



NetIQ[®] Identity Manager Linux* and UNIX* Settings Driver Implementation Guide

July, 2023

Legal Notice

THIS DOCUMENT AND THE SOFTWARE DESCRIBED IN THIS DOCUMENT ARE FURNISHED UNDER AND ARE SUBJECT TO THE TERMS OF A LICENSE AGREEMENT OR A NON-DISCLOSURE AGREEMENT. EXCEPT AS EXPRESSLY SET FORTH IN SUCH LICENSE AGREEMENT OR NON-DISCLOSURE AGREEMENT, NETIQ CORPORATION PROVIDES THIS DOCUMENT AND THE SOFTWARE DESCRIBED IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SOME STATES DO NOT ALLOW DISCLAIMERS OF EXPRESS OR IMPLIED WARRANTIES IN CERTAIN TRANSACTIONS; THEREFORE, THIS STATEMENT MAY NOT APPLY TO YOU.

For purposes of clarity, any module, adapter or other similar material ("Module") is licensed under the terms and conditions of the End User License Agreement for the applicable version of the NetIQ product or software to which it relates or interoperates with, and by accessing, copying or using a Module you agree to be bound by such terms. If you do not agree to the terms of the End User License Agreement you are not authorized to use, access or copy a Module and you must destroy all copies of the Module and contact NetIQ for further instructions.

This document and the software described in this document may not be lent, sold, or given away without the prior written permission of NetIQ Corporation and Omnibond Systems, LLC., except as otherwise permitted by law. Except as expressly set forth in such license agreement or non-disclosure agreement, no part of this document or the software described in this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, or otherwise, without the prior written consent of NetIQ Corporation and Omnibond Systems, LLC. Some companies, names, and data in this document are used for illustration purposes and may not represent real companies, individuals, or data.

This document could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein. These changes may be incorporated in new editions of this document. NetIQ Corporation and Omnibond Systems, LLC. may make improvements in or changes to the software described in this document at any time.

U.S. Government Restricted Rights: If the software and documentation are being acquired by or on behalf of the U.S. Government or by a U.S. Government prime contractor or subcontractor (at any tier), in accordance with 48 C.F.R. 227.7202-4 (for Department of Defense (DOD) acquisitions) and 48 C.F.R. 2.101 and 12.212 (for non-DOD acquisitions), the government's rights in the software and documentation, including its rights to use, modify, reproduce, release, perform, display or disclose the software or documentation, will be subject in all respects to the commercial license rights and restrictions provided in the license agreement.

© 2023 Omnibond Systems, LLC. All Rights Reserved. Licensed to NetIQ Corporation. Portions copyright © 2023 NetIQ Corporation. All Rights Reserved.

For information about NetIQ trademarks, see <https://www.netiq.com/company/legal/>.

Third-Party Materials

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

Contents

About this Book and the Library	5
About NetIQ Corporation	7
1 Overview	9
Basic Information Flow	9
RFC 2307 Attributes	9
2 Planning for the Linux and UNIX Settings Driver	11
3 Installing the Linux and UNIX Settings Driver	13
Installation Procedure	13
Required Knowledge and Skills	13
Prerequisites	13
Creating the Driver in Designer	14
Importing the Current Driver Packages	14
Installing the Driver Packages	15
Configuring the Driver	17
Deploying the Driver	17
Starting the Driver	18
4 Configuring the Linux and UNIX Settings Driver	19
5 Customizing the Linux and UNIX Settings Driver	21
Customizing RFC 2307 Attributes Set by the Driver	21
Customizing UID/GID Ranges	21
Editing UID/GID Ranges	21
NxSettings Style Sheet Details	22
Accommodating Multiple Sets of RFC 2307 Attributes	23
A Troubleshooting	25
Attributes Not Added for New Users	25
Performance	26
B Technical Details	27
LUM Automation	27
Samba Automation	28
Retrieving Values with a Policy	28
C Messages	31

About this Book and the Library

This guide describes implementation of the Identity Manager 4.8 driver for Linux and UNIX Settings, which automates the process of setting and configuring Linux and UNIX attributes in the Identity Vault of the NetIQ® 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.

Other Information in the Library

The library provides the following information resources:

Identity Manager Setup Guide

Provides overview of Identity Manager and its components. This book also provides detailed planning and installation information for Identity Manager.

Designer Administration Guide

Provides information about designing, testing, documenting, and deploying Identity Manager solutions in a highly productive environment.

User Application: Administration Guide

Describes how to administer the Identity Manager User Application.

User Application: User Guide

Describes the user interface of the Identity Manager User Application and how you can use the features it offers, including identity self-service, the Work Dashboard, role and resource management, and compliance management.

User Application: Design Guide

Describes how to use the Designer to create User Application components, including how to work with the Provisioning view, the directory abstraction layer editor, the provisioning request definition editor, the provisioning team editor, and the role catalog.

Identity Reporting Module Guide

Describes the Identity Reporting Module for Identity Manager and how you can use the features it offers, including the Reporting Module user interface and custom report definitions, as well as providing installation instructions.

Analyzer Administration Guide

Describes how to administer Analyzer for Identity Manager.

Identity Manager Common Driver Administration Guide

Provides information about administration tasks that are common to all Identity Manager drivers.

Identity Manager Driver Guides

Provides implementation information about Identity Manager drivers.

About NetIQ Corporation

We are a global, enterprise software company, with a focus on the three persistent challenges in your environment: Change, complexity and risk—and how we can help you control them.

Our Viewpoint

Adapting to change and managing complexity and risk are nothing new

In fact, of all the challenges you face, these are perhaps the most prominent variables that deny you the control you need to securely measure, monitor, and manage your physical, virtual, and cloud computing environments.

Enabling critical business services, better and faster

We believe that providing as much control as possible to IT organizations is the only way to enable timelier and cost effective delivery of services. Persistent pressures like change and complexity will only continue to increase as organizations continue to change and the technologies needed to manage them become inherently more complex.

Our Philosophy

Selling intelligent solutions, not just software

In order to provide reliable control, we first make sure we understand the real-world scenarios in which IT organizations like yours operate — day in and day out. That's the only way we can develop practical, intelligent IT solutions that successfully yield proven, measurable results. And that's so much more rewarding than simply selling software.

Driving your success is our passion

We place your success at the heart of how we do business. From product inception to deployment, we understand that you need IT solutions that work well and integrate seamlessly with your existing investments; you need ongoing support and training post-deployment; and you need someone that is truly easy to work with — for a change. Ultimately, when you succeed, we all succeed.

Our Solutions

- ♦ Identity & Access Governance
- ♦ Access Management
- ♦ Security Management
- ♦ Systems & Application Management
- ♦ Workload Management
- ♦ Service Management

Contacting Sales Support

For questions about products, pricing, and capabilities, contact your local partner. If you cannot contact your partner, contact our Sales Support team.

Worldwide:	www.netiq.com/about_netiq/officelocations.asp
United States and Canada:	1-888-323-6768
Email:	info@netiq.com
Web Site:	www.netiq.com

Contacting Technical Support

For specific product issues, contact our Technical Support team.

Worldwide:	www.netiq.com/support/contactinfo.asp
North and South America:	1-713-418-5555
Europe, Middle East, and Africa:	+353 (0) 91-782 677
Email:	support@netiq.com
Web Site:	www.netiq.com/support

Contacting Documentation Support

Our goal is to provide documentation that meets your needs. If you have suggestions for improvements, click **Add Comment** at the bottom of any page in the HTML versions of the documentation posted at www.netiq.com/documentation. You can also email Documentation-Feedback@netiq.com. We value your input and look forward to hearing from you.

Contacting the Online User Community

Qmunity, the NetIQ online community, is a collaborative network connecting you to your peers and NetIQ experts. By providing more immediate information, useful links to helpful resources, and access to NetIQ experts, Qmunity helps ensure you are mastering the knowledge you need to realize the full potential of IT investments upon which you rely. For more information, visit <http://community.netiq.com>.

1 Overview

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 NetIQ® 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. NetIQ 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 Identity Console to set up each user individually. The driver uses Identity Manager events and performs the same functions as the LUM and NetIQ Samba iManager plug-ins, but without the manual activity. For the detailed steps taken by the driver to set up users, see [“LUM Automation” on page 27](#) and [“Samba Automation” on page 28](#).

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 NetIQ Samba.

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.

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 [“Customizing RFC 2307 Attributes Set by the Driver” on page 21](#).

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 [“Customizing UID/GID Ranges” on page 21](#).

2 Planning for the Linux and UNIX Settings Driver

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 NetIQ® 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?

3 Installing the Linux and UNIX Settings Driver

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

Topics include

- ♦ [“Installation Procedure” on page 13](#)
- ♦ [“Required Knowledge and Skills” on page 13](#)
- ♦ [“Prerequisites” on page 13](#)
- ♦ [“Creating the Driver in Designer” on page 14](#)

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 [“Required Knowledge and Skills” on page 13](#) and [“Prerequisites” on page 13](#).
- 2 Extend the schema to add the classes and attributes defined in RFC 2307.
For details, see the *NetIQ eDirectory Administration Guide*.
- 3 Create the driver.
For details, see [“Creating the Driver in Designer” on page 14](#).
- 4 Use Identity Console to create indexes for uidNumber and gidNumber.
For details, see the *NetIQ eDirectory Administration Guide*.

Required Knowledge and Skills

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

Prerequisites

- NetIQ Identity Manager 4.8
- The server that the driver runs on must hold read/write replicas of the objects the driver manages.

- ❑ To use LUM or NetIQ 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 4.8 Drivers Documentation Web site \(https://www.netiq.com/documentation/identity-manager-48/\)](https://www.netiq.com/documentation/identity-manager-48/). From this index page, you can select a readme file associated with the platform(s) for which you need support.

Creating the Driver in Designer

The Linux and Unix Settings Driver supports Designer 4 Package features, which allows you to create a driver by selecting which packages to install. After you create and configure the driver, you need to deploy it to the Identity Vault and start it.

Topics in this section include

- ◆ [“Importing the Current Driver Packages” on page 14](#)
- ◆ [“Installing the Driver Packages” on page 15](#)
- ◆ [“Configuring the Driver” on page 17](#)
- ◆ [“Deploying the Driver” on page 17](#)
- ◆ [“Starting the Driver” on page 18](#)

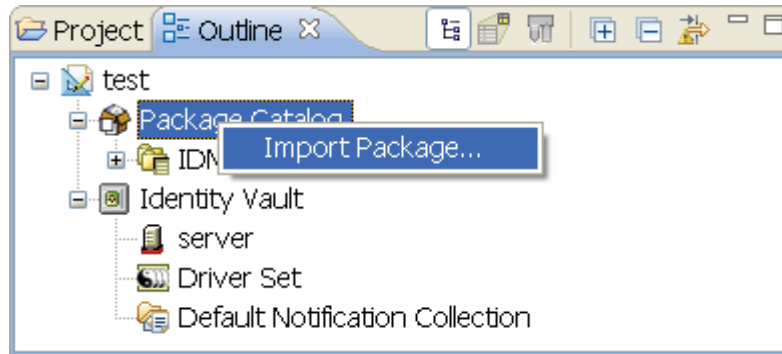
Importing the Current Driver Packages

Driver packages can be updated at any time and are stored in the Package Catalog. Packages are initially imported into the Package Catalog when you create a project, import a project, or convert a project. It is important to verify you have the latest packages imported into the Package Catalog before you install the driver.

To verify you have the latest packages imported into the Package Catalog:

- 1 Open Designer.
- 2 In the toolbar, click **Help** > **Check for Package Updates**.
- 3 Click **OK** if there are no package updates
or
Click **OK** to import the package updates.
- 4 In the Outline view, right-click the **Package Catalog**.
- 5 Click **Import Package**.

Figure 3-1 Import Package



- 6 Select the Linux and Unix Settings Packages

or

Click **Select All** to import all of the packages displayed, then click **OK**.

NOTE: By default, only the base packages are displayed. Deselect **Show Base Packages Only** to display all packages.

- 7 Click **OK** to import the selected packages, then click **OK** in the successfully imported packages message.
- 8 After the current packages are imported, continue to the next section, “[Installing the Driver Packages](#)” on page 15.

Installing the Driver Packages

After you have imported the current driver packages into the Package Catalog, you can install the driver packages to create a new driver.

- 1 In Designer, open your project.
- 2 In the Modeler, right-click the driver set where you want to create the driver, then select **New > Driver**.
- 3 Select **Linux and Unix Settings Base** from the list of base packages, then click **Next**.
- 4 Select the optional features to install for the Linux and Unix Settings driver. The options are:
 - Linux and Unix Settings Posix:** This option contains the policies needed for Posix Account and Group attributes. This option is mandatory.
 - Optional Configuration: Linux and Unix Settings Posix:** This option provides policies necessary for OES, LUM and Samba auto-assigned attributes.
 - Entitlements:** This package contains the policies needed for implementing Entitlements.
- 5 After selecting the optional packages, click **Next**.
- 6 (Conditional) If the packages you selected to install have package dependencies, you must also install them to install the selected package. Click **OK** to install the listed package dependencies.
- 7 (Conditional) If more than one type of package dependency must be installed, you are presented with these packages separately. Continue to click **OK** to install any additional package dependencies.

- 8 (Conditional) The Common Settings page is displayed only if the Common Settings package is installed as a dependency. On the Install Common Settings page, fill in the following fields:
- User Container:** Select the Identity Vault container where Linux and Unix Settings users will be added if they don't already exist in the vault. This value becomes the default for all drivers in the driver set.
- If you want a unique location for this driver, set the value for all drivers on this page. After the driver is created, change the value on the driver's Global Configuration Values page.
- Group Container:** Since the Linux and Unix Settings driver does not synchronize Group objects, this setting can be ignored.
- 9 (Conditional) If not already configured, fill in the following fields on the Common Settings Advanced Edition page, then click **Next**:
- User Application Provisioning Services URL:** specify the User Application Identity Manager Provisioning URL.
- User Application Provisioning Services Administrator:** Specify the DN of the User Application Administrator user. This user should have the rights for creating and assigning resources. For more information, see "Setting Up Administrative Accounts" in the *NetIQ Identity Manager 4.8 Common Driver Administration Guide*.
- 10 On the Install Linux and Unix Settings page, fill in the following field:
- Driver Name:** Specify a name for the driver that is unique within the driver set.
- 11 (Conditional) On the Driver Parameters page, review the default Subscriber and Publisher Options. Edit, if necessary, and click **Next**:
- 12 (Conditional) On the Entitlements Name to CSV File Mappings page, click the **Add Name to File Mapping** icon to populate the page with the entitlement configuration options. Identity Manager uses the CSV file to map Linux and Unix Settings entitlements into corresponding resources in the Identity Manager catalog.

NOTE: This page is displayed only if you installed the Entitlements package.

The information that you specify in this page is used for creating the permission catalog. Fill in the following fields, then click **Next**:

Entitlement Name: Specify a descriptive name for the entitlement to map it to the CSV file that contains the Linux and Unix Settings entitlement details.

Entitlement Name is the name of the entitlement. This parameter corresponds to the Entitlement Assignment Attribute in Linux and Unix Settings. For example, you could define an entitlement called *ParkingPass*.

Entitlement Assignment Attribute: Specify a descriptive name for the assignment attribute for an entitlement.

Entitlement Assignment Attribute holds the entitlement values in Linux and Unix Settings. For example, you could have an attribute called *Parking*.

You must add this parameter to **Field Names** in the Driver Parameters page or modify it in driver settings after creating the driver.

CSV File: Specify the location of the CSV file. This file must be located on the same server as the driver. This file contains the values for the application entitlements.

Multi-valued?: Set the value of this parameter to **True** if you want to assign resources and entitlements multiple times with different values to the same user. Otherwise, set it to **False**.


- 13 Review the summary of tasks that will be completed to create the driver, then click **Finish**.

The driver is created. You can modify the configuration settings by continuing with the next section, “[Configuring the Driver](#)” on page 17. If you don’t need to configure the driver, skip ahead to “[Deploying the Driver](#)” on page 17.

Configuring the Driver


There are many settings that can help you customize and optimize the driver. The settings are divided into categories such as Driver Configuration, Engine Control Values, and Global Configuration Values (GCVs). Although it is important for you to understand all of the settings, your first priority should be to review the Driver Parameters located on the Driver Configuration page and the Global Configuration Values. These settings must be configured properly for the driver to start and function correctly.

To access the Driver Properties page:

- 1 Open your project.
- 2 In the Modeler, right-click the driver icon  or the driver line, then select **Properties**.
- 3 Modify the driver settings as necessary.
- 4 Continue with the next section, “[Deploying the Driver](#)” on page 17.

Deploying the Driver

After a driver is created in Designer, it must be deployed into the Identity Vault:

- 1 In Designer, open your project.
- 2 In the Modeler, right-click the driver icon  or the driver line, then select **Live > Deploy**.
- 3 If you are authenticated to the Identity Vault, skip to [Step 5](#); otherwise, specify the following information:
 - Host:** Specify the IP address or DNS name of the server hosting the Identity Vault.
 - Username:** Specify the DN of the user object used to authenticate to the Identity Vault.
 - Password:** Specify the user’s password.
- 4 Click **OK**.
- 5 Read through the deployment summary, then click **Deploy**.
- 6 Read the successful message, then click **OK**.
- 7 Click **Define Security Equivalence** to assign rights to the driver.

The driver requires rights to objects within the Identity Vault. The Admin user object is most often used to supply these rights. However, you might want to create a DriversUser (for example) and assign security equivalence to that user. Whatever rights that the driver needs to have on the server, the DriversUser object must have the same security rights:

- 7a Click **Add**, then browse to and select the object with the correct rights.
- 7b Click **OK** twice.

- 8 Click **Exclude Administrative Roles** to exclude users that should not be synchronized.


You should exclude any administrative User objects (for example, Admin and DriversUser) from synchronization:

- 8a Click **Add**, then browse to and select the user object you want to exclude.
 - 8b Click **OK**.
 - 8c Repeat [Step 8a](#) and [Step 8b](#) for each object you want to exclude.
 - 8d Click **OK**.
- 9 Click **OK**.

Starting the Driver

When a driver is created, it is stopped by default. To make the driver work, you must start the driver and cause events to occur. Identity Manager is an event-driven system, so after the driver is started, it won't do anything until an event occurs.

To start the driver:

- 1 In Designer, open your project.
- 2 In the Modeler, right-click the driver icon  or the driver line, then select **Live > Start Driver**.

4 Configuring the Linux and UNIX Settings Driver

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 In Identity Console, select the Identity Manager **Drivers** module from the landing page.
- 2 Select your driver and click the **Configuration** tab.
- 3 Select **Global Config Values**.
- 4 Alter the values as desired.
- 5 Click the **Save** icon.

5 Customizing the Linux and UNIX Settings Driver

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

- ♦ [“Customizing RFC 2307 Attributes Set by the Driver” on page 21](#)
- ♦ [“Customizing UID/GID Ranges” on page 21](#)
- ♦ [“Accommodating Multiple Sets of RFC 2307 Attributes” on page 23](#)

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 the Identity Console **Drivers** module to modify the creation policy.

- ♦ If you are using the driver style sheet to assign UID and GID numbers, modify the creation policy, `NOVLNXSPOS-sub-cp-setRequiredPosixAttributes`.
- ♦ If you are using the LUM Linux/UNIX Config object to assign UID and GID numbers, modify the creation policy, `NOVLNXSLUM-sub-cp-setRequiredPosixAttributes`.

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. 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.

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 Identity Console and navigate to the **Drivers** module.
- 2 Select the NxSettings Driver.
- 3 Click the **Data Transformation and Synchronization** tab.
- 4 Under **Policies**, expand the **Driver** panel to reveal the NxSettings policy.
- 5 Make changes as desired. For details, see [“NxSettings Style Sheet Details” on page 22](#).
- 6 Click **Save**.

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 [“Retrieving Values with a Policy” on page 28](#).

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 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.

A Troubleshooting

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

- ♦ “Attributes Not Added for New Users” on page 25
- ♦ “Performance” on page 26

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 Identity Console 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 4.8 Administration Guide* on the [Identity Manager 4.8 Documentation Web site \(https://www.netiq.com/documentation/identity-manager-48/\)](https://www.netiq.com/documentation/identity-manager-48/).

You can obtain diagnostic information from a trace.

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

- 1 Use Identity Console to navigate to the **Drivers** module page.
- 2 Click the NxSettings Driver.
- 3 Select the **Log & Trace Configuration** tab.
- 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 31](#).

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 4 on page 13](#).

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

B Technical Details

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

- ♦ [“LUM Automation” on page 27](#)
- ♦ [“Samba Automation” on page 28](#)
- ♦ [“Retrieving Values with a Policy” on page 28](#)

LUM Automation

The Linux and UNIX Settings driver can automate enabling OES users for NetIQ® Linux User Management (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 global configuration value (GCV) named **Add Users to LUM-Enabled Group** to **No**.
(The GCV named LUM-Enabled Group DN is not used in this case.)

GCV values can be edited on the Driver Properties page. For more information about editing GCVs, see [Chapter 4, “Configuring the Linux and UNIX Settings Driver,” on page 19](#)

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

Samba Automation

The Linux and UNIX Settings driver can automate enabling OES users for NetIQ 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 UID 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.

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

C Messages

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 NetIQ® 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($driverShimInstance,"DefaultSet","gid",$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($driverShimInst
ance,"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>

```

