NetIQ® Identity Manager

Driver for NetIQ Privileged User Manager Implementation Guide

September 2014

Legal Notice

Contents

About this Book and the Library

The *Identity Manager Driver for Privileged User Manager Implementation Guide* explains how to install, configure, and manage the Identity Manager Driver for Privileged User Manager.

Intended Audience

This guide is intended for Privileged User Manager administrators, Identity Manager administrators, and others who implement the Identity Manager driver for Privileged User Manager.

Other Information in the Library

The library provides the following information resources:

Identity Manager Framework Installation Guide

Provides detailed planning and installation information for Identity Manager components.

Identity Manager Integrated Installation Guide

Provides integrated installation information for installing Identity Manager components.

Identity Manager Overview Guide

Provides conceptual information about Identity Manager. This book also provides an overview of the various components and many administration tasks.

Identity Manager Common Driver Administration Guide

Provides implementation information about Identity Manager drivers.

Privileged User Manager Documentation

Provides information about NetIQ Privileged User Manager.

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. The documentation for this product is available on the NetIQ Web site in HTML and PDF formats on a page that does not require you to log in. If you have suggestions for documentation improvements, click **comment on this topic** at the bottom of any page in the HTML version 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

NetIQ Communities, 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, NetIQ Communities 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 community.netiq.com.

1 Understanding the NetIQ Privileged User Manager Driver

This section contains high-level information about how the NetIQ Privileged User Manager (PUM) driver functions.

1.1 Key Terms

1.1.1 Identity Manager

NetIQ Identity Manager is a service that synchronizes data among servers in a set of connected systems by using a robust set of configurable policies. Identity Manager uses the Identity Vault to store shared information, and uses the Identity Manager engine for policy-based management of the information as it changes in the vault or connected system. Identity Manager runs on the server in which the Identity Vault and the Identity Manager engine are located.

1.1.2 Connected System

A connected system is any system that can share data with Identity Manager through a driver. PUM is a connected system.

1.1.3 Identity Vault

The Identity Vault is a persistent database powered by eDirectory. Identity Manager uses Identity Vault to hold data for synchronization with a connected system. The vault can be viewed as a private data store for Identity Manager, or as a metadirectory that holds enterprise-wide data. Data in the vault is available to any protocol supported by eDirectory, including the NetWare Core Protocol (NCP), which is the traditional protocol used by iManager, LDAP, and DSML.

Because the vault is powered by eDirectory, Identity Manager can be easily integrated into your corporate directory infrastructure by using your existing directory tree as the vault.

1.1.4 Identity Manager Engine

The Identity Manager engine is the core server that implements the event management and policies of Identity Manager. The engine runs on the Java Virtual Machine in eDirectory.

1.1.5 Privileged User Manager Driver

The Identity Manager Driver for NetIQ Privileged User Manager (PUM) lets you automate access control to privileged accounts in Windows and Unix/Linux servers. You can now utilize the self-service request and approval workflow capabilities of Identity Manager to provide self-service access to privileged accounts on servers managed by PUM.

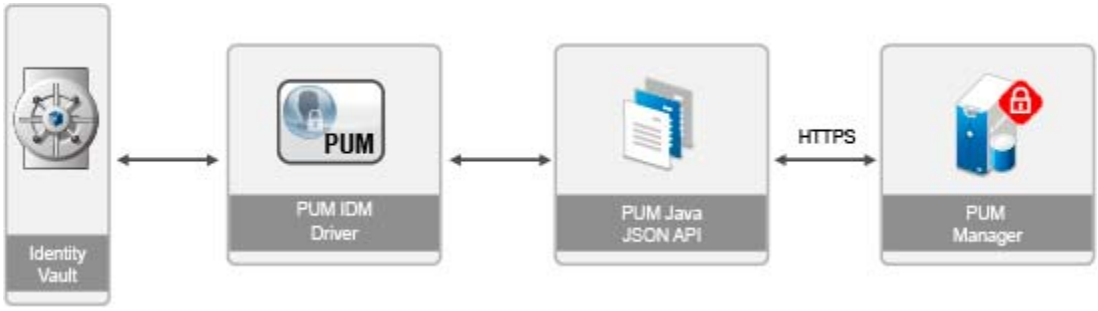
PUM helps IT administrators to provide controlled access to super-user and root accounts, allowing them to perform jobs without needlessly exposing administrative account credentials. PUM delegates privileged access to users and the access control policies are authorized via a centralized database. The PUM driver automates authorization of users into PUM based on the Identity Manager Entitlement grant.

NOTE: While the PUM driver controls privileged access to users on Windows and UNIX/Linux servers, you can use the Drivers for Linux and UNIX to create the user accounts on UNIX/Linux servers.

For more information about PUM, see the NetIQ Privileged User Manager Documentation Web site.

Figure 1-1 illustrates the data flow between Identity Manager and PUM, through the driver.

Figure 1-1 *Data Flow between Identity Manager and PUM*



1.1.6 Driver Shim

A driver shim is the component of a driver that converts the XML-based Identity Manager command and event language (XDS) to the protocols and API calls needed to interact with a connected system. The shim is called to execute commands on the connected system after the Output Transformation runs. Commands are generated on the Subscriber channel.

The PUM driver shim is implemented in Java and the name of the shim is `NPUMDriverShim.jar`. The PUM driver shim communicates with PUM through `NPUM_api.jar`. These APIs require PUM authentication to succeed.

If you use the Remote Loader, `NPUMDriverShim.jar` and `NPUM_api.jar` run on the server where the Remote Loader is running. Otherwise, it runs on the server where the Identity Manager engine is running.

The driver communicates with the PUM server via HTTPS protocol using the JSON API provided by PUM.

1.1.7 Remote Loader

A Remote Loader enables a driver shim to execute outside the Identity Manager engine (perhaps remotely on a different machine).

The Remote Loader is a service that executes the driver shim and passes information between the shim and the Identity Manager engine. When you use a Remote Loader, you install the driver shim on the server where the Remote Loader is running, not on the server where the Identity Manager engine is running. You can choose to use SSL to encrypt the connection between the Identity Manager engine and the Remote Loader. For more information, see the *Identity Manager 4.0.2 Remote Loader Guide*.

When you use the Remote Loader with the PUM driver shim, two network connections exist:

- Between the Identity Manager and the Remote Loader
- Between PUM and the PUM driver shim

1.2 Data Transfer Between Systems

Data flows between PUM and the Identity Vault by using the Subscriber channel.

The Subscriber channel performs the following:

- Watches for additions and modifications to the Identity Vault objects.
- Makes changes to PUM that reflect those changes.

You can configure the driver so that Identity Vault is allowed to update a specific attribute on PUM. In this configuration, the most recent change determines the attribute value, except for merge operations that are controlled by the filters and merge authority.

1.3 Key Driver Features

The following sections contain information about the key driver features.

- Section 1.3.1, “Local Platforms,” on page 12
- Section 1.3.2, “Remote Platforms,” on page 12

1.3.1 Local Platforms

A local installation is an installation of the driver on the Identity Manager server. The PUM driver can be installed on the Windows or Linux supported for the Identity Manager server.

For more information about local installations, see Section 2.2, “Where to Install the PUM Driver,” on page 17.

For additional information about system requirements, see “System Requirements” in the *Identity Manager 4.0.2 Framework Installation Guide*.

1.3.2 Remote Platforms

The PUM driver can use the Remote Loader service to run on a Windows or a Linux server other than the Identity Manager server.

For more information about remote installations, see Section 2.2, “Where to Install the PUM Driver,” on page 17.

For additional information about system requirements, see “System Requirements” in the *Identity Manager 4.0.2 Framework Installation Guide*.

1.3.3 Entitlements

The PUM driver supports entitlements. Entitlements make it easier to integrate Identity Manager with the Identity Manager User Application and Role-Based Services in eDirectory. In the User Application, an action such as provisioning a user to a PUM UserGroup is delayed until the proper approvals have been made. In Role-Based Services, rights assignments are made based on attributes of a user object and not by regular group membership. Both of these services offer a challenge to Identity Manager because it is not obvious from the attributes of an object whether an approval has been granted or the user matches a role.

Entitlements standardize a method of recording this information on objects in the Identity Vault. From the driver perspective, an entitlement grants or revokes the right to perform a task in PUM. You can use entitlements to control PUM UserGroup membership. The driver is unaware of the User Application. It depends on the User Application server or the Entitlements driver to grant or revoke the entitlement for a user based upon its own rules.

UserGroup: This entitlement grants or denies membership to a UserGroup in Privileged User Manager. When the entitlement is revoked, Identity Manager removes the user membership from the UserGroup.

For a new resource, the administrator must not assign the entitlement value as **Submit User** or **Everyone**.

If an administrator assigns a resource to a user in the User Application or in iManager, that change is reflected in PUM server.

The `NOVLPUMENT_x.x.x.xxxxxx.jar` package contains the Entitlement contents for PUM.

For more information about entitlements, see the *Identity Manager 4.0.2 Entitlements Guide*.

1.3.4 Password Synchronization Support

The PUM driver supports password synchronization on the Subscriber channel only. Passwords are not synchronized on the Publisher channel. You can send passwords from the Identity Vault to the connected PUM server.

Password synchronization is used to synchronize passwords of the `DirXML-PUMCredential` objects from the Identity Vault to the target PUM server. When these account objects are created in eDirectory the Identity Vault, passwords are synchronized to the target PUM servers on the Subscriber channel but, if the password is changed on the PUM driver the password is not synchronized in eDirectory.

1.3.5 Data Synchronization Support

The PUM driver synchronizes Privileged Account Domains and Credentials objects from the Identity Vault to the PUM server.

NOTE: The PUM driver does not support eDirectory synchronization for any user or user group but it supports Entitlements. For more information about Entitlements, see Section 1.3.3, “Entitlements,” on page 12

1.4 Default Driver Configuration

The PUM driver is shipped with packages. When the driver is created with packages in Designer, a set of policies and rules are created suitable for synchronizing with PUM. If your requirements for the driver are different from the default policies, you need to modify the default policies to do what you want.

1.4.1 Data Flow

The filters, mappings, and policies of PUM driver control the data flow between Identity Vault and PUM.

Filters

The driver filter determines which classes and attributes are synchronized between PUM and the Identity Vault, and in which direction synchronization takes place.

Schema Mapping

Table 1-1 and Table 1-2 show the Privileged Account Domain and Credential attributes that are mapped to PUM AccountDomain and Credential objects and attributes.

The mappings listed in the tables are default mappings. You can remap same-type attributes.

Table 1-1 *DirXML-PUMAccountDomain Class Attributes*

Identity Vault Attribute	PUM Attribute	Description
OU	name	Name of the AccountDomain. AccountDomain is a Container object, it contains the Credential objects.
DirXML-pumAccDomType	DOM_TYPE	Determines whether the AccountDomain type is SSH or LDAP.
DirXML-pumHost	DOM_HOST	DNS Hostname or IP address of the server.
DirXML-pumPort	DOM_PORT	Port on which the server is listening. Default value is 22 for SSH and 389/636 for LDAP/ LDAPS.
DirXML-pumSSHPublicKey	DOM_SSH_KEY	PublicKey of the SSH server.
DirXML-pumAccDomCredential	DOM_CREDENTIAL	Default Credential of the AccountDomain.
DirXML-pumAccDomProfile	DOM_LDAP_PROFILE	Type of AccountDomain. **NOTE:** For SSH server, it is *Generic UNIX* (value=101). For Windows server, options can be either *Windows ActiveDirectory* (value=1) or *NetIQ Directory* (value=2).
DirXML-pumAccDomSecure	DOM_LDAP_SECURE	Determines whether the LDAP AccountDomain access is over secure or non-secure channel.
DirXML-pumAccDomBaseDN	DOM_LDAP_BASEDN	LDAP baseDN of the LDAP type AccountDomain.
DirXML-pumAccDomScope	DOM_LDAP_SCOPE	LDAP scope for LDAP AccountDomain. **NOTE:** Valid values for this attribute are *one* (value=1) or *subtree* (value=2).

Table 1-2 *DirXML-PUMCredential Class Attributes*

Identity Vault Attribute	PUM Attribute	Description
uniqueID	name	Account name or ID.
nspmDistributionPassword	CRED_PASSWD	Password of the account.
DirXML-pumSSHPrivateKey	CRED_SSH_KEY	SSH privateKey of the SSH account.
DirXML-pumSSHPassPhrase	CRED_SSH_PASSPHRASE	SSH passPhrase of the SSH account.
DirXML-pumLDAPUserDN	CRED_LDAP_USERDN	UserDN of the LDAP account.

Identity Vault Attribute	PUM Attribute	Description
DirXML-pumAccDomName	CRED_DOMAIN_NAME	Name of the AccountDomain to which the Credential objects belong. The value of this attribute is set automatically by the driver based on the parent container name, which is the domain to which the Credential belongs.
DirXML-pumAccDomType	CRED_TYPE	Determines whether the credential type is SSH or LDAP. The value of this attribute is set automatically by the driver based on the parent container name, which is the domain to which the credential belongs.

NOTE: `DirXML-pumSSHPrivateKey` and `DirXML-pumSSHPassPhrase` attributes are sensitive data. You can encrypt these attributes, to ensure that the values are not visible in the trace during synchronization. For more information about attribute encryption, see "Data Encryption" in the *NetIQ eDirectory 8.8 SP8 What's New Guide*.

2 Preparing PUM

Use the information in this section as you prepare to install the PUM driver:

2.1 Driver Prerequisites

For the driver perquisites, see “System Requirements” in the *Identity Manager 4.0.2 Framework Installation Guide*.

2.2 Where to Install the PUM Driver

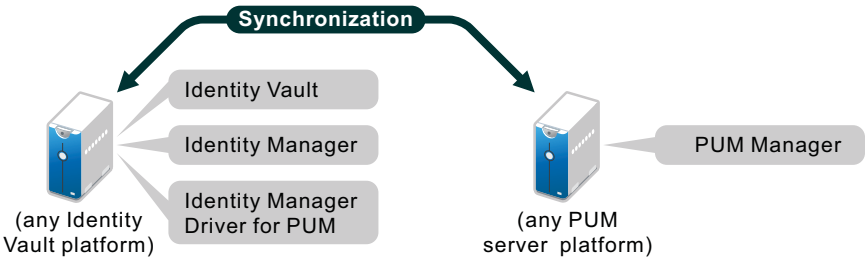
The PUM driver shim must run on one of the supported Windows or Linux platforms. However, you don’t need to install the Identity Manager engine on this same machine. Using a Remote Loader, you can separate the engine and the driver shim, allowing you to balance the load on different machines or accommodate corporate directives.

The installation scenario you select determines how the driver shim is installed. If you choose to install the driver shim on the same machine as Identity Manager (where the Identity Manager engine and the Identity Vault are located), Identity Manager calls the driver shim directly. If you choose to install the driver shim on another machine, you must use the Remote Loader.

2.2.1 Local Installation

A single Windows or Linux server can host the Identity Vault, the Identity Manager engine, and the driver and another Windows or Linux/Unix server can host PUM Manager.

Figure 2-1 *A Local Configuration*

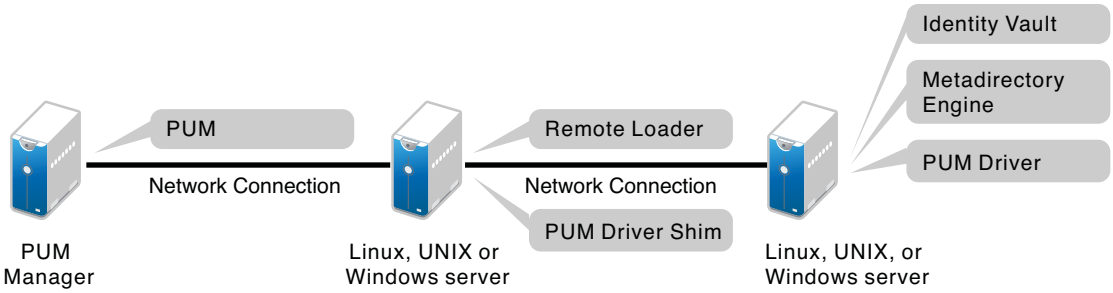


This configuration works well for organizations that want to save on hardware costs.

2.2.2 Remote Installation on Windows or Linux Platforms

If you have platform restrictions in place, you can use a three-server configuration. You can install the Remote Loader and driver shim on one, the Identity Vault and the Identity Manager engine on a second server, and PUM Manager on the third server.

Figure 2-2 *A Remote Configuration*



3 Installing the Driver Files

The PUM driver is a Java-based driver and can be run on the Identity Manager engine or on a Remote Loader server.

For installation of the Identity Manager engine and the Remote Loader (required to run the driver on a non-Identity Manager engine), see "Installing Identity Manager" in the *Identity Manager Integrated Installation Guide*.

When you install IDM 4.5, the following PUM driver files also get installed:

Files	Description
`PUM.sch`	PUM schema file. It gets extended with IDM schema.
`NPUM_api.jar` `NPUMDriverShim.jar`	Driver files
`samples`	Sample files for sample solutions described in Appendix C, "Use Cases," on page 39

The default location of the sample files is as following:

- **For Windows:** C:\NetIQ\IdentityManager\NDS\lib\dirxml\rules\npum\samples
- **For Linux:** /opt/novell/eDirectory/lib/dirxml/rules/npum/samples

4 Creating a New Driver

After the PUM driver files are installed on the server where you want to run the driver (see Chapter 3, "Installing the Driver Files," on page 19), you can create the driver in the Identity Vault. You can do so by installing the driver packages and then modifying the driver configuration to suit your environment. The following sections provide instructions:

4.1 Creating the Driver in Designer

Create the PUM driver by installing the driver packages and then modifying the configuration to suit your environment. After you create and configure the driver, you must deploy it to the Identity Vault and start it.

NOTE: Drivers are created with packages, and iManager does not support packages. To create drivers with the current version of Identity Manager, you must use Designer.

4.1.1 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 that you have the latest packages imported into the Package Catalog before you install the driver.

To verify that 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 update

   or

   Click *OK* to import the package updates. If prompted to restart Designer, click *Yes* and save your project, then wait until Designer restarts.
4. In the Outline view, right-click the Package Catalog.

5 Click *Import Package*.

6 Select the PUM packages

or

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

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

The following PUM packages are available:

- `NOVLPUMENT_x.x.x.xxxxxx.jar`
- `NOVLPUMCFG_x.x.x.xxxxxx.jar`
- `NOVLPUMBASE_x.x.x.xxxxxx.jar`
- `NOVLPUMPWD_x.x.x.xxxxxx.jar`
- `NOVLPUMMSINF_x.x.x.xxxxxx.jar`

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 with Section 4.1.2, "Installing the Driver Packages," on page 22.

4.1.2 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*.

or

Click **Enterprise** from the palette and then drag **NetIQ PUM** to the Modeler.

3 Select *PUM Base* from the list of base packages, then click *Next*.

4 Select the optional features to install for the PUM driver. All options are selected by default. The options are:

- **Default Configuration:** This package contains the default configuration information for the PUM driver. Always leave this option selected.
- **Entitlements:** This package contains policies and GCVs necessary for Entitlement support. Also, it contains GCVs to control the roles and resource mapping. With this Roles/ Resources in RBPM can be mapped to the UserGroup Entitlements and end user can be granted membership to UserGroup objects on the PUM Server.

  For general information about entitlements, see the *Identity Manager Entitlements Guide*.
- **Password Synchronization:** This packages contains the policies that enable the PUM driver to synchronize passwords. If you want to synchronize passwords, verify that this option is selected. For more information, see the *NetIQ Identity Manager 4.5 Password Management Guide (https://www.netiq.com/documentation/idm45/idm_password_management/data/front.html)*.
- **Managed System Information:** This package contains the policies that enable the driver to collect data for reports. PUM driver supports Data collection for the detailed reports. If you are using the Identity Reporting Module, verify that this option is selected. For more information refer, *Identity Reporting Module Guide*.

5 Click *Next*.

6 (Conditional) If there are package dependencies for the packages you selected to install, you must install them to install the selected package. Click *OK* to install the package dependencies listed.

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 On the Driver Information page, specify a name for the driver, then click *Next*.

9 On the Authentication Parameters page, fill in the following fields to authenticate to PUM and click *Next*:

- **Authentication ID:** Specify a PUM account with administrative privileges to be used by Identity Manager. The form of the name used depends on the selected authentication mechanism.
- **Password:** Provide the password for the specified PUM account.
- **Connection Information:** Specify the IP address/DNS name of the PUM Server.

10 On the Remote Loader page, fill in the following fields to configure the driver to connect using the Remote Loader, then click *Next*:

- **Connect to Remote Loader:** By default, the driver is configured to connect using the Remote Loader. If you want to run the driver locally, select *no*, then click *Next*. Otherwise, fill in the remaining fields to configure the driver to connect by using the Remote Loader.
- **Host Name:** Specify the hostname or IP address of the server where the driver's Remote Loader service is running.
- **Port:** Specify the port number where the Remote Loader is installed and is running for this driver. The default port number is 8090.
- **KMO:** Specify the Key Name of the Key Material Object (KMO) that contains the keys and certificate the Remote Loader uses for an SSL connection. This parameter is only used when you use SSL for connections between the Remote Loader and the Identity Manager engine.
- **Other parameters:** Specify any other parameters required to connect to the Remote Loader. Any parameters specified must use a key-value pair format, as follows:
  `paraName1=paraValue1 paraName2=paraValue2`
- **Remote Password:** Specify the Remote Loader's password as defined on the Remote Loader. The Identity Manager server (or Remote Loader shim) requires this password to authenticate to the Remote Loader
- **Driver Password:** Specify the driver object password that is defined in the Remote Loader service. The Remote Loader requires this password to authenticate to the Identity Manager server.

11 Click *Next*.

12 (Conditional) On the General Information page, fill in the following fields to define your PUM system, then click *Next*:

- **Name:** Specify a descriptive name for this PUM system. The name is displayed in reports.
- **Description:** Specify a brief description for this PUM system. The description is displayed in reports.
- **Location:** Specify the physical location of this PUM system. The location is displayed in reports.
- **Vendor:** Leave NetIQ as the vendor of PUM. This information is displayed in reports.
- **Version:** Specify the version of this PUM system. The version is displayed in the reports.

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

---

13 (Conditional) On the System Ownership page, fill in the following fields to define the ownership of the PUM system, then click *Next*:

- **Classification:** Select the classification of the PUM system. This information is displayed in the reports. The available options are:
  - Mission-Critical
  - Vital
  - Not-Critical
  - Other

  If you select *Other*, you must specify a custom classification for the PUM system.
- **Environment:** Select the type of environment the PUM system provides. The available options are:
  - Development
  - Test
  - Staging
  - Production

14 (Conditional) On the System Classification page, fill in the following fields to define the classification of the PUM system, then click *Next*:

- **Business Owner:** Select a user object in the Identity Vault that is the business owner of the PUM system. This can only be a user object, not a role, group, or container.
- **Application Owner:** Select a user object in the Identity Vault that is the application owner of the PUM system. This can only be a user object, not a role, group, or container.

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

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

The driver is now created. You can modify the configuration settings, by continuing with the next section, Section 4.1.3, "Configuring the Driver," on page 24. If you don't need to configure the driver, continue to Section 4.1.4, "Deploying the Driver," on page 25.

4.1.3 Configuring the Driver

There are some settings that 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 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 (Conditional) Click *GCVs* > *Entitlements* and review the following settings:

NOTE: These settings are only displayed if you installed the Entitlements package.

- **UserGroup Entitlement:** Ensure the value of this parameter is set to *true* to enable the driver to manage group memberships using the UserGroup entitlement. By default, the value is set to *true*.

4 Click *Apply*.

5 Modify any other settings as necessary.

In addition to the driver settings, you should review the set of default policies and rules provided by the basic driver configuration. Although these policies and rules are suitable for synchronizing with PUM, your synchronization requirements for the driver might differ from the default policies. If this is the case, you need to change them to carry out the policies you want. The default policies and rules are discussed in Section 1.4, "Default Driver Configuration," on page 13.

6 Click *OK* when finished.

7 Continue with Section 4.1.4, "Deploying the Driver," on page 25.

4.1.4 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 success 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 user account called `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*.

4.1.5 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 will not 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*.

4.2 Activating the Driver

If you created the driver in a driver set where you have already activated the Identity Manager server and service drivers, the driver inherits the activation. If you created the driver in a driver set that has not been activated, you must activate the driver within 90 days. Otherwise, the driver stops working.

For information about activation, see "Activating Novell Identity Manager Products" in the *Identity Manager 4.0.2 Integrated Installation Guide*.

4.3 Adding Packages to an Existing Driver

You can add new functionality to an existing driver by adding new packages to it.

1. Right-click the driver, then click *Properties*.
2. Click *Packages*, then upgrade the already installed PUM Base package.
   - **2a** Select the package from the list of packages, then click the *Select Operation* cell.
   - **2b** Click *Upgrade* from the drop-down list, then click *Apply*.
   - **2c** Click *OK* to close the Package Management page.

     You can upgrade the Password Synchronization package in a similar way.
3. Click the *Add Packages* icon .
4. Select the packages to install.
5. (Optional) If you want to see all available packages for the driver, clear the *Show only applicable package versions* option, if you want to see all available packages for the driver, then click *OK*.

   This option is only displayed on drivers. By default, only the packages that can be installed on the selected driver are displayed.
6. Click *Apply* to install all of the packages listed with the Install operation.
7. (Conditional) Fill in the fields with appropriate information to install the package you selected for the driver, then click *Next*.
8. Read the summary of the installation, then click *Finish*.
9. Click *OK* to close the Package Management page after you have reviewed the installed packages.
10. Modify the driver configuration settings. See Section 4.1.3, "Configuring the Driver," on page 24.
11. Deploy the driver. See Section 4.1.4, "Deploying the Driver," on page 25.
12. Start the driver. See Section 4.1.5, "Starting the Driver," on page 26.
13. Repeat Step 1 through Step 9 for each driver where you want to add the new packages.

5 Managing the Driver

As you work with the PUM driver, there are a variety of management tasks you might need to perform, including the following:

- Starting, stopping, and restarting the driver
- Viewing driver version information
- Using Named Passwords to securely store passwords associated with the driver
- Monitoring the driver's health status
- Backing up the driver
- Inspecting the driver's cache files
- Viewing the driver's statistics
- Using the DirXML Command Line utility to perform management tasks through scripts
- Securing the driver and its information
- Synchronizing objects
- Migrating and resynchronizing data
- Activating the driver
- Upgrading an existing driver

Because these tasks, as well as several others, are common to all Identity Manager drivers, they are included in one reference, the *NetIQ Identity Manager 4.0.2 Common Driver Administration Guide*.

6 Troubleshooting

6.1 Troubleshooting Driver Processes

Viewing driver processes is necessary to analyze unexpected behavior. To view the driver processing events, use DSTrace. You should only use it during testing and troubleshooting the driver. Running DSTrace while the drivers are in production increases the utilization on the Identity Manager server and can cause events to process very slowly. For more information, see “Viewing Identity Manager Processes” in the *NetIQ Identity Manager 4.0.2 Common Driver Administration Guide*.

A Driver Properties

This section provides information about the Driver Configuration and Global Configuration Values properties for the PUM driver. These are the only unique properties for drivers. All other driver properties (Named Password, Engine Control Values, Log Level, and so on) are common to all drivers. For information about the common properties, see "Driver Properties" in the *NetIQ Identity Manager 4.0.2 Common Driver Administration Guide*.

The information is presented from the viewpoint of iManager. If a field is different in Designer, it is marked with an icon.

A.1 Driver Configuration

In iManager:

1. Click to display the Identity Manager Administration page.
2. Open the driver set that contains the driver whose properties you want to edit:
   - **2a** In the *Administration* list, click *Identity Manager Overview*.
   - **2b** Click the *Driver Sets* tab.
   - **2c** If the driver set is not listed on the *Driver Sets* tab, use the *Search In* field to search for and display the driver set.
   - **2d** Click the driver set to open the Driver Set Overview page.
3. Locate the driver icon, then click the upper right corner of the driver icon to display the *Actions* menu.
4. Click *Edit properties* to display the driver's properties page.

   By default, the Driver Configuration page is displayed.

In Designer:

1. Open a project in the Modeler.
2. Right-click the driver line, then select click *Properties*.
3. Click *Driver Configuration*.

The Driver Configuration options are divided into the following sections:

A.1.1 Driver Module

The driver module changes the driver from running locally to running remotely or the reverse.

Java: Used to specify the name of the Java class that is instantiated for the shim component of the driver. This class can be located in the classes directory as a class file, or in the lib directory as a `.jar` file. If this option is selected, the driver is running locally.

The name of the Java class is `com.netiq.nds.dirxml.driver.pum.PUMDriverShim`.

Native: This option is not used in this driver.

Connect to Remote Loader: Used when the driver is connecting remotely to the connected system. Designer includes one sub-option:

- **Remote Loader client configuration for documentation:** Includes information on the Remote Loader client configuration when Designer generates documentation for the driver.

Driver Object Password: Use this option to set a password for the driver object. If you are using the Remote Loader, you must enter a password on this page or the remote driver does not run. This password is used by the Remote Loader to authenticate itself to the remote driver shim.

A.1.2 Driver Object Password (iManager Only)

Driver Object Password: Use this option to set a password for the driver object. If you are using the Remote Loader, you must enter a password on this page or the remote driver does not run. This password is used by the Remote Loader to authenticate itself to the remote driver shim.

A.1.3 Authentication

The Authentication section stores the information required to authenticate to the connected system.

Authentication ID: Specify a user application ID. This ID is used to pass Identity Vault subscription information to the application.

Example: `Admin`

Authentication context/Connection Information: Specify the IP address or name of the server the application shim should communicate with.

Example: `myserver.company.com`

Remote loader connection parameters/Remote Loader authentication: Used only if the driver is connecting to the application through the Remote Loader. The parameter to enter is `hostname=xxx.xxx.xxx.xxx port=xxxx kmo=certificatename`, when the hostname is the IP address of the application server running the Remote Loader server and the port is the port the Remote Loader is listening on. The default port for the Remote Loader is 8090.

The `kmo` entry is optional. It is only used when there is an SSL connection between the Remote Loader and the Identity Manager engine.

Example: `hostname=10.0.0.1 port=8090 kmo=IDMCertificate`

Driver Cache Limit: Specify the maximum event cache file size (in KB). If it is set to zero, the file size is unlimited. Click *Unlimited* to set the file size to unlimited in Designer.

Application password: Specify the password for the user object listed in the *Authentication ID* field.

Remote loader password: Used only if the driver is connecting to the application through the Remote Loader. The password is used to control access to the Remote Loader instance. It must be the same password specified during the configuration of the Remote Loader on the connected system.

A.1.4 Startup Option

The Startup Option section allows you to set the driver state when the Identity Manager server is started.

Auto start: The driver starts every time the Identity Manager server is started.

Manual: The driver does not start when the Identity Manager server is started. The driver must be started through Designer or iManager.

Disabled: The driver has a cache file that stores all of the events. When the driver is set to Disabled, this file is deleted and no new events are stored in the file until the driver state is changed to Manual or Auto Start.

Do not automatically synchronize the driver (Designer only): This option only applies if the driver is deployed and was previously disabled. If this is not selected, the driver re-synchronizes the next time it is started.

A.1.5 Driver Parameters

The Driver Parameters section lets you configure the driver-specific parameters. When you change driver parameters, you tune driver behavior to align with your network environment.

Publisher heartbeat interval: Allows the driver to send a periodic status message on the Publisher channel when there has been no Publisher channel traffic for the given number of minutes.

The default value is 1 minute.

A.1.6 ECMAScript (Designer Only)

Displays an ordered list of ECMAScript resource objects. The objects contain extension functions for the driver that Identity Manager loads when the driver starts. You can add additional ECMAScript objects, remove existing files, or change the order the objects are executed.

A.1.7 Global Configurations (Designer Only)

Displays an ordered list of Global Configuration objects. The objects contain extension GCV definitions for the driver that Identity Manager loads when the driver is started. You can add or remove the Global Configuration objects, and you can change the order in which the objects are executed.

A.2 Global Configuration Values

Global configuration values (GCVs) are values that can be used by the driver to control functionality. GCVs are defined on the driver or on the driver set. Driver set GCVs can be used by all drivers in the driver set. Driver GCVs can be used only by the driver on which they are defined.

The PUM driver includes several predefined GCVs. You can also add your own if you need additional ones as you implement policies in the driver.

To access the driver's GCVs in iManager:

1. Click to display the Identity Manager Administration page.
2. Open the driver set that contains the driver whose properties you want to edit:
   - **2a** In the *Administration* list, click *Identity Manager Overview*.
   - **2b** If the driver set is not listed on the *Driver Sets* tab, use the *Search In* field to search for and display the driver set.
   - **2c** Click the driver set to open the Driver Set Overview page.
3. Locate the PUM driver icon, click the upper right corner of the driver icon to display the *Actions* menu, then click *Edit Properties*.

   or

   To add a GCV to the driver set, click *Driver Set*, then click *Edit Driver Set properties*.

To access the driver's GCVs in Designer:

1. Open a project in the Modeler.
2. Right-click the PUM driver icon or line, then select *Properties* > *Global Configuration Values.*

   or

   To add a GCV to the driver set, right-click the driver set icon, then click *Properties* > *GCVs*.

The global configuration values are organized as follows:

- Section A.2.1, "Managed System Information," on page 34
- Section A.2.2, "Password Synchronization," on page 35
- Section A.2.3, "Entitlements," on page 36

A.2.1 Managed System Information

These settings help the Identity Reporting Module function to generate reports. There are different sections in the *Managed System Information* tab.

- "General Information" on page 34
- "System Ownership" on page 35
- "System Classification" on page 35

General Information

Name: Specify a descriptive name for this PUM system. This name is displayed in the reports.

Description: Specify a brief description of this PUM system. This description is displayed in the reports.

Location: Specify the physical location of this PUM system. This location is displayed in the reports.

Vendor: Select NetIQ as the vendor of the PUM system. This information is displayed in the reports.

Version: Specify the version of this PUM system. This version information is displayed in the reports.

System Ownership

Business Owner: Browse to and select the business owner in the Identity Vault for this PUM system. You must select a user object, not a role, group, or container.

Application Owner: Browse to and select the application owner in the Identity Vault for this PUM system. You must select a user object, not a role, group, or container.

System Classification

Classification: Select the classification of the PUM system. This information is displayed in the reports. The options are:

- Mission-Critical
- Vital
- Not-Critical
- Other

  If you select *Other*, you must specify a custom classification for the PUM system.

Environment: Select the type of environment the PUM system provides. The options are:

- Development
- Test
- Staging
- Production
- Other

  If you select *Other*, you must specify a custom classification for the PUM system.

A.2.2 Password Synchronization

These GCVs enable password synchronization between the Identity Vault and the PUM system.

In Designer, you must click the  icon next to a GCV to edit it. This displays the Password Synchronization Options dialog box for a better view of the relationship between the different GCVs.

In iManager, you should edit the Password Management Options on the *Server Variables* tab rather than under the GCVs. The Server Variables page has a better view of the relationship between the different GCVs.

For more information about how to use the Password Management GCVs, see "Configuring Password Flow" in the *Identity Manager 4.0.2 Password Management Guide*.

Connected System or Driver Name: Specify the name of the PUM system or the driver name. This value is used by the e-mail notification template to identify the source of the notification message.

Notify the user of password synchronization failure via e-mail: If True, notify the user by e-mail of any password synchronization failures.

A.2.3 Entitlements

There are multiple sections in the *Entitlements* tab. Depending on which packages you installed, different options are enabled or displayed.

Entitlements Configuration

For more information about entitlements, see Section 1.3.3, "Entitlements," on page 12.

UserGroup Entitlement: Select *True* to enable the driver to manage PUM UserGroups based on the driver's defined entitlements. Select False to disable management of PUM UserGroups based on the entitlements.

Parameter Format: Select the parameter format the entitlement agent must use. *Identity Manager 4* is the only supported option.

Advanced Settings: Following are the available advanced options.

Data Collection

Data collection enables the Identity Report Module to gather information to generate reports. For more information, see the *NetIQ Identity Reporting Module Guide. (https://www.netiq.com/documentation/idm45/reporting/data/bookinfo.html)*.

Enable data collection: Select *Yes* to enable data collection for the driver through the Data Collection Service by the Managed System Gateway driver. If you are not going to run reports on data collected by this driver, select *No*.

Allow data collection from UserGroups: Select *Yes* to allow data collection by the Data Collection Service through the Managed System Gateway driver for UserGroups.

Role Mapping

The Role Mapping Administrator allows you to map business roles with IT roles. For more information, see the *Novell Identity Manager Role Mapping Administrator 4.0.2 User Guide*.

Enable role mapping: Select *Yes* to make this driver visible to the Role Mapping Administrator.

Resource Mapping

The Roles Based Provisioning Module allows you to map resources to UserGroups. For more information, see the *User Application: User Guide*.

Enables resource mapping: Select *Yes* to make this driver visible to the Roles Based Provisioning Module.

B Trace Levels

The driver supports the following trace levels:

Level	Description
1	Minimal tracing
2	Previous level and some information messages
3	Previous level and error messages
4	Previous level and warning messages
5	Previous level and detailed trace messages on error and warning messages especially

NOTE: If the driver is installed locally on the Identity Manager server, the driver logs all trace messages together on the local server. However, if the driver uses the Remote Loader, the driver logs only driver shim trace messages on the Remote Loader, while the Identity Manager server logs engine trace messages.

For information about setting driver trace levels, see "Viewing Identity Manager Processes" in the *NetIQ Identity Manager Common Driver Administration Guide*.

C Use Cases

C.1 Access Control Using PUM UserGroup as IDM Entitlements

A PUM UserGroup (UG) defines a users' membership who get's privileged access on the servers. The PUM UserGroup object can be associated with a PUM Rule object to define a user’s privileged access to servers based on the UserGroup membership. In the PUM driver, UserGroup is defined as the Entitlement object. From the IDM RBPM(UserApp), the UserApp administrator can query the PUM Server via with driver and get the lists of UserGroups defined on the PUM system. On the UserApp, the administrator can create Roles/Resources and these can be associated with any of the queried UserGroup entitlements. These Roles and Resources can be associated with an IDM WorkFlow for the approval process. Any UserApp user can request for the created Roles/Resources and when they are granted, the PUM driver updates the UserGroup membership with the user's ID on the PUM server.

In addition, on the PUM server along with UserGroup object, other PUM objects such as HostGroup, AccessTime, Command can also be associated with the PUM Rule object to define more specific access to servers based on the requirement.

The user request and approval process flow are as depicted in the following diagrams:

Figure C-1 *Access Provisioning to UNIX/Linux Servers (SSH) Using PUM and PUM Driver*

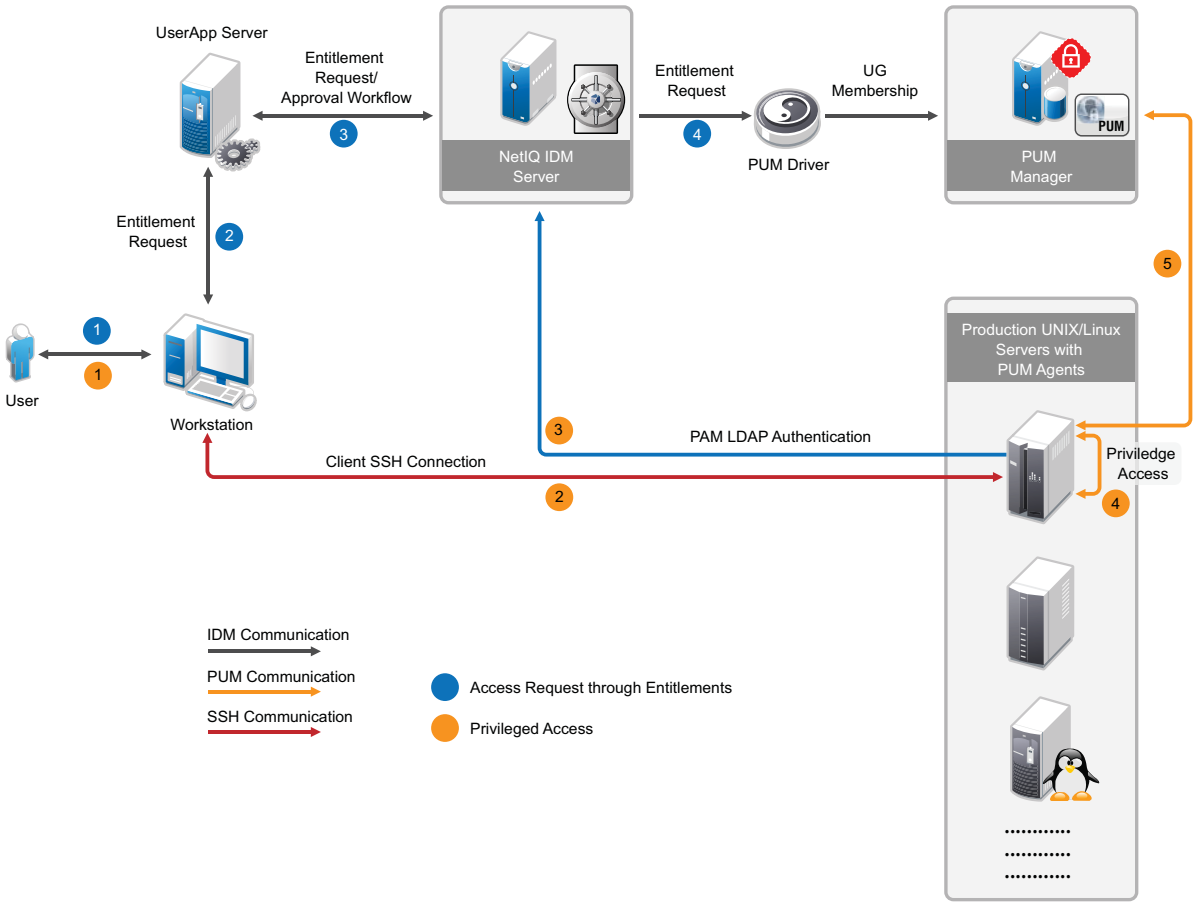
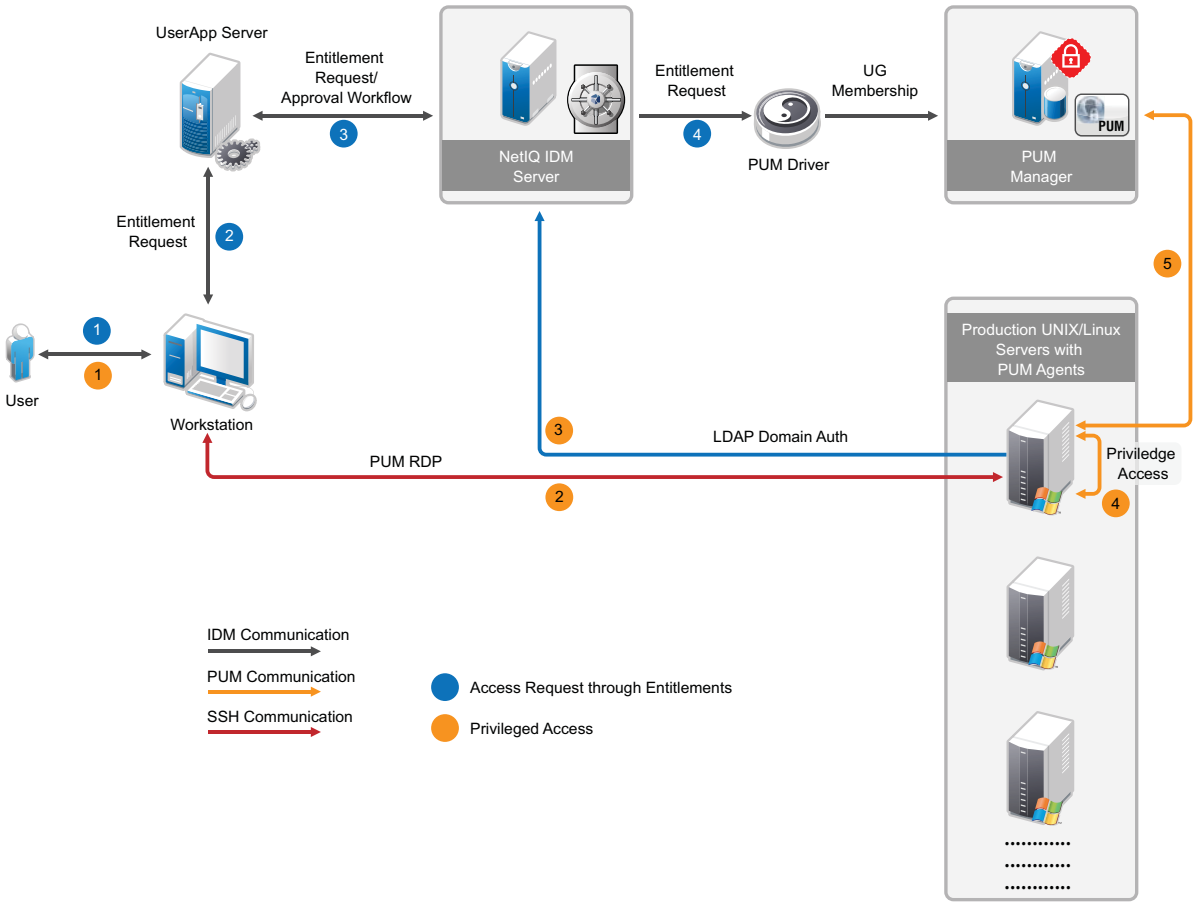
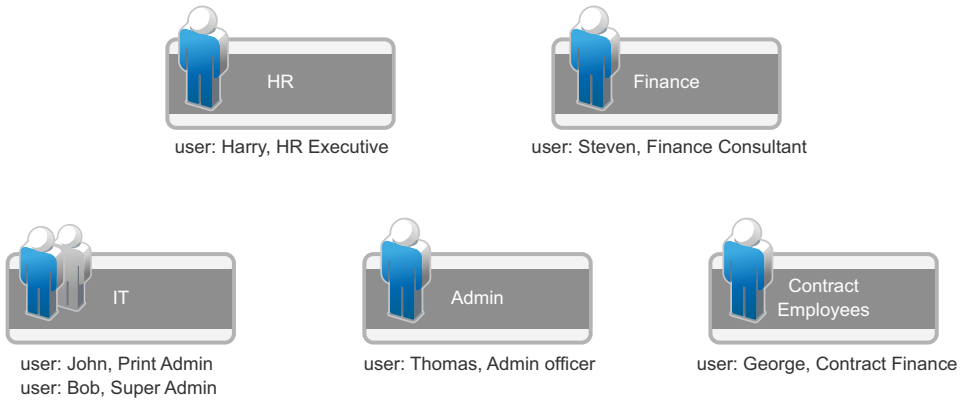


Figure C-2 *Access Provisioning to Windows Servers (RDP) Using PUM and PUM Driver*



Let us take a simple organization structure and see how to define PUM objects and provide access control to various servers used in the organization. let us assume that the organization has various departments, as shown in the following diagram:

Figure C-3 *Organization Structure - Example*



The details of the steps are as described in the following example.

- Section C.1.1, “Setting up PUM and IDM,” on page 42
- Section C.1.2, “Creating the PUM Driver using Designer,” on page 42

C.1.1 Setting up PUM and IDM

To set up PUM and IDM:

1. Install IDM 4.5, iManager, and Designer. For more information, see the NetIQ Identity Manager Documentation Web site.
2. Install PUM Framework Manager on a SLES machine.

   For example: `https://<pumManagerDNSorIP>`

   For more information, see the NetIQ Privileged User Manager 2.4.1 Documentation Web site.
3. Install PUM Agents on the UNIX/Linux and Windows servers to which privileged access are to be provided to the users. For more information, see the NetIQ Privileged User Manager 2.4.1 Documentation Web site.
4. Upload sample user objects using the `sample-users.ldif` file.

C.1.2 Creating the PUM Driver using Designer

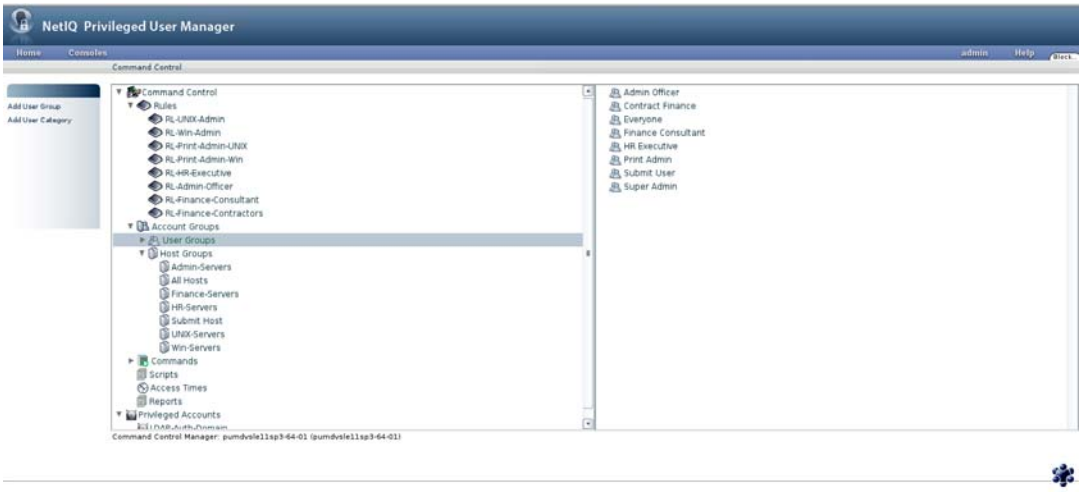
To create the PUM driver using Designer, see Chapter 4, “Creating a New Driver,” on page 21.

C.1.3 Configure PUM

Log in to PUM at `https://<pumManagerDNSorIP>` as admin and perform the following procedure:

1. Configure PUM to authenticate users from LDAP server (eDirectory):
   - **1a** Go to **Home** > **Command Control** > **Privileged Accounts**, click **Add Account Domain**, and provide values for the following fields, as specified:
     - Name: LDAP-Auth-Domain
     - Type: LDAP
     - Profile: NetIQ eDirectory
     - LDAP URL: <IP of the eDirectory server where IDM is running>
     - Base DN: ou=users,o=data
     - Account: admin
     - User DN: cn=admin,ou=sa,o=system
     - Password: <eDirectory admin password>
   - **1b** Go to **Home** > **Framework User Manager** > **Users** > **Account Settings**. In the *Authentication Domain* drop-down list, select **LDAP-Auth-Domain**. Click **Finish**.
2. A sample PUM Configuration to provide the access control to the servers in the various departments is provided in the `samples` folder. Import the sample PUM configuration file, `npumExportSettings-Entitlements.xml`, from the samples/ folder:
   - **2a** Open the `npumExportSettings-Entitlements.xml` file in a text editor and copy it to the clipboard.
   - **2b** Go to **Home** > **Command Control** > **Import Settings** and paste it in the *Import text* field. Click **Finish**.

2c After the import, you can find various PUM objects, such as Rule objects, UserGroup objects, and HostGroup objects, as shown in the following figure:



The following PUM objects are created:

- HostGroup objects: These objects defines various groups of hosts based on the department. For example the HostGroup 'HR-Servers' would contain the servers belonging to the employees in the HR department and only they should be getting access to those servers.
- UserGroup objects: These objects represents the groups to which user can get membership. For example, the UserGroup 'Admin Officer' would contain the members of the Administration department.
- Commands objects: These objects defines the privileged commands that a user can get to run. For example, in the sample there is command called 'printerCommands' which has sample list of printer commands.
- AccessTime objects: These objects define privileged access time duration. For example, in the sample there is an object 'AT-Contract' with allowed timings from 8am to 5pm on days except, Saturday and Sunday.
- Rule objects: These are the objects where all of the above mentioned objects are used to define the access policies. For example, the object 'RL-Finance-Contractors' defines that any user having the membership to the UserGroup 'Contract Finance', the server that is being accessed is in the HostGroup 'Finance-Servers' and the time of access is as per the timings defined in the AccessTime 'AT-Contract', then provide the user with access to the privileged account, 'fin-contract-user' on the servers belonging to the Finance department.

NOTE

- For providing privileged access to Windows server, AccountDomain objects and the corresponding Credential objects should be created under the PrivilegedAccounts container in the Command Control. For more information, see "Adding an Account Domain" in the *NetIQ Privileged User Manager 2.4.1 Administration Guide*.

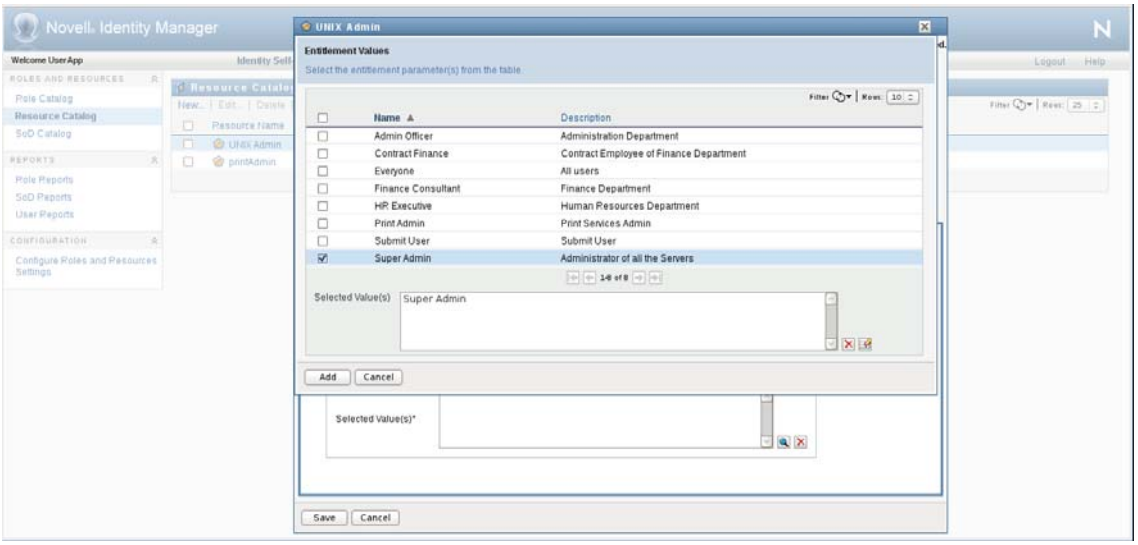
- User accounts must be created on the Production servers (see Figure C-1 and Figure C-2). The account name must be consistent with the user requesting for the Roles/Resources through RBPM/UserApp. Drivers for Linux and UNIX can be used to create local accounts on the UNIX/Linux servers.

  To avoid creation of user accounts on every server, you can configure Windows servers with LDAP Domain authentication and UNIX/Linux servers with PAM-LDAP authentication.

C.1.4 Creating Roles/Resources in UserApp

1. Log in as uaadmin (UserApp admin) to the UserApp URL: `http://<user_app_ip>:8180/IDMProv/`.
2. Import the PUM Entitlements. PUM UserGroup objects are defined as the IDM Entitlements objects for the PUM driver.

   Go to **Roles and Resources** > **Configure Roles and Resources Settings** > **Entitlement Query Settings** and click **Refresh**. This queries the UserGroup objects from the PUM server via the PUM driver.
3. Create role/resource objects in the Role/Resource Catalog and associate them with PUM UserGroup Entitlement. All the UserGroups that were queried from the PUM server are listed for entitlement selection, as shown in the following figure.



C.1.5 Getting Privileged Access

1. An UserApp user can now log in to the UserApp URL and request for the roles/resources. When the role/resource is approved, the PUM driver adds the respective user as the member of the corresponding UserGroup.

   Now, the user can get the privileged access to the servers. For example, if user **bob** gets membership to 'Super Admin' UserGroup, then he has root access to the UNIX servers, lnx-finance-server-01.mycompany.com, solaris-it-webserver.mycompany.com, and hp-it-webserver.mycompany.com. For example:

   ```
   # ssh bob@solaris-it-webserver.mycompany.com
   # id
   ```

   This will return `uid` as bob.

```
# usrun su
```

this will return `uid` as root.

Also, as 'Super Admin', bob has Administrator access to the Windows servers which he can access from PUM RDP relay page by performing the following procedure:

1a Open the following URL in the Internet Explorer:

```
https://<pum_manager_ip>/rdprelay
```

Login as bob, with password is bob123, as specified in the sample ldif file.

1b After successful login, bob can view the servers to which he has access. Click any server to start a Remote Desktop session with Administrator privileges. As per the use case, bob will have access to the Windows servers - `win2k8-hr-server-01.mycompany.com` and `win2k8-admin-server-01.mycompany.com`.

2 Either Admin can revoke the role/resource assignment or the user can delete the assigned role/ eesource. This triggers the Role/Resource Revoke process and the PUM driver triggers the removal of the user's UserGroup membership.

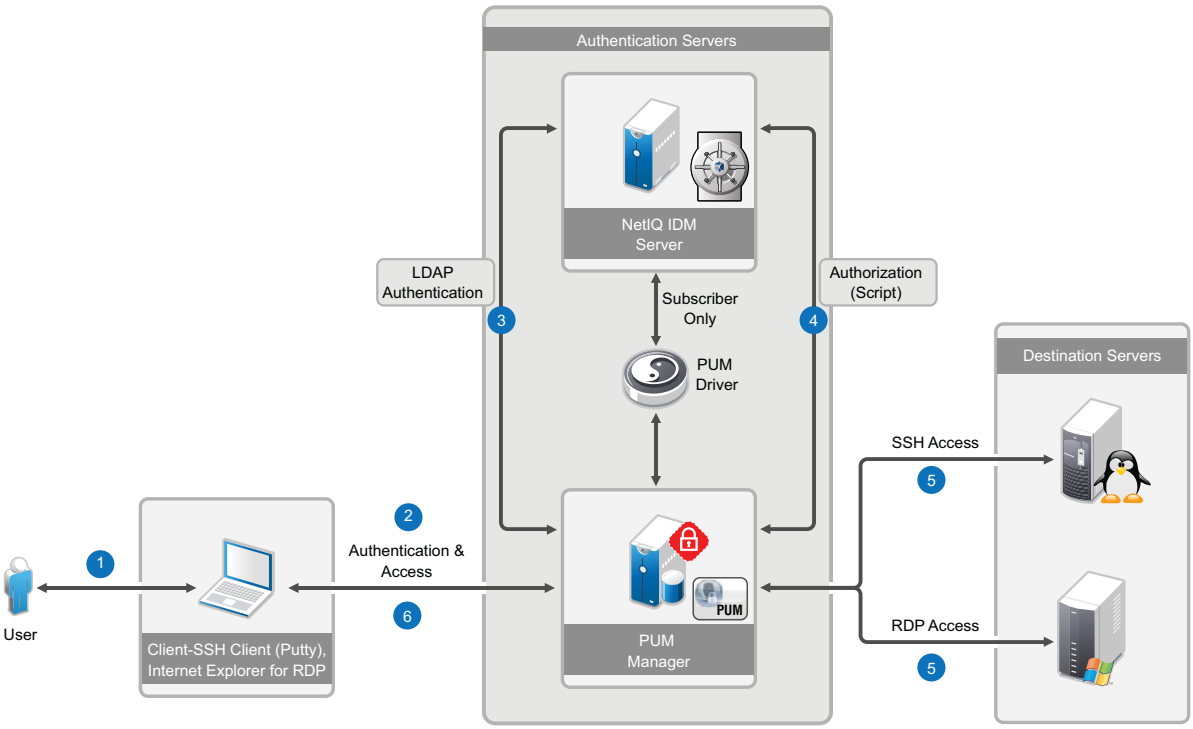
C.2 Access Control Using SSH Relay and RDP Relay Features of PUM

This solution is based on the Credential Vault feature of PUM. With the Credential Vault, a PUM administrator can create Account Domains (servers) for UNIX SSH servers and corresponding accounts called credentials for those servers under the respective domains. Similarly for Windows servers, account domain and the corresponding accounts can be created as credentials objects.

These credentials are consumed in the Rule Objects in PUM to provide the elevated access to users.

To integrate PUM with IDM, the corresponding Credential Vaults eDirectory objects can be created and managed from IDM and these would be synced to PUM by the PUM driver. Since PUM supports user authentication from an LDAP directory, an IDM user can become a PUM user by configuring PUM users to be authenticated to the eDirectory server and the access authorization to the servers would be made by PUM. eDirectory groups are created which defines the users' group membership and the group class's schema is extended with another attribute to add accounts of the hosts as members to it. By doing this, the user's group membership defines the users' privileged access to various hosts i.e. UNIX and Windows servers. So basically, for authorizing a user with privileged access to the servers, a PUM Script object which is associated with a PUM Rule object, queries the eDirectory group objects for the user's and host-account membership.

Figure C-4 *Access Control Using PUM in Relay/Proxy Mode*



The details of the steps are as described in the following example.

- Section C.2.1, “Setting up IDM,” on page 46
- Section C.2.2, “Setting up PUM,” on page 47
- Section C.2.3, “Creating the PUM Driver Using Designer,” on page 47
- Section C.2.4, “Adding eDirectory Objects Using the Sample LDIF File,” on page 47
- Section C.2.5, “Configuring PUM and the PUM Sample Export File,” on page 47
- Section C.2.6, “Getting Privileged Access,” on page 48

C.2.1 Setting up IDM

To set up IDM:

1. Install IDM 4.5, iManager, and Designer. For more information, see the NetIQ Identity Manager Documentation Web site.
2. Extend eDirectory schema (`customSchema.sch`) using iManager.
3. Ensure that a password policy is associated with the user and the servers containers for synchronization of passwords to PUM. In this use case, the users container is `ou=users,o=data` and the servers container is `ou=PUM,ou=users,o=data`. Note that, with IDM 4.5 installation, the container `ou=users,o=data` is associated with the Default Password policy, which is utilized in this use case.

C.2.2 Setting up PUM

To set up PUM:

1. Install PUM v2.4, or later Framework Manager on a Linux machine.

   For example, `https://<pumManagerDNSorIP>`.

   For more information, see the NetIQ Privileged User Manager 2.4.1 Documentation Web site.
2. Have a Windows server (example: `windows-server-01`) and a Linux server (example: `linux-server-01`), which are the destination servers to which privileged access would be provided to IDM/eDirectory users.

C.2.3 Creating the PUM Driver Using Designer

To create the PUM driver using Designer, see Chapter 4, "Creating a New Driver," on page 21.

C.2.4 Adding eDirectory Objects Using the Sample LDIF File

After the objects are added to eDirectory, corresponding PUM Credential Vault objects will be created on the PUM Server, which can be verified from the PUM UI.

To add eDirectory objects using the provided sample LDIF file:

1. Sample PUM objects are provided in the `sample-npumRelay.ldif` file.

   Modify the attributes of the sample objects in the LDIF file, such as IP addresses of the ldap account domains, user passwords, and so on. Note that the server names should be resolvable if DNS names are used.
2. Upload the user objects using the `sample-users.ldif` file.
3. Upload the eDirectory objects using the `sample-npumRelay.ldif` file.
4. After the objects are uploaded successfully, they will be synced to PUM via the PUM driver. You can see this, by logging in to the PUM Framework Manager (`https://<pumManagerDNSorIP>`) as admin, in the **Home** > **Command Control** screen.

The following objects are created:

- eDirectory User objects: john, bob
- eDirectory Group objects: unixAdminGrp, winAdminGrp
- PUM Credential Vault objects: PUM-IDM with credential admin, linux-server-01 with root account, windows-server-01 with Administrator account

C.2.5 Configuring PUM and the PUM Sample Export File

1. Configure PUM Framework Manager to authenticate users from LDAP server (eDirectory) on which IDM is running:
   - **1a** Log in to PUM (`https://<pumManagerDNSorIP>`) as admin.
   - **1b** Configure PUM to authenticate users from LDAP server(eDirectory).
     - **1b1** Go to **Home** > **Command Control** > **Privileged Accounts** and click **Add Account Domain** and provide the following information:
       - Name: LDAP-Auth-Domain
       - Type: LDAP

- Profile: NetIQ eDirectory
- LDAP URL: <IP of the eDirectory server where IDM is running>
- Base DN: ou=users,o=data
- Account: admin
- User DN: cn=admin,ou=sa,o=system
- Password: <eDirectory admin password>

1b2 Go to **Home** > **Framework User Manager**. Click **Users** > **Account Settings**. In *Authentication Domain* drop-down list, select **PUM-IDM** and click **Finish**.

2 Import the custom scripts and modify them:

2a Open the `npumExportSettings-Relay.xml` file in a text editor and copy it to the clipboard.

2b Go to **Home** > **Command Control** > **Import Settings**, and paste it in the **Import text** field. Click **Finish**.

2c After the import, two PUM script objects, `SSH-Relay-Script` and `RDP-Relay-Script`, appear under **Home** > **Command Control** > **Scripts** and two PUM Rule objects, `SSH-Relay-Script` and `RDP-Relay-Script`, appear under **Home** > **Command Control** > **Rules**.

2d Double-click each script to modify the LDAP (eDirectory) information, such as server IP, adminDN, admin credentials, and baseDN. LDAP information is located in the section of the scripts called `### CUSTOMIZE SECTION ####`. Following is the snippet from the script:

```
### CUSTOMIZE SECTION ####
my $ldap_url = "ldaps://<idm_server_ip>";
my $ldap_user = "<admin_dn>";
my $ldap_pwd = "<admin_password>";
my $ldap_user_base = "<user_container_dn>";
my $ldap_host_base = "<unix/windows_servers_container_dn>";
my $ldap_acc_grp_base = "<groups_container_dn>";
my $driver_name = "<pum_driver_dn>";
```

NOTE: The custom script expects the Windows and UNIX servers containers to be different, that is, the value of the variable `$ldap_host_base`.

C.2.6 Getting Privileged Access

NOTE

- The values are based on the sample file.
- `RDP-Relay-Rule` gets executed for RDP Relay. In this rule, **Run User** is set as *Everyone* and **Run Host** is set as *All Hosts*. With these settings, in the RDP relay page, after the user successfully logs in, the user is prompted for account and the IP/DNS address of the server to which user wants the privileged access. This behavior is different from the usual RDP relay, where the Windows server access are listed based on the privileges defined by the rules.

Verify that users **john** and **bob** are provisioned with privileged access to the servers:

1 **SSH Relay**: **bob** is a member of the `unixAdminGrp` group, and gets privileged access to all the UNIX servers that are part of this group. In this case, root account on `linux-server-01` is a member this group, so **bob** can get root access to this server via PUM by running the below command:

```
# ssh -t -p 2222 bob@<pumServerIP> root@linux-server-01
```

Enter the eDirectory password of **bob**, which is `bob123` in our example.

After successful login, **bob** gets a root access shell to `linux-server-01`. Note that **bob** did not need to provide the root password of the Linux server.

bob enters his eDirectory credentials and gets authenticated to the PUM server, which in turn authorizes **bob** with root access to the UNIX server, `linux-server-01`, without asking for root password.

2 **RDP Relay**: **john** is a member of the `winAdminGrp` group and gets privileged access to all the Windows servers that are part of this group. In this case, Administrator account on `windows-server-01` is member of this group, so **john** can get Administrator privileges to this server via PUM by performing the following steps:

2a Open the following URL in Internet Explorer 9 or above:

`https://<pumServerIP>/rdprelay`

2b Log in as **john** with password `john123`. **john** is authenticated to PUM using eDirectory credentials.

2c After successful login, **john** will be prompted to specify values in two fields, **User Name** and **Hostname/IP**. Enter the account name and the IP address of the Windows server to which you have privileged access. In this example, user name is **Administrator** and host name is be `windows-server-01`.

After successful login, **john** gets Remote Desktop access to `windows-server-01`. Note that **john** did not need to provide the Administrator password of the Windows server.

D Known Issues

D.1 Cannot Modify the DirXML-pumAccDomType Attribute

Issue: When you are modifying the objects in the `DirXML-PUMAccountDomain` object class, changing the value of the `DirXML-pumAccDomType` attribute from “ldap” to “ssh” and vice-versa does not work.

Workaround: Delete that `DirXML-pumAccDomType` attribute and create the same attribute with the value you want.

D.2 When Adding a New Account Domain, the DirXML-pumAccDomType Attribute is Disabled

Issue: When you create an account domain using the `DirXML-PUMAccountDomain` object class, and if you set the value of the `DirXML-pumAccDomType` attribute as “ldap”, the `DirXML-PUMAccDomSecure` attribute is disabled.

Workaround: You can log in to the PUM console and change the value of the `DirXML-PUMAccDomSecure` attribute.