on page 11
- ♦ Chapter 3, “Installing the Linux and UNIX Settings Driver,” on page 13
- ♦ Chapter 4, “Configuring the Linux and UNIX Settings Driver,” on page 15
- ♦ Chapter 5, “Customizing the Linux and UNIX Settings Driver,” on page 17
- ♦ Appendix A, “Troubleshooting,” on page 21
- ♦ Appendix B, “Technical Details,” on page 23
- ♦ Appendix C, “Messages,” on page 27

Audience

This guide is for system administrators and others who plan, install, configure, and use the Identity Manager 3.6 driver for Linux and UNIX Settings. It assumes that you are familiar with Identity Manager, Novell eDirectory™, and the administration of systems and platforms you connect to Identity Manager.

Feedback

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

Documentation Updates

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

Additional Documentation

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

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

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

Documentation Conventions

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

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

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

The Identity Manager driver for Linux and UNIX Settings automates the management of Linux and UNIX operational attributes, such as login shell, UID, GID, and home directory, in the Identity Vault.

This enables you to provision systems using other drivers, such as the Linux and UNIX driver or the Fan-Out driver. This also enables you to use account redirection via LDAP, Name Service Switch, or PAM modules.

You can also use the driver to automate enabling users for Linux User Management (LUM) and Novell® Samba. LUM simplifies user management in a networked environment with many Linux workstations and servers by storing all necessary properties in the Identity Vault rather than locally on each machine. Novell Samba provides Windows* access (CIFS and HTTP-WebDAV) to files stored on the OES server.

Without the Linux and UNIX Settings driver, you must use iManager to set up each user individually. The driver uses Identity Manager events and performs the same functions as the LUM and Novell Samba iManager plug-ins, but without the manual activity. For the detailed steps taken by the driver to set up users, see [Section B.1, “LUM Automation,” on page 23](#) and [Section B.2, “Samba Automation,” on page 24](#).

You must modify the Linux Workstation objects, selecting the LUM-enabled groups for the workstations to be members of, in order for users to be able to log in to OES or Novell Samba.

1.1 Basic Information Flow

1. A user is created or granted an entitlement in the Identity Vault.
2. The Linux and UNIX Settings driver receives the Create or Entitlement event and sets RFC 2307 attributes on the User objects.
3. Platform drivers, like the Linux and UNIX driver, can use the attributes set on User objects to populate users on the platforms.

1.2 RFC 2307 Attributes

The Linux and UNIX Settings driver sets the following RFC 2307 attributes:

- ♦ uidNumber
- ♦ gidNumber
- ♦ homeDirectory
- ♦ loginShell
- ♦ uid (represented as uniqueID in the Identity Vault)

You can customize the driver to set additional attributes. For details, see [Section 5.1, “Customizing RFC 2307 Attributes Set by the Driver,” on page 17](#).

The `uidNumber` and `gidNumber` attributes can be assigned using an Identity Manager Stylesheet object, or they can be assigned from a LUM Linux/UNIX Config object. The style sheet allows configuration of multiple ranges of UIDs and GIDs. The driver can skip over ranges of numbers that you do not want UIDs and GIDs allocated from. For details about configuring the UID and GID ranges in the style sheet, see [Section 5.2, “Customizing UID/GID Ranges,” on page 17](#).

Planning for the Linux and UNIX Settings Driver

2

When you install the Identity Manager driver for Linux and UNIX Settings you are prompted to supply certain information. The prompts are self-explanatory, but you should consider the following questions before installing the driver:

- ☐ What range of numbers do you want the driver to use when assigning UID numbers?
- ☐ What range of numbers do you want the driver to use when assigning GID numbers?
- ☐ Will you be using LUM?
- ☐ Will you be using Novell[®] Samba?
- ☐ Do you want the driver to assign UID and GID numbers based on the driver's Identity Manager Stylesheet object or based on the LUM Linux/UNIX Config object?

IMPORTANT: If you use the LUM Linux/UNIX Config object, do not use iManager to enable users for LUM. That could result in duplicate UID assignments.

- ☐ Do you need to import some UIDs and GIDs from platforms before running the driver? If so, how will you do the import?

Installing the Linux and UNIX Settings Driver

3

This section includes information for installing the Identity Manager driver for Linux and UNIX Settings.

Topics include

- ♦ [Section 3.1, “Installation Procedure,” on page 13](#)
- ♦ [Section 3.2, “Required Knowledge and Skills,” on page 13](#)
- ♦ [Section 3.3, “Prerequisites,” on page 14](#)
- ♦ [Section 3.4, “Installing the Jar File,” on page 14](#)
- ♦ [Section 3.5, “Installing the Import File,” on page 14](#)
- ♦ [Section 3.6, “Importing the Driver,” on page 14](#)

3.1 Installation Procedure

To install the Linux and UNIX Settings driver to a Metadirectory server, do the following:

- 1 Ensure that you have the required knowledge and skills, and that the prerequisites are met.
For details, see [Section 3.2, “Required Knowledge and Skills,” on page 13](#) and [Section 3.3, “Prerequisites,” on page 14](#).
- 2 Extend the schema to add the classes and attributes defined in RFC 2307.
For details, see the *Novell eDirectory Administration Guide*.
- 3 Install the jar file.
For details, see [Section 3.4, “Installing the Jar File,” on page 14](#).
- 4 Install the import file.
For details, see [Section 3.5, “Installing the Import File,” on page 14](#).
- 5 Import the driver.
For details, see [Section 3.6, “Importing the Driver,” on page 14](#).
- 6 Use iManager to create indexes for uidNumber and gidNumber.
For details, see the *Novell eDirectory Administration Guide*.

3.2 Required Knowledge and Skills

Successful installation of the Linux and UNIX Settings driver requires knowledge and expertise with Identity Manager and iManager. If you will be using LUM or Novell Samba, you also need knowledge and expertise with OES.

To find other documentation related to this product and its installation, see [“Additional Documentation” on page 7](#).

3.3 Prerequisites

- ❑ Novell Identity Manager 3.6
- ❑ The server that the driver runs on must hold read/write replicas of the objects the driver manages.
- ❑ To use LUM or Novell Samba integration, you must be using OES.
- ❑ For information about required systems and software, as well as supported platforms and operating environments, see [the Identity Manager 3.6 Drivers Documentation Web site \(http://www.novell.com/documentation/idm36drivers\)](http://www.novell.com/documentation/idm36drivers). From this index page, you can select a readme file associated with the platform(s) for which you need support.

3.4 Installing the Jar File

To install the jar file to the Metadirectory server, copy `NxSettings.jar` to the directory shown in the following table:

Operating System	Destination Directory
Linux and Solaris*	<code>/usr/lib/dirxml/classes</code>
Windows	<code>\novell\nds\lib</code>

3.5 Installing the Import File

To install the import file to the Metadirectory server, copy `NxSettings.xml` to the directory shown in the following table:

Operating System	Destination Directory
Linux and Solaris	<code>/usr/lib/dirxml/rules/DirXML.Drivers</code>
Windows	<code>\tomcat\webapps\nps\dirxml.drivers</code>

3.6 Importing the Driver

- 1 In iManager, click *Identity Manager Utilities*.
- 2 Click *Import Drivers* in the menu.
- 3 Specify a location to place the new driver, then click *Next*.
- 4 Select the Linux and UNIX Settings driver from the list.
- 5 Respond to the prompts.

Configuring the Linux and UNIX Settings Driver

4

You can change the configuration of the Identity Manager driver for Linux and UNIX Settings by editing its Global Configuration Variables (GCVs).

To edit GCVs:

- 1** Use iManager to navigate to the Driver Overview page.
- 2** Click the driver image to display the Driver Settings window.
- 3** Select *Global Config Values*.
With Internet Explorer, *Global Config Values* is located under the *Identity Manager* tab.
With other browsers, *Global Config Values* is located in the drop-down menu at the top.
- 4** Alter the values as desired.
- 5** Click *OK*.

Customizing the Linux and UNIX Settings Driver

5

This section provides information on customizing the operation of the Identity Manager driver for Linux and UNIX Settings. Topics include

- [Section 5.1, “Customizing RFC 2307 Attributes Set by the Driver,” on page 17](#)
- [Section 5.2, “Customizing UID/GID Ranges,” on page 17](#)
- [Section 5.3, “Accommodating Multiple Sets of RFC 2307 Attributes,” on page 19](#)

5.1 Customizing RFC 2307 Attributes Set by the Driver

The Linux and UNIX Settings driver sets the following RFC 2307 attributes:

- uidNumber
- gidNumber
- homeDirectory
- loginShell
- uid (represented as uniqueID in the Identity Vault)

You can add other attributes for the driver to set.

To add more attributes for the driver to set, use iManager to modify the creation policy.

- If you are using the driver style sheet to assign UID and GID numbers, modify the XMLDOC, Set Req Posix Attrs creation policy.
- If you are using the LUM Linux/UNIX Config object to assign UID and GID numbers, modify the LUM, Set Req Posix Attrs creation policy.

5.2 Customizing UID/GID Ranges


You can assign UID and GID numbers from a style sheet in the driver. Alternatively, you can use the LUM Linux/UNIX Config object to assign UID and GID numbers. For details about using the LUM Linux/UNIX Config object, see the *Linux User Management Technology Guide*, which is available from the [Novell® OES documentation Web site \(http://www.novell.com/documentation/oes/index.html\)](http://www.novell.com/documentation/oes/index.html).

UID and GID ranges are defined in the NxSettings style sheet. You can edit this style sheet to modify the ranges that UIDs and GIDs are allocated from.

5.2.1 Editing UID/GID Ranges

IMPORTANT: The NxSettings style sheet holds the last assigned UID and GID number. Stop the driver before you edit this document. Failure to stop the driver before editing this document could result in the same UID or GID being assigned to multiple users or groups.

To edit the NxSettings style sheet:

- 1 Open iManager and navigate to the Driver Overview page for the Linux and UNIX Settings driver.
- 2 Click the *View All Policies* icon  to display the View All Policies dialog box.
- 3 Click the link to the NxSettings Stylesheet object under the *Identity Manager Driver* heading to display the NxSettings style sheet, then click *Enable XML editing*.
- 4 Make changes as desired. For details, see [Section 5.2.2, “NxSettings Style Sheet Details,” on page 18](#).
- 5 Click *OK*.

5.2.2 NxSettings Style Sheet Details

The initial NxSettings style sheet is similar to the following example:

```
<?xml version="1.0" encoding="UTF-8"?><nxSettings>
  <settings name="DefaultSet">
    <setting name="uid" type="range">
      <ranges last-used="0">
        <range end="1000" start="400"/>
      </ranges>
    </setting>
    <setting name="gid" type="range">
      <ranges last-used="0">
        <range end="1000" start="400"/>
      </ranges>
    </setting>
  </settings>
</nxSettings>
```

You can change the start and end attributes of a range tag and the last-used attribute for the ranges tag. (Because the attributes are stored in alphabetical order, the end attribute of a range element is listed before the start attribute.)

You can add ranges by adding more range tags as shown in the following example:

```
<ranges last-used="0">
  <range end="1000" start="400"/>
  <range end="2000" start="1001"/>
  <range end="5000" start="3000"/>
</ranges>
```

Ranges do not need to be contiguous. If there are gaps, the driver skips over them. If the value of the last-used attribute falls outside of all ranges, then the next number is assigned from the range with the next start value. In the preceding example, the next number assigned is 400. The value assigned after 2000 is 3000.

You can also add more settings and setting tags to the style sheet for your own purposes. Do not reuse the setting name `DefaultSet`.

For details about retrieving values with a policy, see [Section B.3, “Retrieving Values with a Policy,” on page 24](#).

5.3 Accommodating Multiple Sets of RFC 2307 Attributes

It might be desirable to have more than one set of Linux/UNIX settings attributes on a User object so the user can have different settings on different systems. For example, a user with accounts on a Linux system and a Solaris system might need a different login shell on each system.

You can extend the schema with iManager to accommodate additional attributes. Then you can modify driver policies to set the desired attribute values. Text settings can be added to the GCVs for the driver, and range attributes can be added to the NxSettings style sheet.

Troubleshooting

A

To troubleshoot the Identity Manager driver for Linux and UNIX Settings, see the following topics:

- ♦ [Section A.1, “Attributes Not Added for New Users,” on page 21](#)
- ♦ [Section A.2, “Performance,” on page 22](#)

A.1 Attributes Not Added for New Users

If a user is added to the Identity Vault, but the driver does not add the attributes specified in the creation policy to the User object, ensure that the user meets the appropriate selection criteria.

- ♦ Use iManager to examine the driver’s GCVs under the *Event Selection* heading, and ensure that the user meets the driver’s event selection criteria.
- ♦ If the driver is configured to use Role-Based Entitlements, make sure that the user has the required entitlement.

For details about using entitlements, see the *Identity Manager 3.6 Administration Guide* on the [Identity Manager 3.6 Documentation Web site \(http://www.novell.com/documentation/idm36\)](http://www.novell.com/documentation/idm36).

You can obtain diagnostic information from a trace.

To obtain a driver trace of events related to adding a user:

- 1 Use iManager to navigate to the Driver Overview page.
- 2 Click the driver image to display the Driver Settings window.
- 3 Select *Misc*.

With Internet Explorer, *Misc* is located under the *Identity Manager* tab.

With other browsers, *Misc* is located in the drop-down menu at the top.

- 4 Set *Trace Level* to 10.
- 5 Run `ndstrace` to capture the trace information for the driver.

Because a large volume of debug information is produced by the driver, we recommend that you record the trace to a file.

- 6 Add a user.

Trace messages pertaining to the NxSettings driver are preceded by a number and `DVRS : NxSettings ST:`. Search the trace, case insensitively, for the word `error`. If you find no errors, examine the Identity Vault to see if the driver set the attributes specified in the creation policy on the User object you added.

A null pointer exception from the Metadirectory engine can be caused by a problem with the NxSettings Stylesheet object’s contents or by a lack of free UIDs or GIDs available to be allocated. If you find a null pointer exception in the trace, search the trace for related status messages from the driver. For more information about status messages from the driver, see [Appendix C, “Messages,” on page 27](#).

A.2 Performance

Ensure that indexes for the uidNumber and gidNumber attributes exist and are online.

To avoid assigning duplicate UID and GID numbers, the driver searches the respective subtrees specified during installation, or later by GCVs. For good performance, you must create indexes for uidNumber and gidNumber. For details, see [Step 6 on page 13](#).

For details about managing indexes, see the *Novell eDirectory Administration Guide*.

Technical Details

B

This section describes some technical details about operating the Identity Manager driver for Linux and UNIX Settings. Topics include

- ♦ [Section B.1, “LUM Automation,” on page 23](#)
- ♦ [Section B.2, “Samba Automation,” on page 24](#)
- ♦ [Section B.3, “Retrieving Values with a Policy,” on page 24](#)

B.1 LUM Automation

The Linux and UNIX Settings driver can automate enabling OES users for LUM. The driver performs the following steps:

1. Assigns uidNumber from the NxSettings style sheet or the Linux/UNIX Config object.
2. Adds Public rights to the following RFC 2307 attributes:

Attribute	Rights
uidNumber	Read
gidNumber	Read
loginShell	Read
homeDirectory	Read
gecos	Read
Group Membership	Read
cn	Compare

3. Adds the user to a group enabled for LUM. (The group is chosen during driver import.)
4. Sets the user’s gidNumber equal to the gidNumber of the LUM-enabled group.
5. Adds the object class uamPosixUser to the user.

If you have a very large number of users, you might not want all users to be in a single static group. You can define a dynamic group as your LUM-enabled group.

To use a dynamic group as your LUM-enabled group:

- 1 Define the dynamic group according to your needs.
- 2 Set the GCV named Add Users to LUM-Enabled Group to No.
(The GCV named LUM-Enabled Group DN is not used in this case.)

For details about LUM, see the *Linux User Management Technology Guide*, which is available from the Novell® OES documentation Web site (<http://www.novell.com/documentation/oes/index.html>).

B.2 Samba Automation

The Linux and UNIX Settings driver can automate enabling OES users for Novell Samba. The driver performs the following steps:

1. Obtains the sambaSID and sambaPrimaryGroupSID attributes.

To get the SID attributes, the driver searches the user's container for a sambaDomain object. If one is not found there, the driver searches each higher container until a sambaDomain object is found.

2. Sets the sambaSID attribute on the user object with the value from the sambaDomain object, a dash, and the value of the user's UID.

For example, if the value of the sambaSID attribute from the sambaDomain object is S-1-5-21-948752240-2140611863-4084635154 and the user's GID is 1007, the value of the user's sambaSID attribute is set to S-1-5-21-948752240-2140611863-4084635154-1007.

3. Sets the sambaPrimaryGroupSID attribute on the user object with the value from the sambaDomain object, a dash, and the value of the user's UID.

This works the same way as for the sambaSID attribute in the preceding example.

4. Adds the object class sambaSAMAccount to the user.
5. Adds the object class sambaAccount to the user.

B.3 Retrieving Values with a Policy

You can retrieve values with a policy by calling methods from the driver shim. The following example demonstrates retrieving values with XPath expressions:

```
<do-set-local-variable name="DriverDN">
  <arg-string>
    <token-text
xml:space="preserve">\MYD1\mydriver\drivers\NxSettings</token-text>
    </arg-string>
  </do-set-local-variable>
  <do-set-local-variable name="driverShimInstance">
    <arg-object>
      <token-xpath
expression="NxSettingsInstance:getInstance($DriverDN)"/>
      </arg-object>
    </do-set-local-variable>
    <do-set-local-variable name="uid">
      <arg-string>
        <token-xpath
expression='driverShim:getNextRangeValue($driverShimInstance,"DefaultSet", "uid")'/>
        </arg-string>
      </do-set-local-variable>
```

This works as follows:

1. Local variable "DriverDN" is set to the DN of the driver. This value is used to retrieve the correct instance of the driver shim from the Java* virtual machine.

2. The static `getInstance` method is called on the driver shim to retrieve a handle to the shim object.
3. The `getNextRangeValue` method is called on the driver shim with the following arguments:

Argument	Description
<code>driverShimInstance</code>	The handle to the shim
<code>setName</code>	The name of the settings set to retrieve the range value from (defined by the value of the settings tag name attribute)
<code>SettingName</code>	The name of the setting containing the range information to be used to assign an ID

The Identity Manager driver for Linux and UNIX Settings appends status elements containing message text to XDS commands as necessary to report unexpected conditions. You can view these in a trace. The driver produces the following status messages:

Error finding sambaSID object for context *userDN*

Error finding sambaSID object *searchDN*

Explanation: The sambaSID could not be determined for a user. The samba SID is stored in the sambaSID attribute of a sambaDomain object. The sambaDomain object is located by searching the user's container, and then each higher container up the tree until a sambaDomain object is located. The user's sambaSID is set to the sambaSID of the sambaDomain object concatenated with a hyphen and the value of the user's uidNumber attribute.

The sambaDomain object is created when Novell® Samba is configured on OES using YaST.

Possible Cause: A sambaDomain object does not exist in the same context as the User object or in a parent container of the User object.

Action: Create a sambaDomain object in the same context as the User object or in a parent container of the User object.

Error searching for uidInUse *searchDN*

Error searching for gidInUse *searchDN*

Explanation: The driver received an error from Identity Manager while checking to see if a uidNumber or gidNumber is already in use.

Possible Cause: The searchDN is not valid.

Action: Ensure that the searchDN is correct. You can modify the searchDN with GCVs for the driver. These are located under the *Data Integrity* heading. Inspect and modify the following settings if necessary:

- ♦ Subtree to Search for Existing UIDs to Avoid Duplicate UID Assignment
- ♦ Subtree to Search for Existing GIDs to Avoid Duplicate GID Assignment

Could not retrieve LUM object *UnixConfigDN*

Explanation: The driver could not locate the Linux/UNIX Config object used to hold UID and GID information for LUM.

The Linux/UNIX Config object is created when LUM is configured on OES using YaST.

Possible Cause: The Linux/UNIX Config object is not at the location specified by the GCV named LUM Linux/UNIX Configuration Object DN.

Action: Ensure that the GCV named LUM Linux/UNIX Configuration Object DN is set to the DN of an existing Linux/UNIX Config object.

Error finding NxSettings document

Explanation: The driver could not locate the NxSettings Stylesheet object that holds UID and GID information.

Possible Cause: The NxSettings Stylesheet object has been deleted, or an error occurred when the driver was imported.

Action: Import the driver again.

Setting set *settingSetName* not found

Explanation: The policy uses XPath calls to retrieve settings from the NxSettings Stylesheet object. A settings element name was referenced that does not exist in the NxSettings Stylesheet object.

Possible Cause: A settings element name that does not exist in the NxSettings Stylesheet object was referenced in a policy.

Action: Ensure that the policy does not reference a settings element name that does not exist.

In the following code snippet, *DefaultSet* is the settings name:

```
<token-xpath  
expression='driverShim:getNextGIDFromXMLDoc($driverShim  
Instance,"DefaultSet", "gid", $searchBase)'/>
```

The settings name *DefaultSet* is defined in the following NxSettings Stylesheet object:

```
<?xml version="1.0" encoding="UTF-8"?>  
<nxSettings>  
  <settings name="DefaultSet">  
    <setting name="uid" type="range">  
      <ranges last-used="601">  
        <range end="56500" start="600"/>  
      </ranges>  
    </setting>  
    <setting name="gid" type="range">  
      <ranges last-used="601">  
        <range end="56500" start="600"/>  
      </ranges>  
    </setting>  
  </settings>  
</nxSettings>
```

Setting *settingName* not defined in setting set *settingSetName*

Explanation: The policy uses XPath calls to retrieve settings from the NxSettings Stylesheet object. A setting element name was referenced that does not exist in the NxSettings Stylesheet object.

Possible Cause: A setting element name that does not exist in the NxSettings Stylesheet object was referenced in a policy.

Action: Ensure that the policy does not reference a setting element name that does not exist.

In the following code snippet, `gid` is the setting name:

```
<token-xpath  
expression='driverShim:getNextGIDFromXMLDoc($driverShim  
Instance,"DefaultSet", "gid", $searchBase)'/>
```

The setting name `gid` is defined in the following `NxSettings` Stylesheet object:

```
<?xml version="1.0" encoding="UTF-8"?>  
<nxSettings>  
  <settings name="DefaultSet">  
    <setting name="uid" type="range">  
      <ranges last-used="601">  
        <range end="56500" start="600"/>  
      </ranges>  
    </setting>  
    <setting name="gid" type="range">  
      <ranges last-used="601">  
        <range end="56500" start="600"/>  
      </ranges>  
    </setting>  
  </settings>  
</nxSettings>
```