

Identity Manager 4.0.2 Staging Guide

Identity Manager™ 4.0.2

January 2014

Legal Notices

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

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

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

Copyright © 2013 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

Novell, Inc.
1800 South Novell Place
Provo, UT 84606
U.S.A.
www.novell.com

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

Novell Trademarks

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

Third-Party Materials

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

Contents

About This Guide	5
1 Staging Identity Manager Projects	7
1.1 Understanding Staging	7
1.2 Scope of Staging Documentation	8
1.3 Staging Use Cases	8
1.4 Using Designer for Staging Identity Manager Projects	9
1.5 Moving Authorizations During Staging	9
2 Preparing for Staging	11
2.1 Converting Configuration Files to Packages	11
2.2 Prerequisites	11
2.3 Identity Vault Structure	12
2.4 Driver Configuration	12
2.4.1 Using GCVs in Policies	12
2.4.2 Simulation and Staging	12
2.5 Identity Manager Objects	13
2.5.1 Objects That Designer Models	13
2.5.2 Objects That Designer Does Not Model	14
2.5.3 Importing Objects	15
2.5.4 Exporting LDIF Container Objects to an LDIF File	17
2.5.5 Importing Objects from an LDIF File into an LDIF Container	17
2.5.6 Deploying Additional Objects into eDirectory	17
2.5.7 Editing the LDIF Container Data by Using an Editor	17
2.5.8 Deleting the LDIF Container	18
2.6 Rights	18
2.6.1 Driver Equivalences	18
2.6.2 Roles Based Entitlements Policies	18
2.6.3 Jobs	19
3 Staging Identity Manager Projects	21
3.1 Staging a Project for the First Time	21
3.1.1 Staging Using Packages	21
3.1.2 Staging Using Configuration Files	21
3.1.3 First-Time Staging Process	22
3.2 Staging Changes in an Existing Project	25
3.3 Copying Java Class .jar Files Between Stages	26
3.4 Post-Staging Tasks	26
3.5 Changing the LDAP Properties	28
4 Staging Best Practices	29

About This Guide

Welcome to *Novell Identity Manager 4.0.2 Staging Guide*. This guide provides step-by-step procedures to move your Identity Management solutions from one stage to subsequent stages.

This guide introduces the following:

- ♦ Chapter 1, “Staging Identity Manager Projects,” on page 7
- ♦ Chapter 2, “Preparing for Staging,” on page 11
- ♦ Chapter 3, “Staging Identity Manager Projects,” on page 21
- ♦ Chapter 4, “Staging Best Practices,” on page 29

Audience

The guide is intended for Identity Manager consultants and customers.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation.

Documentation Updates

For the most recent version of the *Novell Identity Manager 4.0.2 Staging Guide*, visit the [Identity Manager Web site \(https://www.netiq.com/documentation/idm402/idm_staging/data/front.html\)](https://www.netiq.com/documentation/idm402/idm_staging/data/front.html).

Additional Documentation

- ♦ *Understanding Designer for Identity Manager* (http://www.netiq.com/documentation/idm402/designer_intro/data/front.html)
- ♦ *Identity Manager 4.0.2 Integrated Installation Guide* (http://www.netiq.com/documentation/idm402/idm_integrated_install/data/front.html)
- ♦ *Identity Manager 4.0.2 Framework Installation Guide* (http://www.netiq.com/documentation/idm402/idm_framework_install/data/front.html)
- ♦ *Understanding Policies for Identity Manager 4.0.2* (<http://www.netiq.com/documentation/idm402/policy/data/bookinfo.html>)
- ♦ *Policies in Designer 4.0.2* (http://www.netiq.com/documentation/idm36/policy_designer/data/bookinfo.html)
- ♦ *Novell Credential Provisioning for Identity Manager 4.0.2* (http://www.netiq.com/documentation/idm401/policy_credprov/data/bookinfo.html)
- ♦ *Identity Manager 4.0.2 DTD Reference* (http://www.netiq.com/documentation/idm402/policy_dtd/data/bookinfo.html)
- ♦ *Identity Manager 4.0.2 Driver Guides* (<http://www.netiq.com/documentation/idm402drivers/index.html>)

1 Staging Identity Manager Projects

The information covered in the following sections helps you understand the basic principles of staging Identity Manager projects. This chapter includes the following information:

- ♦ [Section 1.1, “Understanding Staging,” on page 7](#)
- ♦ [Section 1.2, “Scope of Staging Documentation,” on page 8](#)
- ♦ [Section 1.3, “Staging Use Cases,” on page 8](#)
- ♦ [Section 1.4, “Using Designer for Staging Identity Manager Projects,” on page 9](#)
- ♦ [Section 1.5, “Moving Authorizations During Staging,” on page 9](#)

1.1 Understanding Staging

Software products need testing before they are deployed in an IT environment. To avoid risk to your production Identity Manager environment, we recommend deploying your Identity Manager projects in separate stages, with improvements made and testing of the project in each stage. This process of developing and testing is called **staging**. Staging provides users the flexibility to validate applications in real time to ensure uniformity across all stages.

As with other software deployment processes, deploying Identity Manager can involve two or more stages, including the development environment, the test environment, and the production environment. Managing the movement of a solution across different stages and ensuring that nothing is left out and that everything functions properly can be challenging for Identity Management customers and consultants.

This guide provides step-by-step procedures for staging your Identity Management projects, enabling you to more easily move each project from the initial stage to all subsequent stages. The guide helps you to reduce complexity in your Identity Manager deployment process by helping you to test your Identity Manager project at multiple stages before the project is live.

Figure 1-1 Staging Identity Manager Projects

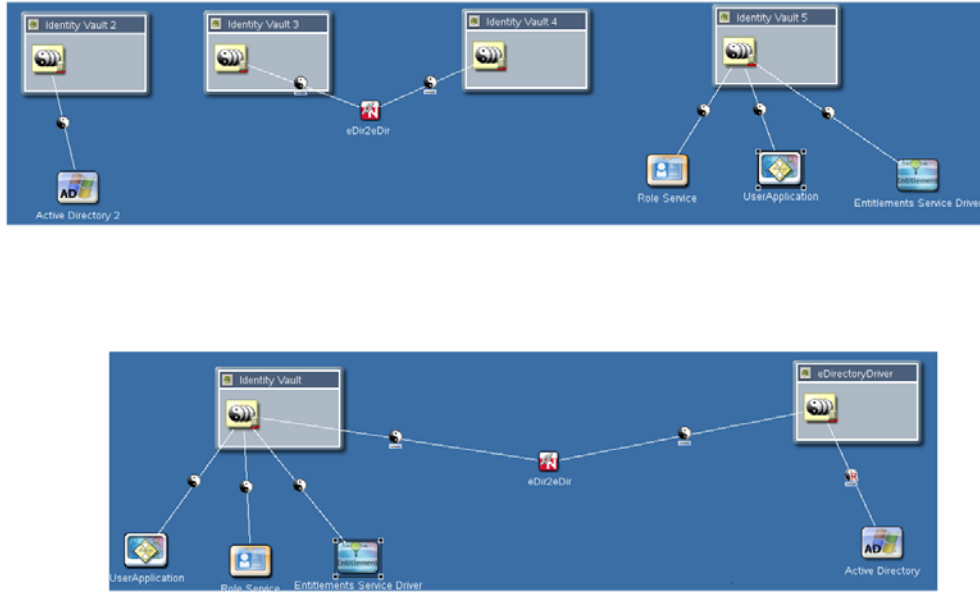


Figure 1-1 shows a basic representation of the movement of Identity Manager projects across different stages. Three projects are developed and tested in three individual environments and then connected in the next stage. The scale of the projects grows with stages, but the scale of an individual driver does not necessarily grow in the same way.

1.2 Scope of Staging Documentation

This guide does not discuss all possible aspects of the staging process. The guide primarily focuses on staging the core components of Identity Manager, particularly the Identity Manager engine and drivers.

The guide does not discuss staging and moving the Identity Manager User Application or any tools associated with that component of Identity Manager. The guide also does not discuss staging the Identity Reporting Module, if installed. Please refer to the [User Application: Design Guide](#) and [Identity Reporting Module Guide](#) for more information on those components.

1.3 Staging Use Cases

The staging discussion includes the following scenarios. Note that some steps in the staging process do not apply to all use cases.

- ♦ **New Deployment:** New drivers and applications are developed during the development stage and are then tested and moved to the test environment. These applications are put together and moved to the production environment.
- ♦ **Existing Deployment:** You already have development, test, and production environments ready and you want to move a new policy from the development environment to the production environment.

1.4 Using Designer for Staging Identity Manager Projects

Designer is used in Identity Manager project development for developing packages, policies, configuration files, and other objects that make up the configuration of a driver. Designer can create the Identity Manager components required for running an Identity Manager project and then deploy the project on another Identity Manager deployment.

You can use Designer to break apart their Identity Manager environment into separate packages, which can then be easily moved from one stage to another. You can also use any version control system with Designer to distribute projects.

Before you begin to use this guide, you should be familiar with Identity Manager and Designer. As you create projects, you should have a uniform Identity Vault design across all the states so that common objects are available. Some objects are moved automatically by Designer, but others should be moved explicitly to make them available in the next stage. See [“Preparing for Staging” on page 11](#) for more information.

1.5 Moving Authorizations During Staging

Moving authorizations across stages is a key issue for an Identity Vault security model. eDirectory™ authorizations are assigned to individual objects or to a collection of objects. These authorizations play an important role in the object security because they determine the permission to access the object to which they have been assigned.

eDirectory authorizations can be performed through Access Control Lists (ACLs) or Security Equivalences/Exclude Roles. Drivers, jobs, RBEs, and so on should have enough permissions to successfully perform the desired operations. See [“Preparing for Staging” on page 11](#) for more information.

2 Preparing for Staging

The information covered in the following sections helps you design your Identity Manager projects in order to stage them easily. This chapter includes the following information:

- ♦ [Section 2.1, “Converting Configuration Files to Packages,” on page 11](#)
- ♦ [Section 2.2, “Prerequisites,” on page 11](#)
- ♦ [Section 2.3, “Identity Vault Structure,” on page 12](#)
- ♦ [Section 2.4, “Driver Configuration,” on page 12](#)
- ♦ [Section 2.5, “Identity Manager Objects,” on page 13](#)
- ♦ [Section 2.6, “Rights,” on page 18](#)

2.1 Converting Configuration Files to Packages

Before you begin the staging process, we recommend you convert any configuration files in your Identity Manager environment into packages. Packages are much more portable than configuration files and enable you to move your custom policies and other content independently from the server and communication settings needed for a particular stage.

For information about converting configuration files to packages, see [“Upgrading Drivers to Packages,”](#) in the *Identity Manager 4.0.2 Upgrade and Migration Guide*.

2.2 Prerequisites

Ensure that the following general prerequisites are met before attempting staging:

- ♦ All the stages should have the same version of the eDirectory, Identity Manager, and Identity Manager drivers.
- ♦ Designer 3.5 or later is present.
- ♦ All the applications and drivers are fully developed and tested in one stage before moving them to the next stage.
- ♦ From your project, gather information about the objects that are not modeled by Designer. For more information, see [“Objects That Designer Does Not Model” on page 14](#).
- ♦ Create an LDIF file for all the objects that are not modeled by Designer. Use Designer to import the additional objects.

You should also be aware of the recommended best practices for moving Identity Manager objects across stages. For more information about staging best practices, see [“Staging Best Practices” on page 29](#).

2.3 Identity Vault Structure

An Identity Vault is typically a flat eDirectory tree, which consists of several containers for users, devices, groups, objects, and so on. Objects are stored in different containers for performance reasons.

Make sure that you are familiar with the basic principles of directory design. A uniform directory design simplifies administrative tasks for staging. For more information on directory design, refer to “Directory Design for Identity Management Solutions” (<http://www.novell.com/coolsolutions/appnote/14533.html>).

2.4 Driver Configuration

You must create a common data model to allow drivers to work together.

Even though each driver is unique and uses different policies, all drivers use the same guidelines to make the driver configuration file consistent. For example, all policies and driver configuration files have the same naming conventions and support the same common data module.

See “Identity Manager Driver Configuration Development Guidelines” (http://www.novell.com/documentation/ncmp10/rk12_architecture/data/bg89kav.html) for guidelines on developing new drivers.

- ♦ Section 2.4.1, “Using GCVs in Policies,” on page 12
- ♦ Section 2.4.2, “Simulation and Staging,” on page 12

2.4.1 Using GCVs in Policies

Global Configuration Values (GCV) are global configuration values or constants, not global variables. There is no way to change a GCV value at runtime. GCVs are globally accessible to the driver and driver set, but not to the tree or network. GCVs can be consumed by all drivers in a driver set or by all policies in a driver. For more information on using GCVs, see “Configuring Global Configuration Objects” in the *Designer 4.0.2 for Identity Manager 4.0.2 Administration Guide*.

2.4.2 Simulation and Staging

The Policy Simulator allows you to test and debug a single policy or a group of policies contained in a policy set or all the policies in a driver or a driver set without implementing the policy in the Identity Vault. It also provides a graphical editor to create the XDS Input documents. You can use these features to test the policies without affecting the production environment or the connected system. This means that you can essentially use Designer as the first stage in your deployment process by developing and then testing your policies through simulation.

2.5 Identity Manager Objects

Designer provides the ability to develop Identity Manager projects even in offline mode. You can easily move your Identity Manager objects from one environment to another. You can also export and import projects into a simple configuration file, which can be stored for future use.

Some Identity Manager objects are not visible in a Designer project, even though they may be necessary for your Identity Manager installation. To ensure that you move all necessary objects from one stage to another, you should import any objects not modeled in Designer from eDirectory into an LDIF container, back up those objects by exporting the LDIF container to an external LDIF file, and then import the LDIF file to an LDIF container in the next stage.

2.5.1 Objects That Designer Models

You can model the following objects in Designer:

Object	Description
Driver Sets	A driver set is a container that holds Identity Manager drivers. Only one driver set can be active on a server at a time. As a result, all active drivers must be grouped into the same driver set.
Drivers	A driver provides the connection between an application and the Identity Vault. The driver is the connector that enables data synchronization and sharing between systems.
GCVs on Driver set and Drivers	Global configuration values (GCVs) are settings that are similar to driver parameters. GCVs can be specified for an individual driver as well as a driver set. If a driver does not have a GCV, the driver inherits the value for that GCV from the driver set.
Policies	Policies cover DirXMLScript, ECMAScript, Entitlement, MappingTable, Resource, Credential Application, Credential Respository, SchemaMap, Filter, and XSLT.
Libraries	You need to provide a context if the library is outside the driver set.
Provisioning Objects	Workflows, roles, resources, teams, etc.
Notification Templates	Notification templates enable you to customize and send e-mail messages that users receive when triggers occur.
Identity Vault Schema, Application Schema	
Role Based Entitlements	Identity Manager allows you to synchronize data between connected systems. Entitlements allow you to set up criteria for a person or group that, once met, initiate an event to grant or revoke access to business resources within the connected system.
Named Passwords	

2.5.2 Objects That Designer Does Not Model

Object	Description
O (Organization) and OU (Organizational Unit)	<p>Ensure that O or OU objects are created before deploying them.</p> <p>Import the containers that contain O or OU objects. The following objects must be included in O or OU objects:</p> <ul style="list-style-type: none">◆ All O or OU objects that are Security Equivalences objects for any drivers.◆ O or OU objects that are used in any policies.◆ O or OU objects that are used in any job configurations.◆ O or OU objects that are used in GCVs.
Users	<p>Ensure that the User objects are created before deploying them, especially the admin users. The list of users can be collected in two different ways:</p> <p>Import the containers that contain the user objects. The following objects must be included in the list:</p> <ul style="list-style-type: none">◆ Security Equivalences and Exclude Administrator Roles for all the drivers.◆ Static Members on groups and RBE policies.◆ Search identities and Membership Filter on Dynamic groups and RBE policies.◆ Users that are used in any policies.◆ Users that are used in any job configurations.◆ Users that are used in GCVs.
Groups	<p>Ensure that the static and dynamic group objects are created before deploying them.</p> <p>Import the containers that contain the groups. The following objects must be included in the list:</p> <ul style="list-style-type: none">◆ Groups that are used in any policies.◆ Groups that are used in any job configurations.◆ Groups that are used in GCVs.
Password Policies	<p>Ensure that the policies are created before deploying them.</p>
Indices	<p>Ensure that indices are created before deploying them.</p>
Custom Objects	<p>User-defined objects are not defined in Designer. Manually create them before deploying.</p> <p>Import the containers that contain the custom objects. The following objects must be included in the list:</p> <ul style="list-style-type: none">◆ All custom objects that are Security Equivalences objects for all the drivers.◆ Custom objects that are used in any policies.◆ Custom objects that are used in any job configurations.◆ Custom objects that are used in GCVs.

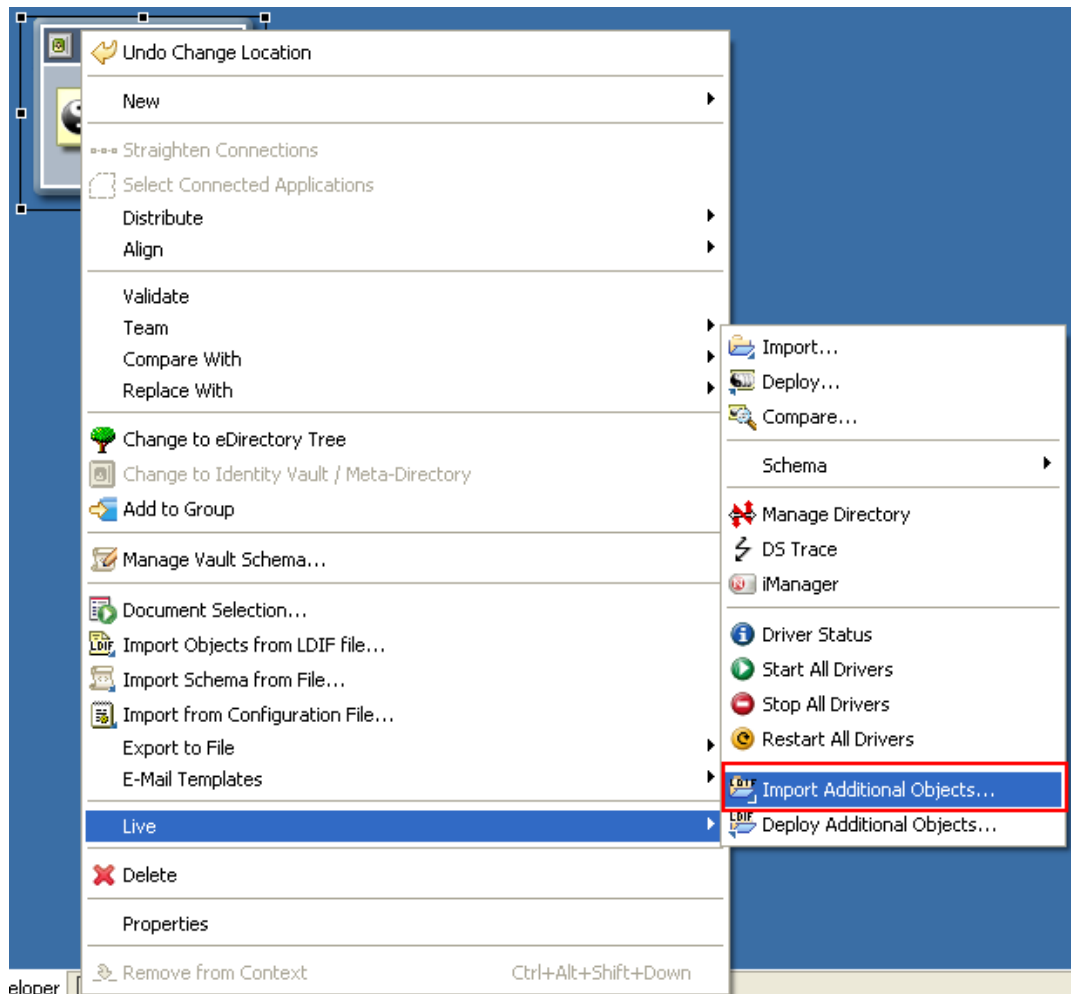
Designer 3.5 and later allows you to import objects listed in the above table in LDIF format and then deploy them along with other objects that are being deployed.

NOTE: These objects are not modeled as drivers or driver sets in Designer. They can be modified by modifying the LDIF file that contains these objects in Designer. For more information, refer to [“Importing Objects” on page 15](#).

2.5.3 Importing Objects

Before copying a staged project, you should import any additional objects not modeled in Designer from eDirectory into an LDIF container. For information about objects not modeled in Designer, see [“Objects That Designer Does Not Model” on page 14](#).

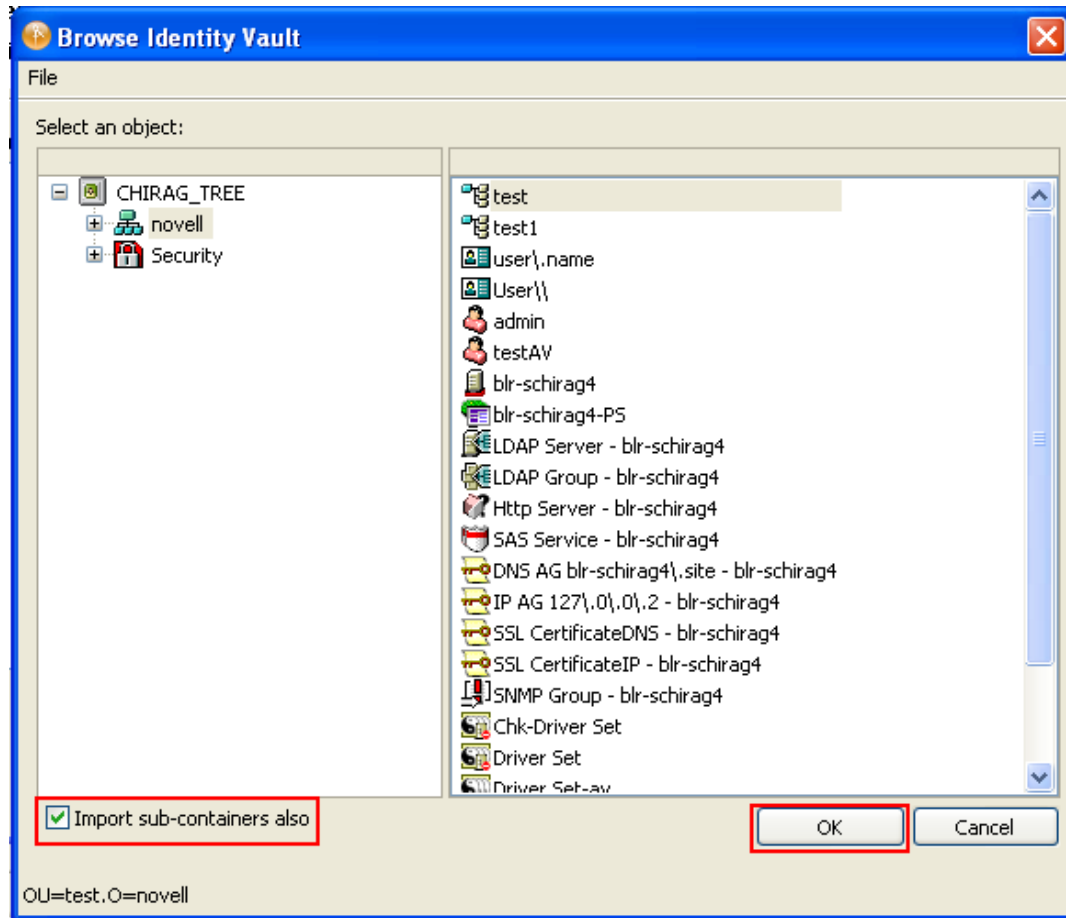
- 1 In Designer, right-click *Identity Vault* and select *Live > Import Additional Objects*.



- 2 Browse to and select the objects you want to add to the LDIF file.

Or

If you want to select all the objects in a container, select *Import sub-containers also* in the Browse Identity Vault dialog box.



- 3 Click *OK*.
- 4 Click *Continue* on the Import Dialog to import all the objects into Designer. These objects are stored in the LDIF container.

NOTE: These objects are retrieved by an LDAP channel. If you are running the LDAP service on non-default ports, see [“Changing the LDAP Properties” on page 28](#) for more information.

- 5 Repeat [Step 1](#) through [Step 4](#) for all the Identity Vaults in your projects.

You can edit the LDIF objects from the LDIF container. Go to the Outline View, expand the Identity Vault, then double-click the LDIF container.

The objects in the LDIF container are overwritten. To keep a record of the objects stored in the LDIF container, export the information of the LDIF container into an LDIF file. For more information about exporting LDIF objects to an LDIF file, see [“Exporting LDIF Container Objects to an LDIF File” on page 17](#).

IMPORTANT: You should back up your project by using a version control system or export it to a file.

2.5.4 Exporting LDIF Container Objects to an LDIF File

To back up the objects currently stored in the LDIF container in your project, you can export those objects and store them in the LDIF file.

- 1 In the Modeler, select *Identity Vault > Live > Export to File > Additional Objects*.
- 2 In the window displayed, specify the name of the file into which objects stored in the LDIF container should be exported, then click *Save* or *OK*. The following window is displayed in a Windows environment.
If there is no information in the LDIF container, a warning is displayed.
- 3 If prompted, click *OK*.

2.5.5 Importing Objects from an LDIF File into an LDIF Container

You can import objects from an LDIF file to an LDIF container. This overwrites the existing objects in the LDIF container.

- 1 In the Modeler, right-click the *Identity Vault* and select *Import Objects from LDIF File*.
- 2 In the window displayed, select the file from which the LDIF objects should be imported, then click *Open* or *OK*.
- 3 If prompted, click *OK*.

2.5.6 Deploying Additional Objects into eDirectory

Designer allows you to update objects that are already present in eDirectory. The current functionality does not support the deployment of objects containing references to objects in eDirectory. To deploy these objects, you can either manually create the objects in eDirectory or remove the references. For more information about modifying LDIF container data, see [“Editing the LDIF Container Data by Using an Editor” on page 17](#).

Designer allows you to update objects that are in eDirectory.

- 1 In the Modeler, select *Identity Vault > Live > Deploy Additional Objects*.
- 2 Select or deselect the objects by clicking the *Select All* icon, then click *Deploy*.
If the objects are already present in eDirectory, a warning is displayed.
- 3 If the objects are already in eDirectory, click *Update Existing Objects in eDirectory* to update them, click *Deploy*, then click *OK*.
If there are no objects or the information is not in a proper format in the LDIF container, a warning is displayed.

2.5.7 Editing the LDIF Container Data by Using an Editor

Designer allows you to modify the LDIF objects by using different editors. Under the Identity Vault in the Outline view, right-click the LDIF container and select *Open With > Designer Built-in Editor*. You can also double-click the LDIF container to open the container.

You can then use the built-in editor to modify the LDIF data stored in the container.

2.5.8 Deleting the LDIF Container

Designer allows you to delete the LDIF container. Under the Identity Vault in the Outline view, right-click the LDIF container and select *Delete*.

2.6 Rights

- ♦ [Section 2.6.1, “Driver Equivalences,” on page 18](#)
- ♦ [Section 2.6.2, “Roles Based Entitlements Policies,” on page 18](#)
- ♦ [Section 2.6.3, “Jobs,” on page 19](#)

2.6.1 Driver Equivalences

Designer allows you to define and deploy the Security Equivalences objects for the drivers in eDirectory.

Security Equivalences require rights to the objects within the Identity Vault in order to perform tasks on them. For example, an Oracle™ database driver has a policy to create a user in the Identity Vault in a container every time a user is created in the database, but the driver doesn't have enough permissions on the container to create the user, so the process fails. The driver has similar rights as that of the users/objects who have permissions on the container. All the policies should be carefully evaluated for finding out what permissions should be given to the drivers.

Designer 3.5 and later can store the Security Equivalences and Exclude Administrative Roles of the drivers in the project and can assign them to the drivers. Before moving to another staging environment, ensure that you know the Security Equivalences and Exclude Administrative Roles associated with each driver and ensure that these objects are imported as LDIF objects and moved along with other objects before being assigned in the next stage after deployment.

If the Security Equivalences object and the Exclude Administrative Roles objects are stored as LDIF objects, Designer ensures that they are created in the next stage before they are assigned.

For more information about Security Equivalence, see [“Establishing a Security Equivalent User”](#) in the *Identity Manager 4.0.2 Security Guide*.

2.6.2 Roles Based Entitlements Policies

Roles Based Entitlements policies are used by the Entitlements Service driver, which grants entitlements to and revokes entitlements from the users.

An entitlement policy contains the following:

Membership: The list of users assigned to a policy. A user can be dynamically assigned to a policy when he or she meets the criteria for the policy, or the user can be statically (manually) assigned to the policy.

Entitlements: The list of entitlements associated with the policy. Users assigned to the policy receive all of the entitlements associated with the policy. If the user is removed from the policy, he or she loses all entitlements associated with the policy.

You can assign any Identity Vault objects for which you want the entitlement policy to be a trustee. Each member of the policy becomes a trustee of the objects you add.

There are several reasons why you might want to make the policy a trustee of an object:

- ♦ One of the policy's entitlements requires the policy's members to have rights to an object.
- ♦ You want to use the policy to assign users as trustees of an object even though rights to the object are not required for an entitlement. In this case, you are using the entitlement policy to grant and revoke trustee rights for members of the policy.

These rights are not stored in Designer. You should assign the rights after moving to the next stage.

2.6.3 Jobs

Identity Manager has a job scheduling utility that schedules events, such as setting the system to disable an account on a specific day, or initiating a workflow to request an extension for a person to access a corporate resource. The Job Manager runs on every Identity Manager server in the background. Based on the job definition, it checks every minute to see if a job needs to run. When it encounters a job, it runs the appropriate Job implementation.

The Job Manager needs appropriate permissions to run successfully. For example, a job that disables a user account from the Identity Vault needs adequate permissions. Appropriate access must be granted to the job object in the Identity Vault so that it can modify a user object. Use iManager to grant the required rights for the jobs because Designer does not allow you to grant rights for jobs.

3 Staging Identity Manager Projects

This chapter contains the following information:

- ♦ [Section 3.1, “Staging a Project for the First Time,” on page 21](#)
- ♦ [Section 3.2, “Staging Changes in an Existing Project,” on page 25](#)
- ♦ [Section 3.3, “Copying Java Class .jar Files Between Stages,” on page 26](#)
- ♦ [Section 3.4, “Post-Staging Tasks,” on page 26](#)
- ♦ [Section 3.5, “Changing the LDAP Properties,” on page 28](#)

3.1 Staging a Project for the First Time

You should ensure that all the applications and Identity Manager systems are up and running in the next stage before moving the configurations. You can stage projects using either packages or configuration files, as necessary in your environment.

- ♦ [Section 3.1.1, “Staging Using Packages,” on page 21](#)
- ♦ [Section 3.1.2, “Staging Using Configuration Files,” on page 21](#)
- ♦ [Section 3.1.3, “First-Time Staging Process,” on page 22](#)

3.1.1 Staging Using Packages

The simplest, most efficient way to stage your Identity Manager project is by using the package functionality included in Identity Manager 4.0 and later.

We recommend using this approach because unlike configuration files, packages are configured to keep server-specific settings separate from the actual Identity Manager content. You move all your policies from one stage to the next, not your server configurations.

For more information about converting configuration files to packages, see [“Converting Configuration Files to Packages” on page 11](#) in this guide and [“Upgrading Drivers to Packages,”](#) in the *Identity Manager 4.0.2 Upgrade and Migration Guide*.

3.1.2 Staging Using Configuration Files

Users with Identity Manager version 3.6 installed may need to perform the staging process using configuration files. Because of the difficulty inherent in updating configuration files, we do not recommend using this process but instead recommend converting your existing configuration files to packages.

For more information about converting configuration files to packages, see [“Converting Configuration Files to Packages” on page 11](#) in this guide and [“Upgrading Drivers to Packages,”](#) in the *Identity Manager 4.0.2 Upgrade and Migration Guide*.


3.1.3 First-Time Staging Process

To stage an Identity Manager project for the first time, complete the following steps:

- 1 Import any additional objects not modeled in Designer from eDirectory into an LDIF container. For more information on importing additional objects, see [“Importing Objects” on page 15](#).
- 2 Compare and import any schema changes from the Identity Vault (eDirectory) to the schema in Designer:
 - 2a Right-click *ID Vault > Live > Schema > Compare*.
 - 2b In the Information pane, select *Update Designer*.
 - 2c Click *Reconcile*.
 - 2d Click *OK*.
- 3 (Optional) If you want to keep a backup of your first-stage project, you can export the existing project to an archive file:
 - 3a Right-click the first-stage project and select *Export Project*.
 - 3b Select the project you want to export.
 - 3c Click *Browse* and specify the name of the archive file you want to use, then click *OK*.
 - 3d Click *Finish*.
 - 3e Click *OK*.
- 4 Copy the first-stage project to reuse it in the next stage:
 - 4a In Designer, go to *Window > Show View > Project*.
 - 4b Right-click the first-stage project and select *Copy Project*.
 - 4c Enter the name for the second-stage project. We recommend you use a name that clearly indicates the project is used for the second stage of the staging process.
 - 4d Click *OK*.
- 5 (Optional) After copying the existing first-stage project, you may want to rename the project to specify that the project is for the first stage. Complete the following steps to rename the first-stage project:
 - 5a In the project view in Designer, right-click the first stage project and select *Rename*.
 - 5b Specify a new name for the project and click *OK*.
- 6 In the project view, expand the second-stage project and double-click *System Model*.
- 7 Change the configuration of one of the Identity Vaults in your project.
 - 7a In the Outline view or the Modeler, double-click the *ID Vault*.
 - 7b In the Configuration page, change the *Hostname*, *Admin Username*, and *Admin Password* settings to match those of the Identity Vault you want to use for the second-stage project.
 - 7c Click *Test Connection* to check the connectivity, then click *OK*.
 - 7d If necessary, add more servers and associate those servers with the driver set.
- 8 (Optional) If your second-stage project uses one or more different connected systems, change the configuration of the connected system or systems of the second-stage Identity Vault. To change the system configuration, complete the following steps:
 - 8a In the Modeler, double-click a driver or a driver line.
 - 8b In the Driver Configuration page, change the authentication information in the *Authentication* tab.

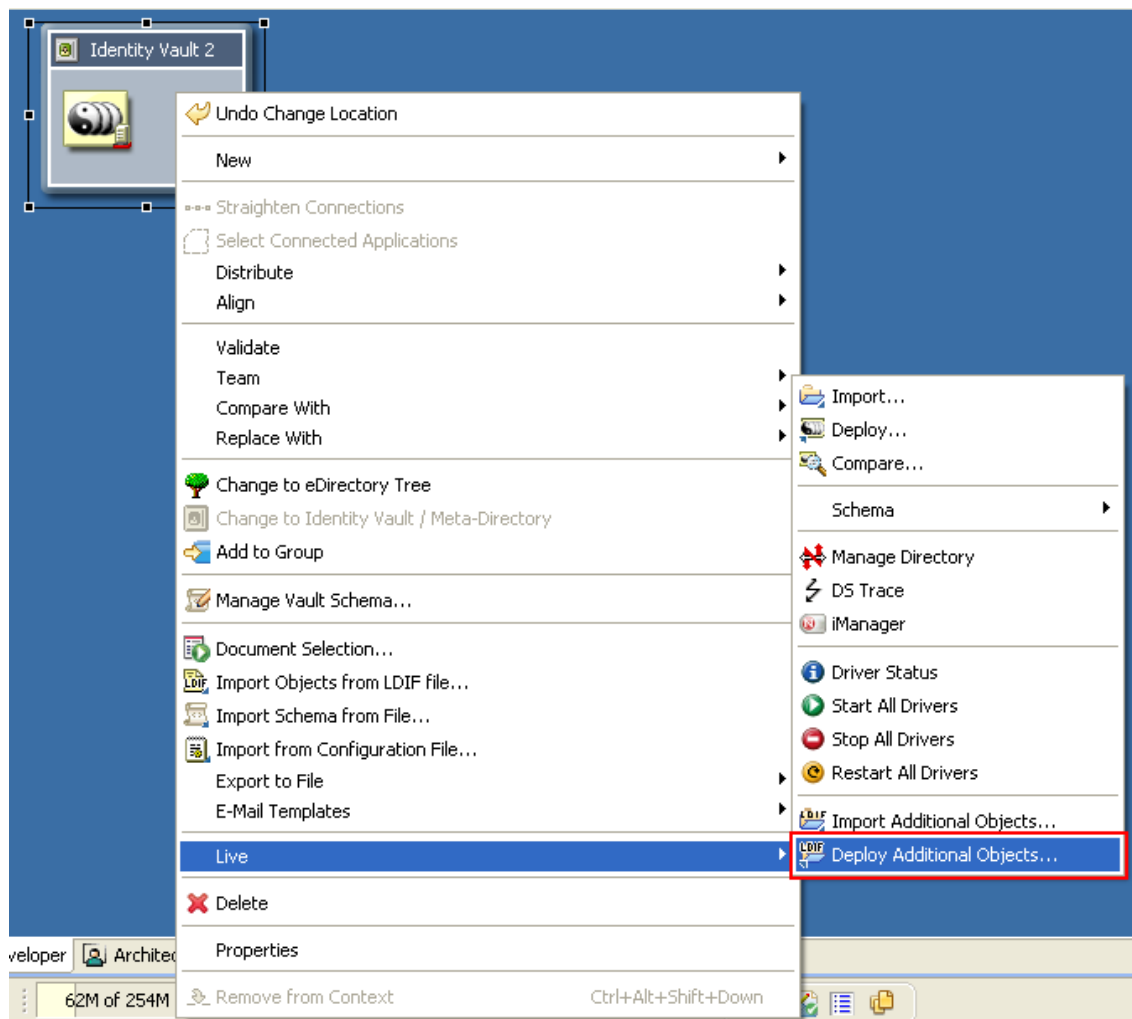
- 8c** In the Driver properties page, change the driver related information in the *Driver Parameters* tab.
- 8d** The driver parameters depend on the servers on which the drivers reside. Ensure that you change the driver parameters on multiple servers if you have multiple servers running a driver.
- 9** (Optional) If your second-stage project uses a different connected system or different configuration settings for provisioning, change the GCVs for the drivers and driver sets of the second-stage Identity Vault as necessary.

GCVs should be the only changes that you make on the drivers and the driver set along with the configuration. Your policies won't change if they are properly designed.

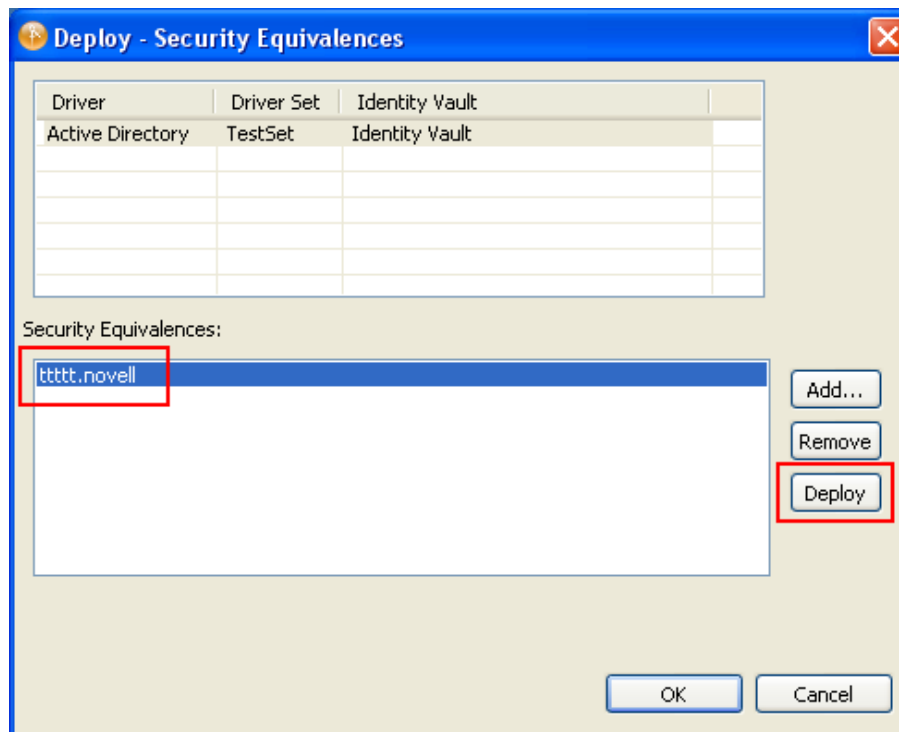
 - 9a** Update all the GCVs that change with the environment, as necessary.
 - 9b** Move or add new GCVs to any new servers added in [Step 7d](#).
- 10** To ensure the integrity of your project, run the Project Checker:
 - 10a** Click *Window > Show View > Project Checker*.
 - 10b** In the Project Checker view, click the *Run the Project Checker* icon .
 - 10c** Review the results and correct any issues. For more information about using the Project Checker, see "[Checking Your Projects](#)" in the *Designer 4.0.2 for Identity Manager 4.0.2 Administration Guide*.
- 11** Compare and import any schema changes from Designer into the second-stage the Identity Vault (eDirectory):

Compare the schema in Designer with the eDirectory schema and deploy:

 - 11a** Right-click *ID Vault > Live > Schema > Compare*.
 - 11b** In the Information pane, select *Update eDirectory*.
 - 11c** Click *Reconcile*.
 - 11d** Click *OK*.
- 12** In the Modeler, right-click the Identity Vault and select *Live > Deploy Additional Objects* to deploy additional objects gathered in the [Prerequisites](#).



- 13 To deploy the Identity Vault, right-click *ID Vault* > *Live* > *Deploy*.
- 14 Deploy the appropriate Security Equivalences and Exclude Admin Roles objects for each driver. See [Section 2.6, "Rights,"](#) on page 18 for more information.



15 Repeat [Step 7](#) through [Step 14](#) for each Identity Vault in your project.

3.2 Staging Changes in an Existing Project

Once you stage an Identity Manager project for the first time using packages, you can also move later changes to your environment between the stages you established.

IMPORTANT: We strongly recommend against using configuration files to move changes from one stage to another, because any customizations you have made can be overwritten. Instead, convert your existing configuration files into packages.

For more information about converting configuration files to packages, see [“Converting Configuration Files to Packages” on page 11](#).

- 1 In Designer, make any changes or customizations to your driver to a single package in your first-stage project. You can then test those changes in a non-production environment before moving any customizations to a subsequent stage for production use.
- 2 If you modify any policies, entitlements, or prompts on the first-stage driver, right-click each object in the Outline view and select *Sync to Package*.

NOTE: Any changes not synced to the change package will not be moved to the next stage.

- 3 When finished making changes, navigate to the change package in the Package Catalog.
- 4 Right-click the package and select *New Package Version*.
- 5 In the New Version window, increment the patch-level version number by 1, then click *Next*.
- 6 Click *Next* two more times to confirm existing package information, then click *Finish*. Designer creates a new version of the change package with an incremented version number and new date stamp.

- 7 Right-click the new version of the change package and select *Build*.
- 8 Right-click the package and click *Build*.
- 9 Click *Browse*, then browse to and select the directory where you want to build the package.
- 10 Click *OK* twice.
- 11 Review the summary information, then click *OK*.
- 12 In the Project view, open your second-stage project.
- 13 In the Outline view, right-click *Package Catalog* and select *Import Package*.
- 14 Click *Browse*, then browse to and select the .jar file for the first-stage package you built and click *OK*.
- 15 Click *OK* to import the selected package.
- 16 Review the import message, then click *OK*.

3.3 Copying Java Class .jar Files Between Stages

If you use any custom Java-based functionality in your Identity Manager policies, you must ensure you move that functionality from one stage to the next. Designer does not automatically move this functionality between stages.

To move this functionality, you must use SCP or FTP to manually copy each related Java class, stored in a .jar file, from the first-stage computer to the second-stage computer. Note that this process takes place outside of Designer and requires moving the actual files from one computer to another, rather than from one project to another.

IMPORTANT: You must copy all Java class .jar files from one stage to the next each time you stage your Identity Manager environment, even if you have made no changes to the .jar files.

Identity Manager typically stores Java classes in the following directory:

```
/opt/novell/edirectory/lib/dirxml/classes
```

However, in some environments, Java classes may be stored in a different Java classpath. To determine if your driver set uses a different Java classpath, complete the following steps:

- 1 Right-click the driver set in the Designer Modeler and select *Properties*.
- 2 Click *Java*. Designer displays any additional Java classpath locations in the *Classpath additions* field.

3.4 Post-Staging Tasks

Designer does not move all the configurations to the next stage. Users are expected to manually perform a few tasks to ensure that the configurations work properly.

- ♦ **Security Equivalences and Exclude Admin Roles:** Check whether all the drivers have appropriate Security Equivalences and Exclude Admin Roles objects, as defined in the previous stage. For more information, see [“Driver Equivalences” on page 18](#).
- ♦ **Bi-directional eDirectory (eDir2eDir) Driver Certificates:** If you have Bi-directional eDirectory driver certificates created in the current stage, ensure that these certificates are created in the next stage.
 1. In Designer, right-click the *Bi-directional eDirectory* driver and select *Secure Connection Settings*.

2. Click *Enable SSL/TLS*, select the required options, then click *OK*.
 3. Right-click *Bi-directional eDirectory*, then click *Live > Create eDir-to-eDir Certificates*.
- ♦ **Java Environment Parameters:** The Java* environment parameters enable you to configure the Java Virtual Machine™ (JVM) on the Metadirectory server associated with the driver set. You might need to change the Java classpath options if the .jar files your Metadirectory server is looking for reside at a different place in the new stage. To change the location, go to *DriverSet > Properties > Java > Classpath additions* and provide the correct classpaths. When you enter multiple classpaths, separate them with a semi colon (;) for a Windows JVM and a colon (:) for a UNIX* or Linux* JVM. Deploy the driver set if you make any changes.
 - ♦ **Indexes:** Make sure that all the customized indexes from the previous stage have been copied to the new stage. eDirectory uses these indexes to significantly improve the query performance. Some indexes are shipped with eDirectory. These default indexes are for the following attributes:
 - ♦ CN
 - ♦ Aliased Object Name
 - ♦ dc
 - ♦ Obituary
 - ♦ Given Name
 - ♦ Member
 - ♦ Surname
 - ♦ Reference
 - ♦ uniqueID
 - ♦ Equivalent to Me
 - ♦ GUID
 - ♦ NLS: Common Certificate
 - ♦ cn_SS
 - ♦ Revision
 - ♦ uniqueID_SS
 - ♦ extensionInfo
 - ♦ ldapAttributeList
 - ♦ ldapClassList

You can visit each Identity Vault server and collect the customized index information by doing the following:

1. In Novell® iManager, click the *Roles and Tasks* tab.
2. Click *eDirectory Maintenance > Indexes*.
3. Select a server from the list of available servers.
iManager lists all the active and offline indexes on the selected server.
4. Make a note of all the customized indexes.

Ensure that you add these indexes to the corresponding servers in the next stage. See “Index Manager” (<http://www.novell.com/documentation/edir88/edir88/data/a5tuuu5.html>) in the *Novell eDirectory 8.8 SP7 Administration Guide* (<http://www.novell.com/documentation/edir88/edir88/data/a2iii88.html>) for more information on creating, adding, or deleting indexes.

- ♦ **Password Policies:** Ensure that password policies assigned to the containers, users, groups in the previous stage are assigned again in the current stage.
- ♦ **Challenge Response Objects:** In addition to password policies, ensure that you migrate or recreate any challenge response objects used in the previous stage in the current stage. You can either import your existing challenge response objects into the LDIF container in the first-stage project or note the details of the challenge response objects in the first-stage project and create new objects in the next stage.

For information importing objects into the LDIF container, see [“Importing Objects” on page 15](#).

- ♦ **Restarting All Drivers:** Start the drivers after moving the driver configuration to the next stage. In the Modeler, right-click each driver and select *Driver > Start Driver*.

3.5 Changing the LDAP Properties

You can modify non-default LDAP properties by using the Properties view of the Identity Vault. This is used for importing and deploying objects.

- 1 Under the Properties view, click *ID Vault*, then specify the LDAP clear text port and LDAP secure port numbers.

Property	Value
1. Identity Vault	
Name	Identity Vault 2
Host Address	164.99.136.134
User Name	admin.novell
Password	*****
Context	
ldapClearTextPort	389
ldapSecurePort	636
useLDAPSecureChannel	false
2. Administrator	
Name	
Cell	
Department	
E-mail	
Fax	
Location	
Notes	
Pager	
Phone	
Title	

- 2 Save the project.

4 Staging Best Practices

- ♦ If you delete drivers and driver sets from Stage 2 in order to deploy the drivers from Stage 1, you can lose the associations.
- ♦ We recommend you use the same Designer workspace for all stage projects.
- ♦ Don't deploy Stage 1 objects directly into the Stage 2 environment.
- ♦ When performing the staging process, ensure you store server-specific settings as GCVs. If using packages, you can then leave those GCVs behind when you change stages. However, if using configuration files, you need to copy the GCVs from one stage to the next so they do not get overwritten.
- ♦ When creating new GCVs for staging, ensure you add those GCVs at the driver set level.
- ♦ Before moving to any stage, understand the existing stage and the objects that Designer does not automatically bring in (see [“Objects That Designer Does Not Model”](#) on page 14) for the next stage.

Ensure that you know which objects are required in the subsequent stages. Consolidate these objects in the LDIF file.

- ♦ You can store any eDirectory objects not modeled in Designer as a DS object in your first-stage project and add that object to a package so that you can move the DS object to the second-stage.
- ♦ Ensure that you assign the Security Equivalences, Trustees, and Server Certificates of Stage 1 in Stage 2 after deployment.
- ♦ LDIF files that contain additional objects should be stored locally. You can use the Import Convert Export (ICE) utility to deploy these objects in any stage.
- ♦ For a new deployment in Stage 2, ensure that LDIF objects are deployed before importing the configuration file or the project file.
- ♦ For an existing deployment in Stage 2, ensure that you compare the existing project with the Stage 1 configuration, deploy the necessary LDIF objects, then import the configuration file.
- ♦ Ensure that objects are up-to-date when you import them into the LDIF file.

Always import the additional objects into Stage 1 before moving to Stage 2.

- ♦ Export the additional objects of Stage 1 into an LDIF file before moving to Stage 2 so that these objects can be manually created in Stage 2 before deployment.
- ♦ Rather than directly modifying your filters, we recommend you create a filter object in your first-stage project, add that filter object to a package, and then install the package on a driver in your first-stage project. Using the package, you can then easily move any changes to the filter to the next stage.

