Novell Identity Manager

3.0

December 8, 2005

www.novell.com

IDENTITY MANAGER USER APPLICATION: ADMINISTRATION GUIDE

Novell®

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.

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

Title to the Software and its documentation, and patents, copyrights and all other property rights applicable thereto, shall at all times remain solely and exclusively with Novell and its licensors, and you shall not take any action inconsistent with such title. The Software is protected by copyright laws and international treaty provisions. You shall not remove any copyright notices or other proprietary notices from the Software or its documentation, and you must reproduce such notices on all copies or extracts of the Software or its documentation. You do not acquire any rights of ownership in the Software.

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

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

Novell Trademarks

Novell is a registered trademark of Novell, Inc., in the United States and other countries.

SUSE is a registered trademark of Novell, Inc., in the United States and other countries.

Third-Party Materials

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

Third-Party Legal Notices

The Apache Software License, Version 1.1

Copyright (c) 2000 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)."

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "Apache" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.
5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Autonomy

Copyright ©1996-2000 Autonomy, Inc.

Bouncy Castle

License Copyright (c) 2000 - 2004 The Legion Of The Bouncy Castle (http://www.bouncycastle.org)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER

IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Castor Library

The original license is found at http://www.castor.org/license.html

The code of this project is released under a BSD-like license [license.txt]:

Copyright 1999-2004 (C) Intalio Inc., and others. All Rights Reserved.

Redistribution and use of this software and associated documentation ("Software"), with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain copyright statements and notices. Redistributions must also contain a copy of this document.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name "ExoLab" must not be used to endorse or promote products derived from this Software without prior written permission of Intalio Inc. For written permission, please contact info@exolab.org.
4. Products derived from this Software may not be called "Castor" nor may "Castor" appear in their names without prior written permission of Intalio Inc. Exolab, Castor and Intalio are trademarks of Intalio Inc.
5. Due credit should be given to the ExoLab? Project (http://www.exolab.org/).

THIS SOFTWARE IS PROVIDED BY INTALIO AND CONTRIBUTORS ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL INTALIO OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Indiana University Extreme! Lab Software License

Version 1.1.1

Copyright (c) 2002 Extreme! Lab, Indiana University. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Indiana University Extreme! Lab (http://www.extreme.indiana.edu/)."

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "Indiana University" and "Indiana University Extreme! Lab" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact http://www.extreme.indiana.edu/.
5. Products derived from this software may not use "Indiana University" name nor may "Indiana University" appear in their name, without prior written permission of the Indiana University.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS, COPYRIGHT HOLDERS OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,

INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

JDOM.JAR

Copyright (C) 2000-2002 Brett McLaughlin & Jason Hunter. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the disclaimer that follows these conditions in the documentation and/or other materials provided with the distribution.
3. The name "JDOM" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact license@jdom.org.
4. Products derived from this software may not be called "JDOM", nor may "JDOM" appear in their name, without prior written permission from the JDOM Project Management (pm@jdom.org).

In addition, we request (but do not require) that you include in the end-user documentation provided with the redistribution and/or in the software itself an acknowledgement equivalent to the following: "This product includes software developed by the JDOM Project (http://www.jdom.org/)."

Alternatively, the acknowledgment may be graphical using the logos available at http://www.jdom.org/images/logos.

THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JDOM AUTHORS OR THE PROJECT CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Phaos

This Software is derived in part from the SSLavaTM Toolkit, which is Copyright ©1996-1998 by Phaos Technology Corporation. All Rights Reserved. Customer is prohibited from accessing the functionality of the Phaos software.

W3C

W3C® SOFTWARE NOTICE AND LICENSE

This work (and included software, documentation such as READMEs, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

Permission to copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, the W3C Software Short Notice should be included (hypertext is preferred, text is permitted) within the body of any redistributed or derivative code.
3. Notice of any changes or modifications to the files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

Contents

About This Book

Purpose

This book describes how to administer the Novell Identity Manager *user application*, including:

- The *identity self-service* features provided with Identity Manager
- The *workflow-based provisioning* features provided if you add the Provisioning Module for Identity Manager

To learn about administering the other features of Identity Manager (which are common to all packagings), see the *Novell Identity Manager: Administration Guide*.

Audience

The information in this book is for *system administrators, architects, and consultants* who are responsible for *configuring, deploying, and managing* the identity self-service features and/or workflow-based provisioning features of the Identity Manager user application.

End-user documentation for these features is provided in the *Identity Manager User Application: User Guide*.

Prerequisites

This book assumes that:

- *You have installed* Identity Manager, and possibly the Provisioning Module for Identity Manager as well

For instructions on installing these products, see the *Novell Identity Manager: Installation Guide*.

- *You have configured* the other features of Identity Manager as appropriate for your needs

See the *Novell Identity Manager: Administration Guide*.

Organization

Here's a summary of what you'll find in this book:

Part	Description
Part I, "Overview," on page 17	Introduces you to the Identity Manager user application, and helps you plan for its use in your organization
Part II, "Configuring the User Application Environment," on page 57	How to configure various aspects of the Identity Manager user application environment (including user application driver, directory abstraction layer, and logging) to meet the needs of your organization
Part III, "Administering the User Application," on page 123	How to configure and manage the Identity Manager user application by using the Administration tab of the user interface

Part	Description
Part IV, “Portlet Reference,” on page 225	How to configure the identity and system portlets used in the Identity Manager user interface
Part V, “Designing and Managing Provisioning Requests,” on page 293	How to configure, deploy, and manage the resources, workflows, and request definitions required for provisioning with the Provisioning Module for Identity Manager **NOTE:** This part applies only if you have the Provisioning Module for Identity Manager.
Part VI, “Appendixes,” on page 345	Additional reference information (schema extensions) and advanced topics (configuring the application archive) for the Identity Manager user application

See also

For other related manuals and readme information, go to the Identity Manager page (http://www.novell.com/idm/) of the Novell documentation Web site.

Overview

These chapters introduce you to the Identity Manager user application, and help you plan for its use in your organization.

Overview

1

The Novell Identity Manager user application is a powerful Web application designed to provide a rich, intuitive, highly configurable, highly administrable user experience atop a sophisticated identity-services framework. When used in conjunction with the Provisioning Module for Identity Manager and Novell Audit, the Identity Manager user application provides a complete, end-to-end provisioning solution that's secure, scalable, and easy to manage.

The user application offers the following Web-based end user features:

- White pages
- Organizational charts
- User search (with ability to save custom search configurations)
- Self-service password management
- Lightweight user administration tools
- Initiation and monitoring of workflows (if Provisioning Module is installed)
- Management of personal and/or team tasks (if Provisioning Module is installed)
- Delegation and proxy capabilities

For the system administrator, the user application offers a rich assortment of configuration and administration capabilities, including:

- An interface that allows setup and management of proxy and delegation rights
- Access to logging tools and customized Crystal Reports
- Wizard-based configuration of workflows (if Provisioning Module is installed)
- Workflow management (if Provisioning Module is installed), including the ability to reassign or terminate workflows in progress
- Eclipse-based Designer for creating custom directory abstraction definitions and relationships

A more complete listing of features and capabilities is shown in the table below.

Feature	Description
Standards-based, browser-agnostic, extensible Web-UI user environment	Administrator can change page layouts, default (home) page, add new pages, and modify overall appearance (themes). The user application is extensible through the addition of JSR-168 compliant portlets.
Provisioning workflows (with Provisioning Module installed)	The administrator can create tailored workflows for processing provisioning requests. Those workflows can in turn be initiated by end-users who have the appropriate rights.

Feature	Description
Event-driven workflows (with Provisioning Module installed)	In addition to user-initiated workflows, the administrator can configure workflows in such a way that they are fired automatically when specified events occur in the identity vault.
Enhanced White Pages	Display user information alphabetically, geographically, by skill set, and so forth.
Organization Chart	The user application includes an advanced organizational charting portlet that leverages AJAX to give a richly interactive experience.
User Search	The user can perform searches on identities and save custom search definitions for later reuse.
Password Self-Service	The user application allows end users to access password management functions, eliminating Help Desk calls.
Lightweight User Administration	The user application allows end-users who are non-IT-administrators to perform a limited set of identity management chores.
Eclipse-based Designer	System administrators, developers, consultants, and other IT specialists can perform a variety of configuration and other tasks quickly and easily with the Designer application. For example, the Designer allows one to work offline with entity definitions and relationships, driver policies and filters, and a variety of driver and driver-set configuration tasks. Changes can be saved in a project and/or uploaded to the identity vault.
Proxy Roles (with Provisioning Module installed)	The user application user interface allows appropriately qualified individuals to define proxy roles for specific users. (A proxy can perform tasks on behalf of another user, with all the rights of the other user.)
Delegation of Tasks (with Provisioning Module installed)	The user interface allows managers (and users with appropriate rights) to set up automatic delegation of tasks to peers based on a given user's unavailability. The delegation is fine-grained in that specific types of tasks can be delegated to different individuals.
Directory abstraction layer	The runtime framework isolates Web application logic from the low-level mechanics of identity vault access and workflow, for a secure, robust directory abstraction architecture. Isolation is achieved via a mediation layer known as the directory abstraction layer (or just abstraction layer).
Access control on all user-facing data	The abstraction layer (which leverages eDirectory's sophisticated Effective Rights model) automatically limits the visibility of identity data and workflows, as well as the user's right to modify data, in a way that's transparent to the user and transparent even to the portlets themselves.

Feature	Description
End-user Identity Data Verification	The user application provides a means for users to view and validate/update their own identity information, as it is represented within the identity vault.
Flexible logging	Easily log a wide variety of events to a server log (via log4j) or to Novell Audit, or both.
Novell Audit Reports	The product includes pre-templated Crystal Reports that reflect common reporting tasks relating to provisioning.
High availability	The user application and approval flow elements of the product can be clustered for scalability. **IMPORTANT:** In this version of the Provisioning Module, automatic fail-over of in-process workflow instances is not supported. An in-process flow that has been interrupted can, however, be continued to completion on remaining server nodes with a manual intervention step.
E-mail template management UI	Associate and customize e-mail templates for workflows, using iManager.
Accessory portlets	A variety of ready-to-use portlets come with the user application, including portlets for GroupWise, Exchange, Lotus Notes, Web-mail, Network File, NetStorage, HTML, Shortcut, RSS, and Message portlets.

These features are in addition to the standard functionalities offered by Identity Manager. See the *Identity Manager Administrator's Guide* for more information on the product's standard feature set.

1.1 Types of Roles Supported

The Identity Manager user application encompasses a broad set of identity-management capabilities. Not every user will need to use (nor be able to see) every type of capability; the capability will depend on the person's role.

Users are assumed to fall into one or more of the following categories, each served by different tools and features. (The following vocabulary will be used throughout this documentation.)

1.1.1 LDAP administrator

The LDAP administrator is the person who has maximum configuration and system-administration rights with respect to the identity vault (eDirectory 8.7.x or 8.8). This is a logical role that may be shared also by the User Application Administrator (below), which is the person or entity with system rights to the application server (JBoss), the database (for example, MySQL), and/or the portal-based Web UI itself.

The LDAP administrator can choose from two kinds of tools to accomplish his job: The Eclipse-based Designer for Identity Manager infrequent (possibly one-time) tasks, and iManager tools for daily administration tasks.

Infrequent tasks that you would typically do in Designer Designer for Identity Manager include:

- Configuring the abstraction layer definitions, attributes, and relationships that can be used in the Identity Manager user application. (See the chapter on Chapter 4, “Configuring the Directory Abstraction Layer,” on page 75 for more information.)
- Validating directory abstraction layer definitions. (See the chapter on Chapter 4, “Configuring the Directory Abstraction Layer,” on page 75.)
- Making changes to User Application Driver settings. (See the chapter on Chapter 3, “Configuring the User Application Driver,” on page 59.)
- Localizing the display text for entity and attribute display labels; org chart relationship names; and global and local list items. (See the chapter on Chapter 4, “Configuring the Directory Abstraction Layer,” on page 75.)
- Import or export the User Application Driver and its settings.
- Other kinds of offline tasks.

Everyday tasks in which the administrator (whether it’s the LDAP administrator or the User Application Administrator, described below) is typically operating on a live system are done in iManager. Such tasks might include:

- Managing e-mail templates.
- Defining or designating provisioned resources and provisioning request definitions.
- Enabling or disabling a workflow definition, thereby making it active or not.
- Terminating an in-process workflow.
- Running reports on Novell Audit logging data.

Some of these tasks (the workflow-related ones) apply only when the Provisioning Module has been installed. Also, many of them might be done by the User Application Administrator (below) rather than the LDAP administrator.

1.1.2 User Application Administrator

The User Application Administrator performs tasks associated with administering the Web application (the browsing-based application running on JBoss). Access to the administration tools for this role occurs via the Administration tab of the Identity Manager user interface.

Actions that you might carry out in the user application include:

- Configuring various application settings, such as those that tell the user application how to connect to the identity vault (LDAP provider). For details, see Chapter 10, “Portal Configuration,” on page 189.
- Determining the pages displayed in the Identity Manager user interface and who has permission to access them. (See Chapter 7, “Page Administration,” on page 131.)
- Determining the portlets available in the Identity Manager user interface and who has permission to access them. (See Chapter 9, “Portlet Administration,” on page 171.)
- Determining the look and feel of the Identity Manager user interface. (See Chapter 8, “Theme Configuration,” on page 165.)
- Controlling the levels of logging messages you want the Identity Manager user application to generate and which of those messages (if any) are sent to Novell Audit. (See Chapter 12, “Logging Configuration,” on page 201.)

- Managing various caches maintained by the Identity Manager user application. (See Chapter 13, “Caching Configuration,” on page 207.)
- Exporting or importing Web content (pages and portlets) used in the Identity Manager user application. (See Chapter 14, “Tools for Exporting and Importing Portal Data,” on page 217.)
- Setting up proxy rights for particular individuals.
- Many other tasks related to the user interface that the end user sees.

Tasks that you would perform in iManager include:

- Managing e-mail templates.
- Defining or designating provisioned resources and provisioning request definitions.
- Enabling or disabling a workflow definition, thereby making it active or not.
- Terminating an in-process workflow.
- Running reports on Novell Audit logging data.

Some of these tasks (the workflow-related ones) apply only when the Provisioning Module has been installed.

1.1.3 End User

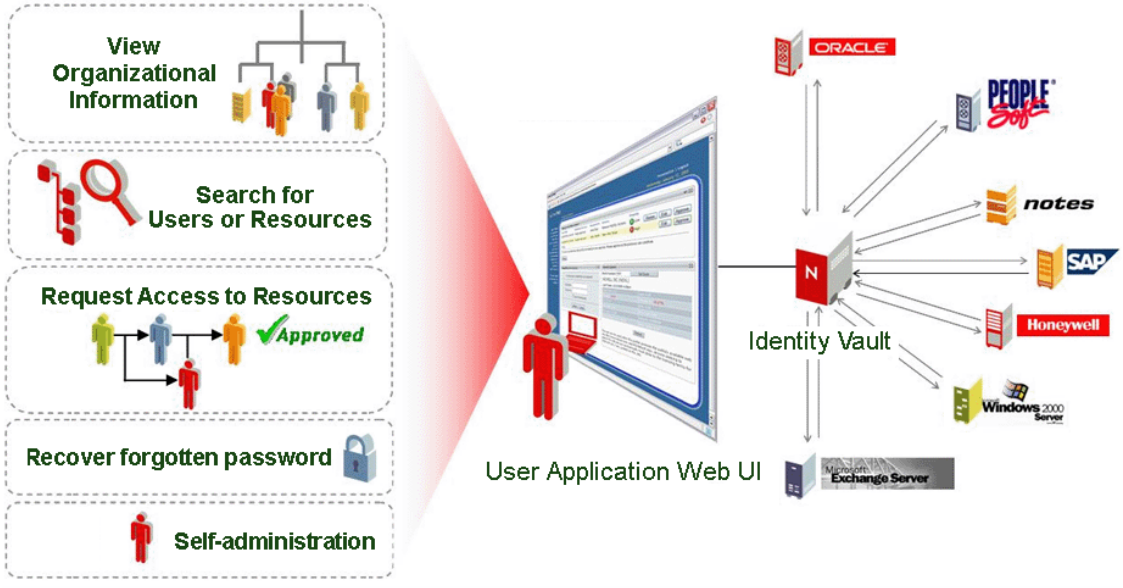
The end user is the person who views and interacts with the various portlets and Web pages that together comprise the user application’s user interface. In this context, end user is intended to mean an employee, a manager, or a proxy or delegate for an employee or manager.

The end user has a potentially vast array of capabilities, depending on how many features are enabled by the administrator. At a minimum, end users will be able to use the Identity Manager user application to:

- View hierarchical relationships between User objects using the Org Chart portlet.
- View and edit user information (with appropriate rights).
- Search for users or resources using advanced search criteria (which can be saved for later reuse).
- Recover forgotten passwords.
- Send e-mail to team members (individually or en masse).

In addition, with the Provisioning Module installed, the user application’s Web interface allows users to:

- Request a resource (start one of potentially many predefined workflows).
- View the status of previous requests.
- Claim tasks and view tasklists (by resource, recipient, or other characteristics).
- View proxy assignments.
- View delegate assignments.
- Specify one’s (un)availability.
- Enter proxy mode in order to claim tasks on behalf of another.
- View team tasks, request team resources, and so forth (managers only).



1.1.4 Delegate user

A delegate user or delegate is an end user to whom one or more specific tasks (appropriate to that user's rights) can be delegated, so that the delegates can work on those specific tasks on behalf of another. For example, John is going on vacation and wants Mary to handle his tasks while he is away. Assuming Mary has rights appropriate to the task (or tasks) John is delegating, Mary can become John's delegates. When John marks himself unavailable in the user application, any tasks that would normally show up in John's task list will show up in Mary's task list instead. Mary thus acts in the role of delegate user. She can claim a task of John's as fully hers (it is no longer John's). Contrast this with the definition of a proxy user, below.

Notice that delegation occurs on a task-by-task basis. It is not necessarily an all-or-nothing transfer of responsibility (although in actual fact, the user interface does allow for a global delegation of all of a user's tasks to a particular delegate, if that's what's called for). A given user may designate more than one delegate. Each delegate can take responsibility only for the task(s) he or she has been given. (For example, John may want Mary to handle any incoming new business card request tasks, but he may want Bill to handle new Siebel account requests.) The transfer of responsibility—the reassignment of new tasks—happens automatically when the original owner of the task declares himself (or herself) unavailable for a particular kind of task. (The declarer can optionally specify an expiration period for the delegation, again on a per-task basis.) This transfer is logged for compliance reasons.

A detailed description of the user interface features for delegate users can be found in Chapter 1 of the *Identity Manager User Application: User's Guide*. See also Section 21.3, "Provisioning security," on page 305 in this guide.

1.1.5 Proxy user

A proxy user is an end user who acts in the role of another user by temporarily assuming that user's identity. All of the rights of the original user apply to the proxy. Work owned by the original user continues to be owned by that user. For example, while John is on a trip to China he wants his administrative assistant, Clive, to be able to access and act on all of his (John's) tasks. John, if he has

appropriate rights, can designate Clive as his proxy. (If he does not have appropriate rights, the User Application Administrator will set this up.) Once the proxy relationship is established, Clive can act in two roles: the role of Clive, or the role of John. In the John role, he can do anything John can do. When work items are accomplished by Clive, it is as if John did them himself.

Notice that, in contrast to the delegation mechanism described in the previous section, a proxy relationship gives the proxy user total visibility into (and the power to act on) the original user's tasks and settings. Also, any attributes or relationships or system settings that John has access to will be accessible by his proxy, for the duration of the proxy's role.

Another distinction between a delegate and a proxy is that whereas a user might delegate some tasks to one delegate and another category of task to another delegate, a proxy always gets all of the tasks of the original user. In other words, when you name someone as your proxy, you can be assured that all of your tasks can be seen and worked by that one individual. It is as if that individual becomes you.

Note that proxy actions undertaken on behalf of another user are logged to Novell Audit as such (for demonstrating compliance).

Additional information on proxy scenarios can be found in "Configuring Your Provisioning Settings" of the *Identity Manager User Application: User's Guide*.

1.2 Data abstraction: The key to flexible identity management

A concept that's key to understanding the Identity Manager user application is that of data abstraction, or being able to define, view, and manipulate instances of directory abstraction layer definitions.

Traditional storage technology, whether it involves relational databases, X.500 directories, or other repositories, typically requires that data entries (rows in a database, objects in an X.500 directory, and so forth) conform strictly to a well-defined schema. Queries over the stored data can be arbitrarily complex (in theory) and the data may include indexes and/or backlinks, but the actual data entries themselves are expected to conform to a fixed definition. Moreover, it's assumed that the applicable schema(s) will not change markedly, if at all, over time.

This is a problem when information (possibly from disparate data sources, relying on disparate schemas) needs to be brought together to create composite data objects conforming to arbitrary new (and possibly transient) schema. Identity data is a classic example, because identities tend to be compositional and non-static. Moreover, the pieces of data on which a given identity is based can come from different sources, each of which might have administrators who are (understandably) protective of the information.

The distributed nature of identity data poses identity-management challenges that can be hard to solve in the face of rigid (and politically bound) schema definitions. One way to attack the problem is to bring together identity data in a logical vault (implemented as a directory) and assemble logical identities from the source data as needed, according to one or more logical schemas that map traditional LDAP objects and attributes (for example) to arbitrary abstraction layer definitions and attributes. In this way, identity data becomes highly compositional and dynamic. Changing the definition of an identity does not require making changes to an LDAP schema. Identity objects can be redefined at will, to suit particular applications or even particular users of particular applications.

This overall approach is often referred to as data abstraction, meaning that identities are materialized as needed, in the form needed.

Abstraction of identity data has a number of advantages:

- It's possible to avoid disruptive, potentially risky changes to LDAP-directory schemas
- Abstraction technology is non-intrusive, requiring no changes to connected systems
- New relationships between data are possible
- The abstraction layer definition(s) can be changed or extended at any time
- Objects can have as many or as few attributes as needed
- Attributes from unrelated LDAP objectclasses can be merged in an abstraction layer definition
- Arbitrary names can be used for attribute naming (there's no requirement to use LDAP names)
- Fine-grained access control policy is still applicable (users see only the data they have the rights to see)
- Complex searches can be performed against new object types (or attribute combinations) which might otherwise be impossible in a pure-LDAP environment

Identity Manager leverages abstraction to achieve all of these goals and more.

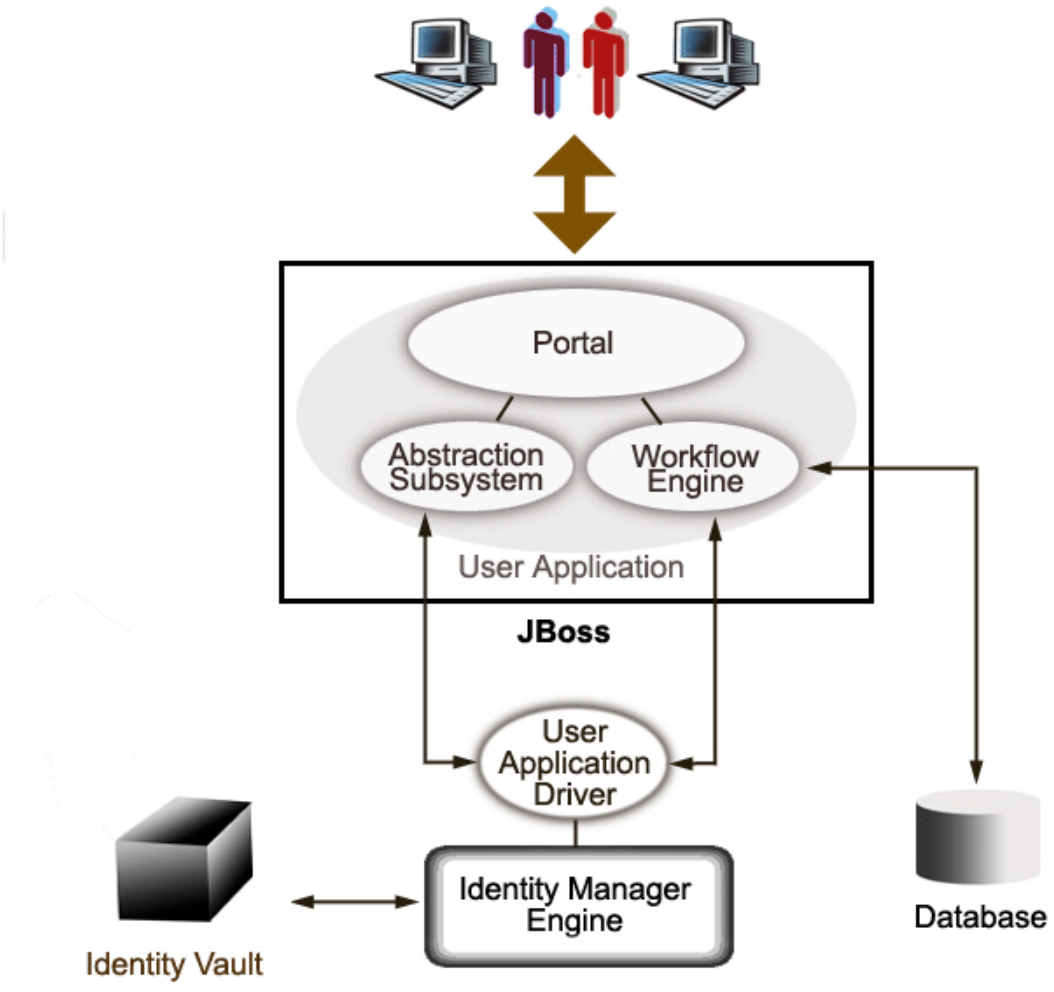
1.3 High-Level Architectural Overview

The Identity Manager user application relies on a number of independent components acting together. The core components, and their fundamental responsibilities, are described in the following table.

Component	Description
Identity vault (eDirectory 8.7.3 or 8.8)	Repository for user data (and other identity data) plus IDM driver set and drivers, as well as various abstraction layer artifacts and (if the Provisioning Module is installed) workflow artifacts.
Identity Manager engine	This is the Identity Manager runtime framework that monitors events in eDirectory (and connected systems), enforcing policies and routing data to and from the identity vault.
User Application Driver	The User Application Driver communicates with the user application to let the latter refresh its cache when the abstraction layer's definitions have changed. When the Provisioning Module is installed, the User Application Driver also can be configured to allow events in the identity vault trigger workflows. It also communicates entitlement information back to the identity vault so that there is a record of the entitlement having been granted (or not) when the workflow is complete.
User Application: Web UI	The user application's Web UI is a browser-based Java application into which JSR 168-compliant portlets plug.
User Application: Abstraction layer	The abstraction layer isolates presentation-layer logic from the identity vault, so that all requests for identity data have to go through the abstraction layer. Portlet code cannot obtain direct access to identity information. All requests go through the abstraction layer and are subject to its constraints (on access control, for example).

Component	Description
User Application: Workflow Engine (available with Provisioning Module only)	The Workflow Engine is a set of Java executables responsible for managing and executing steps in an administrator-defined workflow.
JBoss application server	The open-source JBoss application server provides the runtime framework in which the user application, abstraction layer, and Workflow Engine execute.
Database (MySQL by default)	The database (see the Installation Guide for a list of supported databases) stores certain kinds of configuration information on behalf of the user application, as well as workflow state (if the Provisioning Module is installed).
Composer service driver	The Composer service driver is the portion of the User Application Driver that can be custom-configured to respond to identity vault events by firing workflows.
Novell Audit	Novell Audit is an independent logging server that can persist a variety of kinds of data (such as data generated by steps of a workflow). For more information, see the chapter on setting up logging, later in this book.

In terms of information flow, the above-mentioned components are logically linked in the manner depicted in the diagram below. Physically, the individual components may be (and in most cases will be) located on more than one machine. For example, although the identity vault (and its main administration tool, iManager) will be collocated on the machine that hosts the Identity Manager engine, JBoss (and the user application) will typically be hosted on a separate machine (or group of machines, if clustered). Likewise, for reasons not only of performance but also for security and disaster recovery, the database (MySQL) will typically be on its own machine.



1.3.1 Identity vault

The identity vault is used to store identity data and abstraction layer definitions of various kinds. An instance of eDirectory (running on Windows, Solaris, or Linux) is used for this purpose. By using eDirectory, Identity Manager is able to leverage a well-proven, massively scalable enterprise-class LDAPv3 directory with partitioning and replication capabilities, plus a flexible Web-based management and configuration tool (iManager) which offers an all-in-one administrative integration point between Identity Manager and eDirectory itself.

1.3.2 JBoss

The user application is packaged as a Java Web Application Archive, or WAR file. The WAR is deployed into JBoss, the popular open-source Java application server (which uses Tomcat as its servlet engine; not shown in the diagram). The use of JBoss as an execution environment brings many advantages, including the following:

- The source code is freely available.
- Starting with version 4.0.3, JBoss is clusterable.

- JBoss is fully J2EE compliant, which means any J2EE application can run on it. You can host additional applications (for example, Web Services) on the same instance of JBoss that the user application runs on.
- JBoss supports standard JAAS and JACC Java security and authorization services (which the user application relies on for identity vault access).
- JBoss runs on many different platforms, including popular versions of Windows and Linux.

The user application WAR contains executable code for the user application, which in turn is built using a Model-View-Controller (MVC) architecture, for separation of concerns. The user-facing interfaces run as modular portlets within the user application. Separate portlets exist for viewing org charts, conducting searches, viewing user details, resetting passwords, and so forth.

For more information on the various aspects of deploying Web applications to JBoss, consult the JBoss documentation at http://www.jboss.org/products/jbossas/docs (http://www.jboss.org/products/jbossas/docs).

1.3.3 Database

The user application relies on a database (MySQL by default; see the Installation Guide for a list of supported databases) to store several kinds of information:

- User application configuration data: for example, Web page definitions, portlet instance registrations, and preference values.
- If the Provisioning Module is installed, workflow state information is persisted in the database. (The actual workflow definitions are stored in the identity vault.)
- Novell Audit logs

1.3.4 Identity Manager Engine

The Identity Manager product consists of a runtime engine, drivers, and policies. The Identity Manager engine responds to events in the identity vault and manages the flow and transformation of data to and from the vault. Driver objects encapsulate executable code and artifacts (such as policy documents) designed to provide data-handling behaviors specific to a particular connected system. The Identity Manager user application is a connected system. Communication between the identity vault, the user application's abstraction layer and Workflow Engine occurs via the User Application Driver (see below).

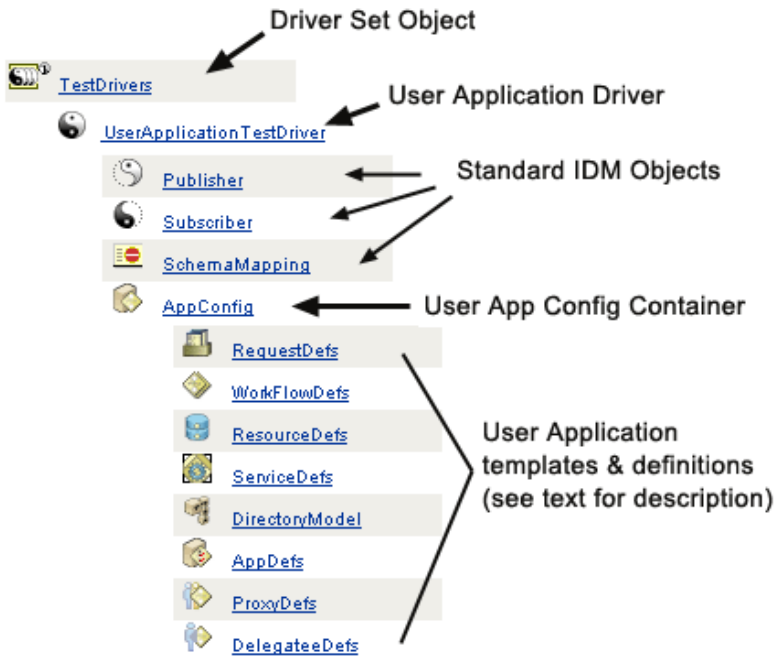
Because the user application relies on various directory objects for storage of abstraction layer artifacts, it's necessary to extend the eDirectory schema to accommodate the custom LDAP objects and attributes required by the user application. Extension of the schema occurs automatically as part of the Identity Manager installation process. Population of custom objects and attributes with default values does not occur, however, until the User Application Driver is installed and activated.

1.3.5 User Application Driver

The User Application Driver is an important enabling piece of the user application. One of the responsibilities of the User Application Driver is to notify the abstraction layer when important data values change in the identity vault, so that the abstraction layer knows to update its cache.

If the Provisioning Module is installed, the User Application Driver can be configured to kick off workflows automatically in response to changes of attribute values in the identity vault.

The User Application Driver is not only a runtime component but a storage wrapper for directory objects (comprising the user application's runtime artifacts). A typical representation of the directory artifacts associated with the User Application Driver is shown below.



NOTE: The names shown represent LDAP common-name (cn) values. The actual schema naming of the various objectclasses is discussed elsewhere.

These artifact categories are discussed in greater detail below.

Driver Set Object

Every Identity Manager installation requires that drivers be grouped into driver sets. Only one driver set can be active at a time (on a given directory server). The drivers within that set can be toggled on or off individually without affecting the driver set as a whole. The User Application Driver (like any other IDM driver) must exist inside a driver set. The driver set is not automatically created by the user application; you must create one ahead of time and then create the User Application Driver within it.

User Application Driver

The User Application Driver object (which can be given any arbitrary name) is the container for a variety of artifacts. As with all Identity Manager drivers, the User Application Driver implements Publisher and Subscriber channel objects and policies. The Publisher channel is not used by the user application, although it is available for custom use cases.

App Config Object

The AppConfig object is a container for various user application configuration objects:

RequestDefs

This is a container for Provisioning Request Definitions, the administrator-configured request definitions available to the user application runtime (if the Provisioning Module exists). The definitions stored here (as XML) represent the classes of requests that end users with appropriate rights can instantiate via the user application. The RequestDef associates a WorkFlowDef (below) with a ResourceDef.

WorkFlowDefs

A container for Workflow objects, including design-time descriptions plus any template or unused flows.

ResourceDefs

A container for Provisioned Resource definitions, including design-time descriptions plus any templates or unused targets.

ServiceDefs

A container for Service Definition objects, which wrap Web Services called by Workflows.

DirectoryModel

Abstraction layer meta-level objects (ChoiceDefs, EntityDefs, RelationshipDefs), which represent different types of content (some user-definable, others administrator-set) of the directory that can be exposed by the Identity Portlets.

AppDefs

A container for configuration objects used to initialize the runtime environment, such as cache config information and e-mail notification properties.

ProxyDefs

A container for proxy definitions.

DelegateeDefs

A container for delegate definitions.

1.3.6 Directory Abstraction Layer

Portlets get their identity data via queries into the directory abstraction layer, which is a code layer that isolates details of identity-data access from client processes. When a portlet needs to perform a search on identity data, for example, the abstraction layer makes the appropriate LDAP queries against the target container in the identity vault, on behalf of the portlet. At no time does any portlet make direct queries into the identity vault.

The abstraction layer is also the code layer through which abstraction layer definitions, as specified by administrators or other qualified users of the system, are created or changed. To make such changes, the system expert uses the Designer application's directory abstraction layer editor, which is described later in this guide, in Chapter 4, "Configuring the Directory Abstraction Layer," on page 75.

At runtime, the abstraction layer caches a wide variety of configuration and entity-definition data obtained from the identity vault. The various caches maintained by the user application can be managed at a detailed level by the administrator. For additional information on caches and cache management, see Chapter 13, “Caching Configuration,” on page 207.

1.3.7 Workflow engine

The workflow engine (available with the Provisioning Module) is the set of runtime classes responsible for executing the steps of a workflow as specified by a process definition (a runtime artifact created when a workflow is instantiated) and keeping track of state information, which is persisted in a database, such as MySQL or Oracle; see Section 1.3.3, “Database,” on page 29, above.

Additional details about the Workflow System, including how to create workflows, can be found in the chapter called Chapter 21, “Introduction to Workflow-Based Provisioning,” on page 295, later in this guide.

1.3.8 User interface

The Identity Manager user interface is comprised of a collection of JSR168-compliant portlets (and, in the case of the Provisioning Module, some Java Server Pages) that run within a Java Web application on JBoss. The portlet architecture provides for a high degree of modularity, content customization, and user control over page appearances. The user application framework provides container services of various sorts. It manages window state, portlet preferences, persistence, caching, theming, logging, and so forth, and acts as a security gatekeeper. The application server on which the user application runs in turn provides various services to the application as a whole, such as scalability through clustering, database access via JDBC, and support for certificate-based security.

The high degree of encapsulation afforded by this architecture provides for a robust and secure presentation-tier environment for the Identity Manager user application. It also guarantees a high degree of administrative control over all aspects of the user interface.

For more information about administration of the various pieces of the user interface, consult the various chapters in this guide under Part III, “Administering the User Application,” on page 123.

1.4 Design and Configuration Tools

Various Identity Manager user application functionalities can be customized or custom-configured through the use of the Identity Manager Designer tool (which is based on the Eclipse Rich Client Platform) or via iManager plug-ins.

The available tools and their intended usages are described in the following table.

Tool	Purpose
Designer for Identity Manager	General configuration tool for Identity Manager, allowing the developer, consultant, or system administrator to make detailed configuration changes to driver sets, drivers, policy definitions, and other artifacts.

Tool	Purpose
Directory abstraction layer Editor plug-in for Designer	Allows you to define custom objects and relationships, and make changes to various configuration settings of the abstraction layer. See Chapter 4, “Configuring the Directory Abstraction Layer,” on page 75, later in this guide.
Provisioning Request Configuration plug-in	Allow the definition and configuration of available provisioning request types (in iManager)
Provisioned Resource Editor (available soon)	Designer plug-in to allow creation and configuration of resources (objects that represent the resource that will be granted in response to a workflow)
Workflow Definition Editor (available soon)	Graphical workflow definition plug-in for Designer
Workflow E-mail Templates Editor	An iManager plug-in allowing administrators to add, delete, and edit e-mail templates. These templates may be used by the workflow system to notify users of workflow events.
lreport.exe (log report tool) and iManager Auditing and Logging feature	A number of predefined log reports (that come with Identity Manager) are available in Crystal Reports (.rpt) format for filtering data logged to the Novell Audit database. The **lreport.exe** log report tool (Windows only) is one way to generate the reports. You can also use other methods to create the reports; see Chapter 5, “Setting up Logging,” on page 115 for details.

A system design expert would typically begin by using the directory abstraction layer editor (in Designer for Identity Manager) to set up custom abstraction layer definitions for the user application. These objects then become available for use by the abstraction layer (and therefore users of the user interface). Fine-grained access control settings can be exercised in the definition and use of these objects so that the administrator and end-users can see and manipulate only those objects (and attributes on the objects) for which they have appropriate rights.

If the Provisioning Module is installed, the system design expert or administrator would use the Provisioning Request Configuration wizards in iManager to define the provisioned resources and workflows that will be available to users of the user application. At the same time, the administrator would also use the e-mail templates editor functionality (in iManager) to define the content of the body of any e-mail notifications that will be sent by the workflow(s). See Chapter 23, “Managing Provisioning Workflows,” on page 331 for more information on this.

After configuring the abstraction layer, provisioning request definitions, audit requirements, and e-mail templates, the administrator would typically perform various configuration operations affecting the user application (involving security, caching, and other functionalities), using the administration features described in Chapter 10, “Portal Configuration,” on page 189. Finally, the administrator would configure individual portlets as necessary, using the interfaces described in the various chapters under Part IV of this guide.

NOTE: The chapter immediately following this one describes some of these tasks in greater detail and should be consulted prior to implementing a production environment.

1.5 Usage Scenarios

The functionalities available in the Identity Manager user application are numerous. A few examples will give some insight into the ways in which the user application can be used to solve real-world problems.

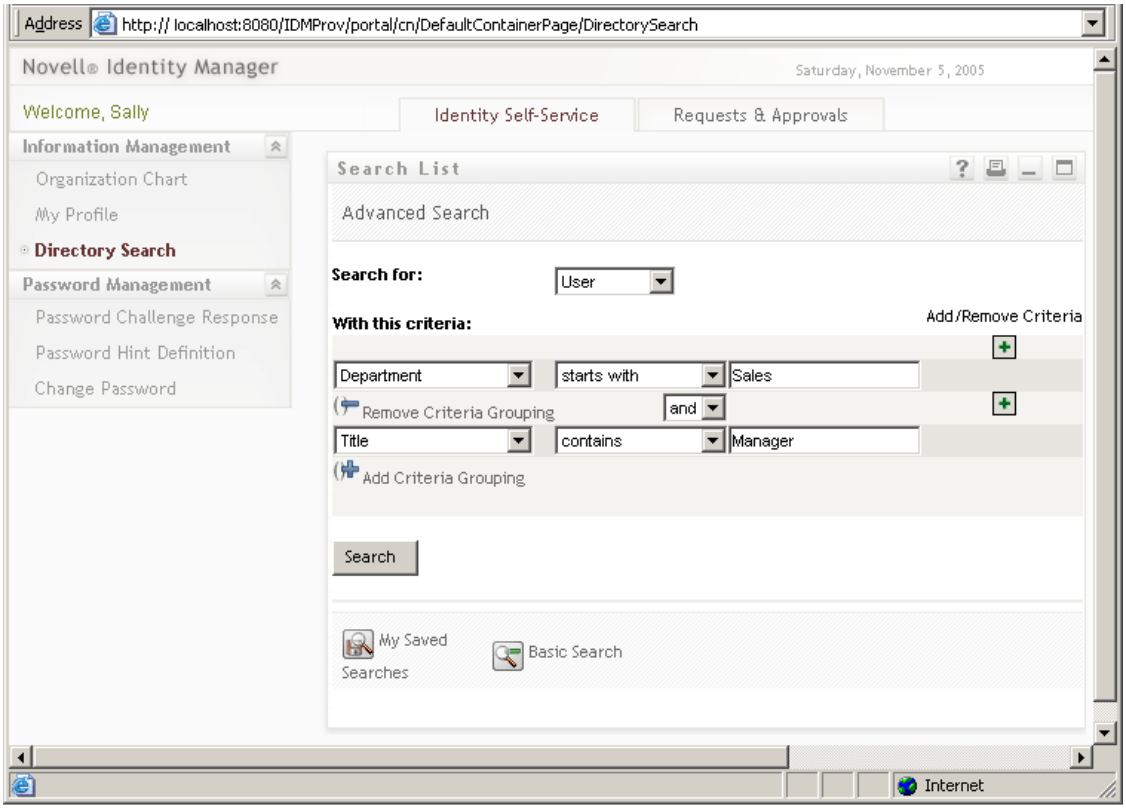
1.5.1 Scenario A: User searches for information about other persons in the organization

A common use case is that an employee wishes to find out information about another person in the organization. For example:

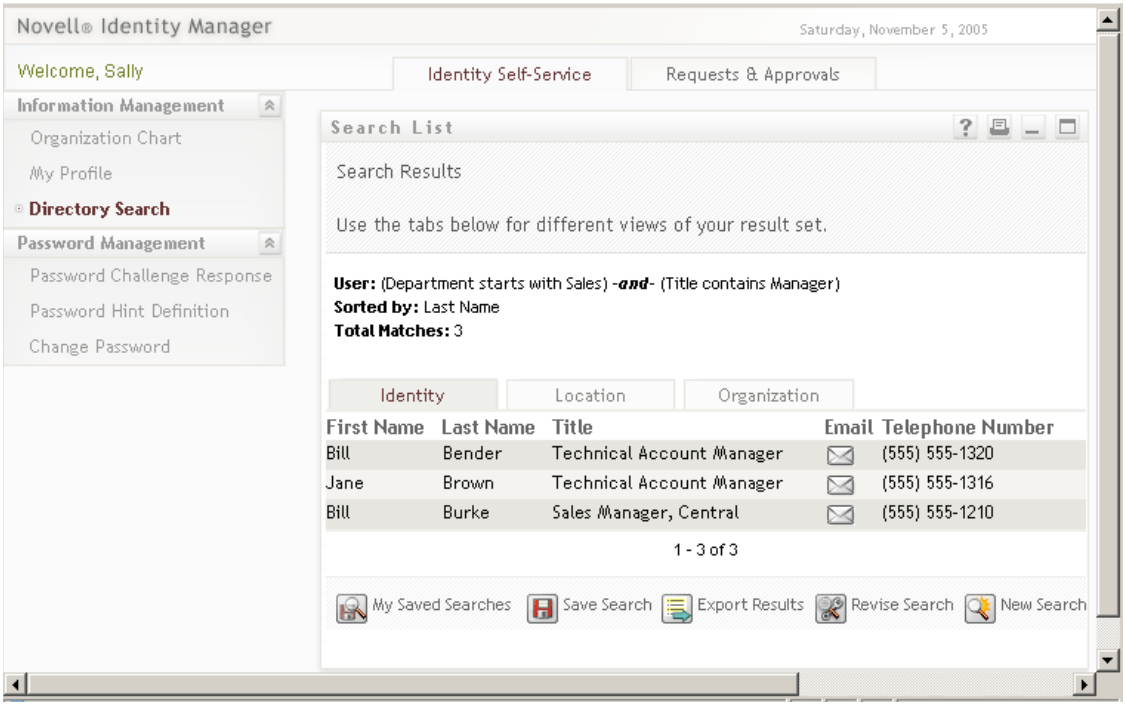
- Obtain a coworker’s full name, contact information
- Find all people with a certain skill set, within a geographic area
- Determine who a particular person's manager is

These kinds of operations (including more advanced searches based on complex queries) can be done easily via the Directory Search interface. Typically, the end-user would log into the user application and bring the Identity Self-Service tab to the front (if it isn’t already on top), then click the Directory Search link in the column of navigation links on the left.

In the screen below, the logged-in user has set up an advanced search to find any User(s) whose department starts with `Sales` and whose `Title` contains Manager.



When this search completes, it provides a results-screen that looks like this:

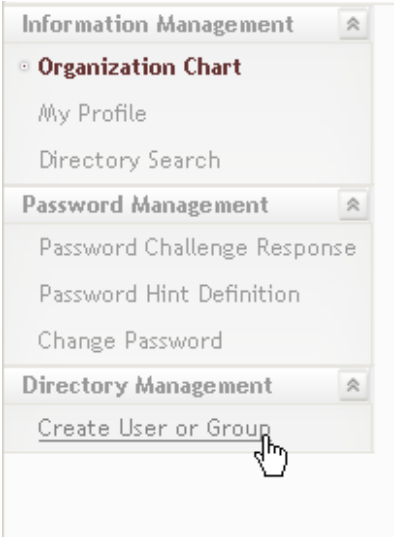


Notice the row of buttons on the bottom, allowing the user to Save this particular advanced query, Revise the query, start over with a new search, and so forth. Notice also the tabs above the list of located individuals. The individuals are currently listed by Identity, but can also be viewed by Location or Organization using the appropriate tab.

1.5.2 Scenario B: Manager creates a new User

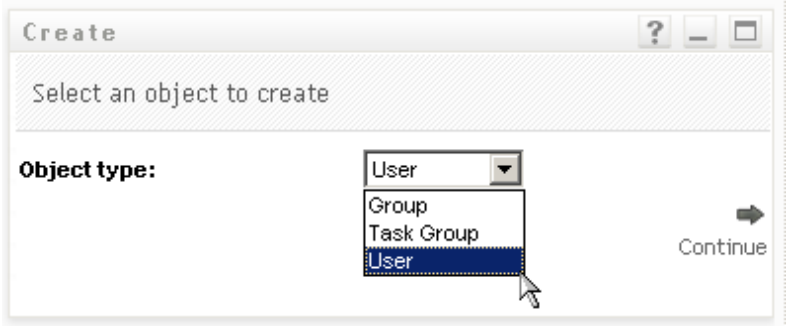
Imagine that a department of a company has taken on a new intern, contractor, or other non-employee (who might only be with the company for a fixed amount of time). The new person needs to be in the system so that he or she can be provisioned with an appropriately limited set of resources (and also so that he or she can be located via User searches of the type described above). Since this person is not a regular employee, the person won't be part of the company's regular Human Resources system. Yet the person's identity (and access to resources) needs to be managed in secure fashion.

As manager of the department in question, you are authorized to enter users into the system. To do this, you log in and find that there is a Create User or Group link in the column of navigation links on the left side of the page (see below):

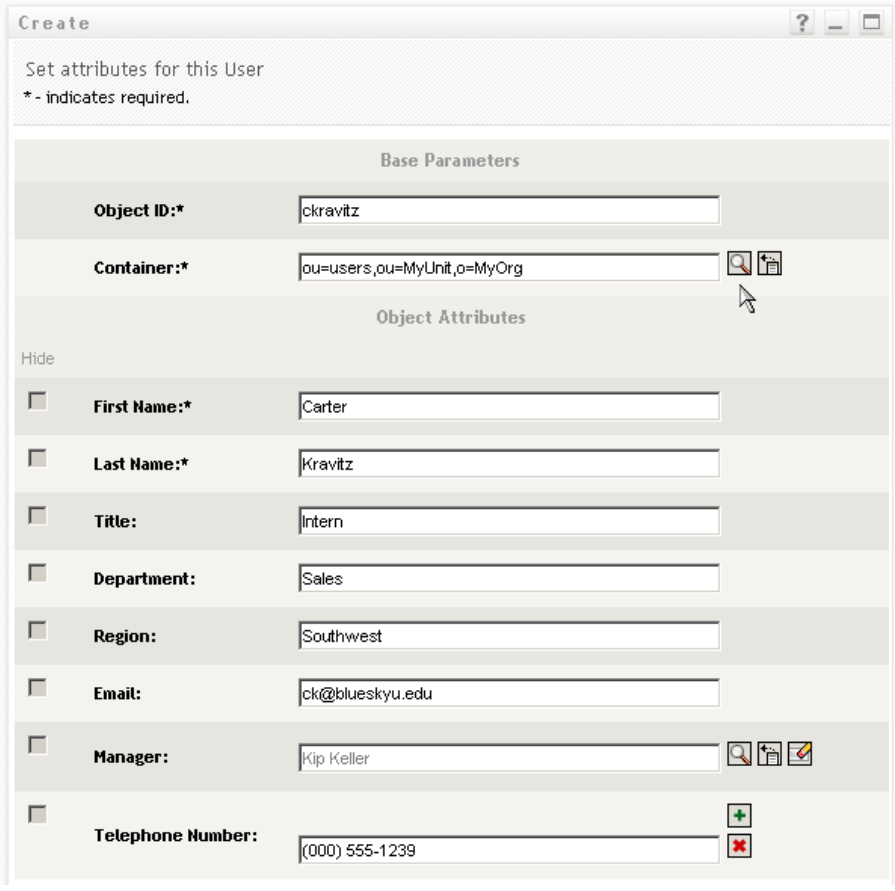


NOTE: This link will not appear unless the logged-in user has appropriate rights.

After clicking this link, you reach a screen that asks you whether you wish to create a new Group, Task Group, or User (as shown below).

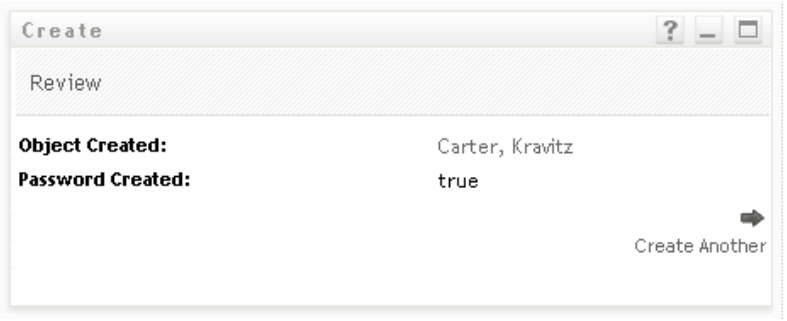


After selecting User and clicking Continue, the next wizard panel allows you to enter this User's personal information:



The next screen allows you to assign a password to the new User:

The final screen shows the net result of the process.



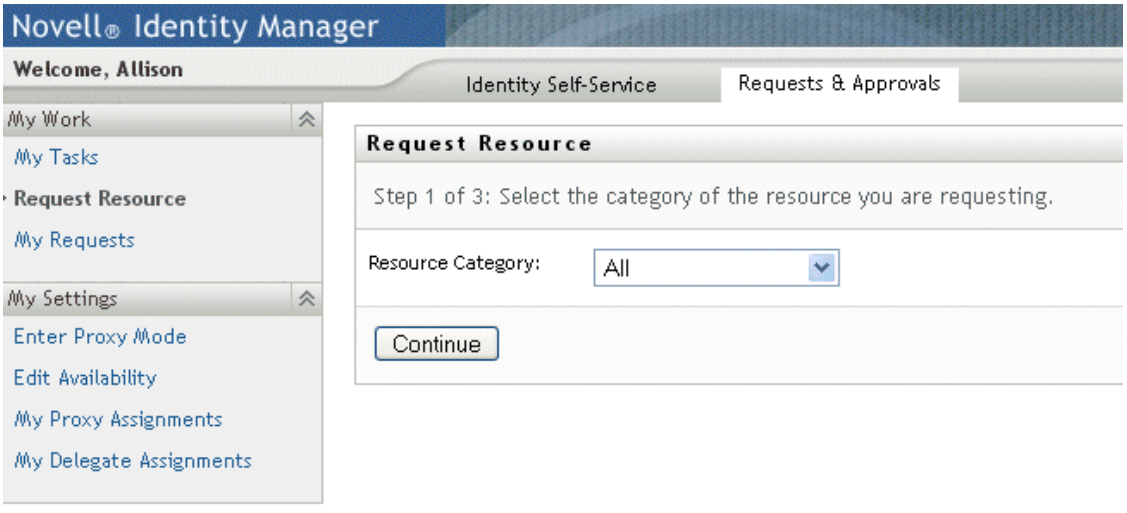
In this example, the newly entered person becomes a User with all of the rights of a normal User. But it is possible to define, say, an Intern object using the directory abstraction layer editor, with unique attributes and rights appropriate strictly to that kind of object. In that case, `Intern` would have shown up as one of the choices in the earlier picklist along with Group, Task Group, and User.

1.5.3 Scenario C: User provisioning

A common situation involves an employee needing to obtain a resource (whether it's a piece of office equipment, a company credit card, or access to a database) that requires approval by another person. This is known as a provisioning request. In Identity Manager, if the Provisioning Module is installed and configured, such requests can be serviced via workflows.

NOTE: Unlike the preceding examples, this example requires that the Provisioning Module be installed and configured.

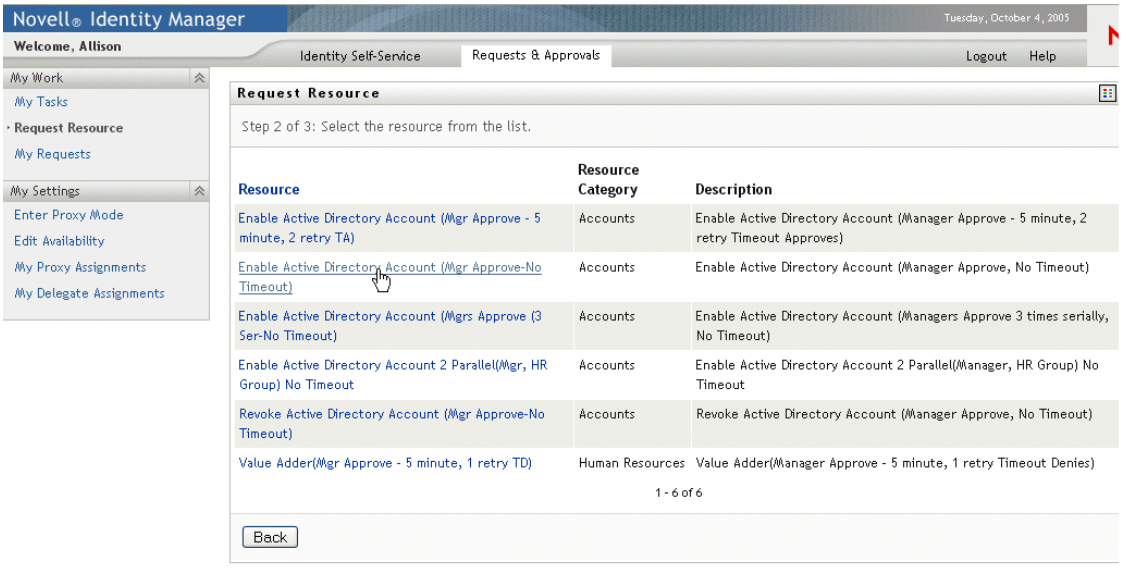
The user would first log in to the user application to arrive at his or her landing page. At the top of the page, the user would click the *Requests & Approvals* tab, then locate the *Request Resource* link on the left-hand navigation frame. When the *Request Resource* link is clicked, the user application displays the initial request form.



The Resource Category dropdown menu might contain any number of resource types, including entitlements with arbitrary names. (See the Identity Manager main Administration Guide for more

information on entitlements and how they are created.) To view all available provisioned resources (in other words, anything that this particular user, with his or her present rights, can request), one need only select `All`, as shown.

If the user clicks Continue, the next screen will display all of the provisioning request types that this user is permitted to access.



In this example, the user wishes to request an Active Directory account, something that requires manager approval. By merely clicking the appropriate link and filling in a simple form, the associated workflow is kicked off and the person's manager receives an e-mail notification pertinent to the task that the manager needs to perform. The manager, in turn, can log in to his *Requests and Approvals* page and find the employee's request waiting in his task list, ready to approve or deny. (If the manager is on vacation, his or her designated proxy will be notified and can log in and take the same actions the manager normally would.) Meanwhile, the browser screen changes to show a summary page that confirms that the workflow request was submitted successfully.

The granting of an account in a company's directory (as shown here) is an example of an entitlement request. Many types of entitlement requests can be configured in the Identity Manager user application, and many kinds of workflows (single- or multi-manager approval, serial flow or parallel flow, with or without time-outs, and so forth) can be created. In all cases, fine-grained access control is available to manage the visibility of workflows and other information.

More information on these features can be found in the last chapters of this guide. (The information in those chapters is mainly of interest to administrators. The usage of the features is described more fully in the Identity Manager User Application User's Guide.)

1.6 Where to go next

If you are ready to learn more about designing a production environment, go to the next chapter (Chapter 2, "Designing the Production Environment," on page 41). Otherwise, you may wish to go directly to one of the later chapters in this book for the following kinds of information:

To learn more about the *logging and audit capabilities* of the user application, see Chapter 5, "Setting up Logging," on page 115.

To learn more about customizing the *look and feel of the user interface*, see Chapter 8, “Theme Configuration,” on page 165.

To learn more about *security* as administered through the user application’s administrative interface (as opposed to iManager), see Chapter 11, “Security Configuration,” on page 197.

To learn more about the user application’s *cache management* facilities, see Chapter 13, “Caching Configuration,” on page 207.

To learn more about *password management* functionality, see Chapter 19, “Password Management Portlets Reference,” on page 267.

To learn more about *portlet administration*, see Chapter 9, “Portlet Administration,” on page 171.

To learn about import and export of portal data, see Chapter 14, “Tools for Exporting and Importing Portal Data,” on page 217.

To learn more about *organizational-chart* features, see Chapter 18, “Org Chart Portlet Reference,” on page 251.

To learn more about *directory-search* functionality, see Chapter 20, “Search List Portlet Reference,” on page 281.

To learn more about new-object creation (*Create* portlet) options and how they are administered, see Chapter 16, “Create Portlet Reference,” on page 231.

To learn more about *workflow* setup and administration, consult Chapter 21, “Introduction to Workflow-Based Provisioning,” on page 295, as well as Chapter 22, “Configuring Provisioning Request Definitions,” on page 309 and Chapter 23, “Managing Provisioning Workflows,” on page 331.

Designing the Production Environment

2

This chapter discusses issues relating to setting up a production environment. It provides guidance on a number of considerations that will come into play when making the transition from a sandbox/test (or other pre-production) environment to a production environment.

This chapter is organized according to the following major sections:

2.1 Topology

The number of instances of each major subsystem and the ways in which they can be connected are potentially great in number. Not every possible layout is supported. It is important to understand not only the possibilities but why some configurations are preferred over others.

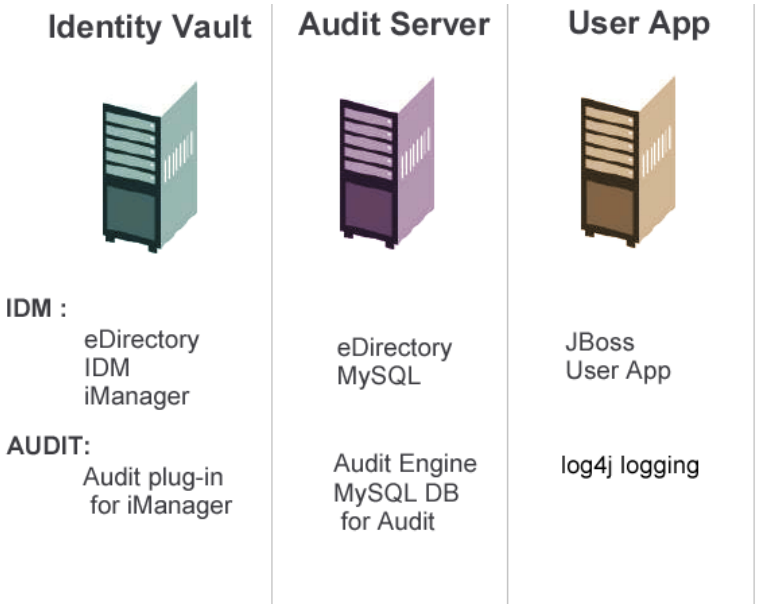
2.1.1 Minimal design

The simplest logical configuration of the user application is a one-of-everything installation, consisting of one identity vault tree, one instance of the Identity Manager engine and drivers, and one instance of JBoss running a single instance of the user application. In terms of physical implementation, you could, in theory, run all of this on one box. But you would not do that in the real world, for a variety of reasons (security, maintainability, and performance chief among them). In deciding on the number of machines needed for a practical real-world installation, you would want (at a minimum) to take the following into account:

- *Novell Audit server:* This piece is responsible for capturing event information (and possibly a good deal of other information) from the user application environment at runtime. It may also be doing double-duty as a persistence store for other applications in your company. For a variety of reasons, you probably will not want to put other major pieces of the Identity Manager system (for example, JBoss or the identity vault) on the same machine as the Audit server.
- *Identity vault:* This is a heavily trafficked component with a need for good performance and good scalability. In all likelihood, you will want to consider having the identity vault exist on a dedicated machine. That is to say, you probably do not want another high-traffic system, such as JBoss with a deployment of the user application, running alongside the identity vault on the same machine.
- *Database:* If this instance of MySQL (or other supported database) is also your Novell Audit database, it will probably be on a dedicated machine. Consider that this piece is used by the user application in the following ways:
- As a persistence store for portal configuration data

- As the persistence store for state information on in-process workflows (if the Provisioning Module is installed)
- Optionally, as the logging store for Novell Audit.
- *JBoss:* For performance and capacity reasons, you will probably want to run this piece on a dedicated machine.

These considerations suggest the following minimal 3-machine configuration:



2.1.2 High availability design

Clustering for high availability/capacity is discussed in detail in a later section of this chapter. For now, you should know that:

- Identity Manager supports high availability of the identity vault, engine, and drivers through the multinode installation and shared-storage mechanisms described in the chapter on "High Availability" in the main Identity Manager Administration Guide. A comprehensive recipe for setting up such a system using SUSE Linux is given in the article at:

  http://support.novell.com/cgi-bin/search/searchtid.cgi?/10093317.htm (http://support.novell.com/cgi-bin/search/searchtid.cgi?/10093317.htm)
- High availability of the user application is available through JBoss clustering. You can set up a JBoss cluster such that each node runs one user application instance. The instances will all be coequals (peers). Nevertheless, there is no session replication across instances. Each instance is responsible for its own unit of work and will not finish a session that was started on a sister node.
- Automatic failover is not supported (for the reasons just stated). But an interrupted workflow can resume again after the loss of a cluster node, if a new node is brought online with the same workflow engine ID as the one that went down. (In this case, resumption of the interrupted workflow occurs automatically, as soon as the new workflow engine starts.)

Again, see Section 2.4, "Clustering," on page 49 (further below) for a more detailed discussion of these issues.

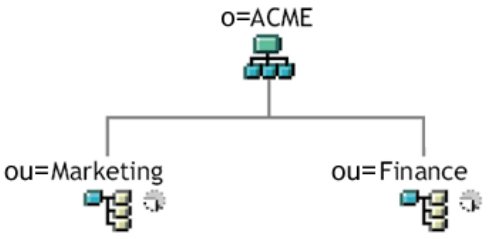
2.1.3 Design constraints

In general, the two most important architectural constraints to be aware of are:

- No user application instance can service (search/query, add users to, and so forth) more than one user container. Also, once a user container has been associated with the application, that association is meant to be permanent.
- No User Application Driver can be associated with more than one user application, except when the user applications are installed on sister nodes of the same JBoss cluster. In other words, a one-to-many mapping of drivers to user applications is not supported.

The first constraint enforces a high degree of encapsulation in user application design.

Suppose you have the following organizational structure:



During installation of the user application, you are asked to specify the top-level user container that your installation will look for in the identity vault. In this case, you could specify *ou=Marketing,o=ACME* or (alternatively) *ou=Finance,o=ACME*. You cannot specify both. All user application searches and queries (and administrator log-ins) will be scoped to whichever container you specify.

NOTE: In theory, you could specify a scope of *o=ACME* in order to encompass Marketing and FInance. But in a large organization, with potentially many *ou* containers (rather than just two relating to Marketing and Finance), this is not likely to be practical.

It is possible, of course, to create two independent installations of the user application (sharing no resources in common), one for Marketing and another for Finance. Each installation would have its own database, its own appropriately configured User Application Driver, and each user application would be administered separately, possibly having unique themes.

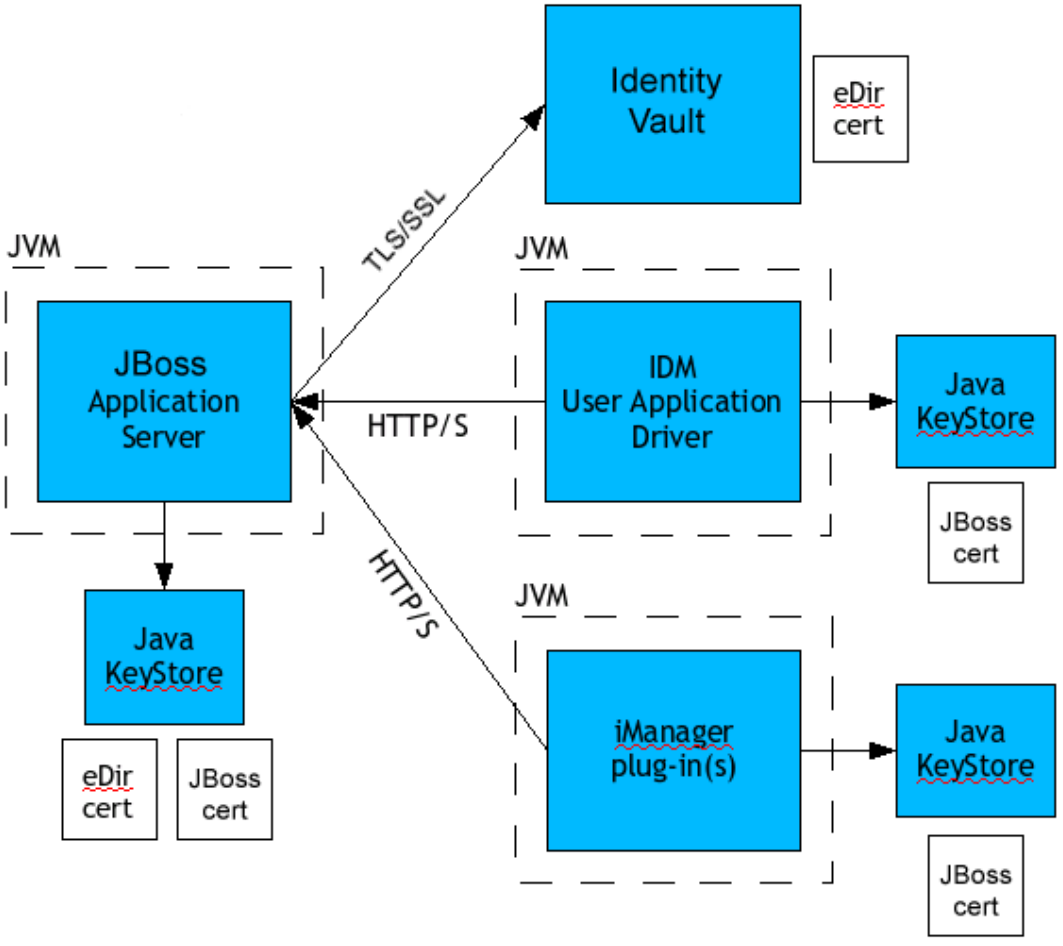
If you truly need to place Marketing and Finance within the same scope for one user application installation, there are two possible tactics to consider. One is to insert a new container object (for example, *ou=MarketingAndFinance*) in the hierarchy, above the two sibling nodes; then point to the new container as the scope root. Another tactic is to create a filtered replica (a special type of eDirectory tree) that combines the needed parts of the original ACME tree, and point the user application at the replica's root container. (Consult the Novell eDirectory Administration Guide for more information on filtered replicas.)

If you have questions about a particular system layout, contact your Novell representative for assistance or advice.

2.2 Security

Moving from pre-production to production usually involves hardening the security aspects of the system. In sandbox testing, you may have been using regular HTTP to connect the user application driver to JBoss, or you may have been using a self-signed certificate (as a temporary measure) for driver/app-server communication. In production, on the other hand, you will probably use secure connections, with server authentication based on your company's Verisign (or other trusted-provider) certificate.

It is typical for X.509 certificates to be used in a variety of places in the Identity Manager user application environment, as shown in the diagram below.



All communication between the user application and the identity vault is secure, using Transport Layer Security, by default. The installation of the identity vault (eDirectory) certificate into the JBoss keystore is done automatically at install time. Unless you specify otherwise, the user application installer places a copy of the eDirectory certificate in the JRE's default *cacerts* store.

The server certificate needs to be in several places, if communications are to be secure, as shown in the diagram. Different setup steps may be needed depending on whether you intend to use a self-signed certificate in the various places in the diagram shown with a *JBoss cert* box, or you intend (instead) to use a certificate issued by a trusted certificate authority (CA) such as Verisign.

Self-Signed Certificates

If you are using a certificate from a well-known trusted issuer (for example, Verisign), no special configuration steps should be necessary. But if you intend to create and use a self-signed certificate, you will need to do these steps:

1. Create a keystore with a self-signed certificate, using command-line syntax similar to the following:

   ```
   keytool -genkey -alias tomcat -keyalg RSA -storepass changeit -
   keystore jboss.jks -dname
   "cn=JBoss,ou=exteNd,o=Novell,l=Waltham,s=MA,c=US" -keypass
   changeit
   ```

   Notice that you are creating the file “jboss.jks” as well as the certificate.

2. Copy the keystore file (jboss.jks) to your JBoss user application directory, for example:

   ```
   cp jboss.jks ~/jboss-4.0.2/server/spitfire/conf
   ```

Turning on SSL in JBoss

To enable SSL in JBoss, locate *jbossweb-tomcat55.sar* file under *[IDM]/jboss/server/IDM/deploy/*. In it, find *server.xml* and open that file in a text editor. Enable SSL by uncommenting or adding a section that looks like:

```
<Connector port="8443" address="${jboss.bind.address}"
     maxThreads="100" strategy="ms" maxHttpHeaderSize="8192"
     emptySessionPath="true"
     scheme="https" secure="true" clientAuth="false"
     keystoreFile="${jboss.server.home.dir}/spitfire/conf/jboss.jks"
     keystorePass="changeit" sslProtocol = "TLS" />
```

Turning on SOAP Security

In *IDM.war*, find the *web.xml* file and open it in a text editor. At the bottom of the file, uncomment the following section:

```
<security-constraint>
        <web-resource-collection>
                <web-resource-name>IDMProv</web-resource-name>
                <url-pattern>/*</url-pattern>
                <http-method>POST</http-method>
                <http-method>GET</http-method>
                <description>IDM Provisioning Edition</description>
                </web-resource-collection>
        <user-data-constraint>
               <transport-guarantee>CONFIDENTIAL</transport guarantee>
        </user-data-constraint>
</security-constraint>
```

Save the file and the archive. Restart JBoss.

2.2.1 Mutual Authentication

The Identity Manager user application supports traditional *server authentication* scenarios (as commonly used in https sessions with *secure web pages* on the Web), but does not support

bidirectional certificate-based authentication out of the box. That functionality can be obtained, however, by using Novell iChain. So if (for example) your organization has a need to allow users to log in via a user certificate, rather than via password, you would be able to achieve this by adding iChain to your environment.

See your Novell representative for more information.

2.3 Performance Tuning

Performance tuning is a complex subject. The Identity Manager user application relies on diverse technologies with many interactions. It is not possible to anticipate every single configuration scenario or user interaction scenario that could result in poor performance. Nevertheless, some subsystems are subject to best practices that can boost performance. These are discussed below.

2.3.1 Logging

The user application allows logging via Novell Audit as well as via the open-source Apache *log4j* framework. Logging via Novell Audit is turned off by default. However, file and console logging via *log4j* are enabled by default.

NOTE: The kinds of events you can log, and how to enable or disable logging, are covered in Chapter 5, “Setting up Logging,” on page 115 and Chapter 12, “Logging Configuration,” on page 201 later in this guide.

The *log4j* configuration settings are contained in a file called *log4j.xml* under *$IDMINSTALL/jboss/server/IDMProv/conf/*. Near the bottom of this file, you will find the following entry:

```
<root>
    <priority value="INFO" />
      <appender-ref ref="CONSOLE" />
      <appender-ref ref="FILE" />
  </root>
```

Assigning a value to the `root` ensures that any log appenders that do not have a level explicitly assigned inherit the `root` level (in this case, INFO). For example, by default, the FILE appender does not have a threshold level assigned and so it assumes the root’s.

The possible log levels used by *log4j* are DEBUG, INFO, WARN, ERROR, and FATAL, as defined in the *org.apache.log4j.Level* class. Inattention to the proper use of these settings can be costly in terms of performance.

A good rule of thumb is to use INFO or DEBUG only when debugging a particular problem.

Any appender included in the root that does have a level-threshold set, should have that threshold set to ERROR, WARN, or FATAL unless (as just explained) you are debugging something.

The performance hit with high log levels has less to do with verbosity of messages than with the simple fact that console and file logging, in *log4j*, involve synchronous writes. An `AsyncAppender` class is available, but its use does not guarantee better performance. The issues (which are well-known and are Apache *log4j* issues, not Identity Manager issues) are set forth at *http://logging.apache.org/log4j/docs/api-1.2.8/org/apache/log4j/performance/Logging.html*.

The default of INFO in the user application's log config file (above) is satisfactory for many environments, but where performance is critical, you should consider changing the above *log4j.xml* entry to:

```
<root>
      <priority value="ERROR"/>
      <appender-ref ref="FILE"/>
</root>
```

In other words, remove CONSOLE and set the log level to ERROR. For a fully tested/debugged production setup, there is no need to log at the INFO level, nor any need to leave CONSOLE logging enabled. The performance payoff of turning these off can be significant.

For more information on *log4j*, consult the documentation available at *http://logging.apache.org/log4j/docs*.

For more information on the use of Novell Audit with Identity Manager, consult the Novell Identity Manager Administration Guide.

2.3.2 Identity Vault

LDAP queries can be a bottleneck in a heavily utilized directory-server environment. To maintain a high level of performance with large numbers of objects, Novell eDirectory (which is the basis of the identity vault in Identity Manager) records frequently requested information and stores it in indexes. When a complex query is run against objects with indexed attributes, the query returns much faster.

Out of the box, eDirectory comes with the following attributes already indexed:

```
Aliased Object Name
cn
dc
Equivalent to Me
extensionInfo
Given Name
GUID
ldapAttributeList
ldapClassList
Member
NLS: Common Certificate
Obituary
Reference
Revision
Surname
uniqueID
uniqueID_SS
```

When you install Identity Manager, the default directory schema is extended with new objectclass types and new attributes pertaining to the user application. User-application-specific attributes are (by default) not indexed. For better performance, you may find it useful to index some of those attributes (and perhaps a few traditional LDAP attributes as well), particularly if your user container will contain over 5,000 objects.

The general idea is to index only those attributes that you know will be regularly queried. (Which could very well be different attributes for different production environments.) The only way to know

for sure which attributes are heavily used is to collect predicate statistics at runtime. (The collection process itself is performance-degrading, however.)

The process for collecting predicate statistics in discussed in detail in the eDirectory Administration Guide. Indexing is also discussed in more detail there. In general, you will need to do the following:

- Use Console One to turn on predicate-statistics collection for attributes of interest
- Put the system under load
- Disable statistics collection and analyze the results
- Create an index for each type of attribute that might benefit from having one

If you already know which attributes you want to index, there is no need to use Console One. You can create and manage indexes in iManager via eDirectory Maintenance > Indexes. For example, if you know that users of your org chart will very likely perform searches based on the *isManager* attribute, you can try indexing that attribute to see if performance is enhanced.

NOTE: As a best practice, it is recommended that you index, at a minimum, the *manager* and *isManager* attributes.

For an in-depth discussion of attribute indexing and performance, see the chapter on "Tuning eDirectory" in *Novell's Guide to Troubleshooting eDirectory* by Peter Kuo and Jim Henderson (QUE Books, ISBN 0-7897-3146-0).

Also see the chapter on "Maintaining Novell eDirectory" (which has performance-tuning guidance) in the main *eDirectory Administration Guide*.

2.3.3 JVM

The amount of heap memory allocated to the Java virtual machine can impact performance. If you specify min or max memory values that are either too low or too high (too high meaning more than the physical memory of the machine), you could experience excessive pagefile swapping.

You can set the max JVM size for the JBoss server by editing the `run.conf` or `run.bat` file (the former for Linux, the latter for Windows) under `[IDM]/jboss/bin/` in a text editor. Increase "-Xmx" from *128m* to *512m*, or possibly higher. Some experimentation may be needed to determine the optimal setting for your particular environment.

NOTE: JBoss and Tomcat performance tuning tips can be found at http://wiki.jboss.org/wiki/Wiki.jsp?page=JBossASTuningSliming (http://wiki.jboss.org/wiki/Wiki.jsp?page=JBossASTuningSliming)

2.3.4 Session Timeout Value

The session timeout (the amount of time a user can leave a page unattended in his or her Web browser before the server causes a session-timeout warning dialog to appear) can be changed in the *web.xml* file in the *IDM.war* archive. This value should be tuned to match the server and usage environment in which the application will run. In general, it is advised that the session timeout be as small as practicable If business requirements can tolerate a 5-minute session timeout, this would allow the server to release unused resources twice as early as it would if the timeout value were 10 minutes. This makes the Web application more performant and scalable.

Please consider the following when adjusting the session timeout:

- Longer session time-outs could potentially cause the JBoss server to run out of memory if many users were to log in over a short period of time. This is true of any application server that has too many open sessions.
- When a user logs in to the user application, an LDAP connection is created for the user and bound to the session.Thus, the more sessions that are open, the greater the number of LDAP connections that are held. The longer the session timeout, the longer these connections are held open.Too many open connections to the LDAP server (even if they are idle) could cause system performance degradation.
- If the server starts experiencing OutOfMemoryErrors, and the JVM heap and garbage collection tuning parameters have already been optimally tuned for the server and usage environments, then lowering the session timeout should be considered.

To adjust the session timeout value, you will need to open the *IDM.war* archive, find the *web.xml* file inside it, and edit the following portion of that file (in particular, the numeric value, shown here as 20, meaning 20 minutes, which is the default):

```
<session-config>
    <session-timeout>20</session-timeout>
</session-config>
```

Then you will need to save the file and the archive, and restart the server.

NOTE: Hand-editing of Web archive files is best done by a person experienced in Java Web application development and deployment.

2.4 Clustering

There are three things that you must consider when using the user application in a cluster environment:

- The JBoss cluster configuration (see Section 2.4.1, “Clustering JBoss,” on page 49)
- The user application caching configuration (see Section 2.4.3, “Configuring the User Application Cluster Group Caching Configuration,” on page 54)
- The workflow engine configuration (see Section 2.4.4, “Configuring Workflows for Clustering,” on page 54)

2.4.1 Clustering JBoss

A cluster is a collection of application server nodes that provide a set of services. The purpose of a cluster is to increase performance and reliability of applications. In general, a cluster provides three key benefits for enterprise applications:

- High availability
- Scalability (more capacity)
- Load balancing

High availability means that an application is reliable and available for a high percentage of the time that it is deployed. Clusters provide high availability because the same application is running on all nodes. If one node fails, the application is still running on other nodes. The Identity Manager user

application benefits from higher availability when running in a cluster. However, the Identity Manager user application does not support HTTP session replication. This means that if a session is in process on a node and that node fails, the session information will be lost.

Load balancing is the practice of distributing the workload among the members of a cluster. The goal of load balancing is to improve performance. Load balancing can be achieved by a variety of means (for example, DNS round robin, hardware load balancing). See http://www.onjava.com/pub/a/onjava/2001/09/26/load.html (http://www.onjava.com/pub/a/onjava/2001/09/26/load.html) for a discussion of various load balancing methods. Regardless of the method selected, you will want to include load balancing in your cluster configuration.

JBoss Cluster Groups

JBoss clusters are based upon a communications module named JGroups. JGroups is installed with JBoss (it also can be used without JBoss). JGroups provides communications among groups, which share a common name, multicast address, and multicast port.

When you install a clustered JBoss server, JBoss defines two different JGroups groups for use in managing the cluster. One is called *DefaultPartition* and is defined in /deploy/cluster-service.xml. This cluster group is used by JBoss to provide core clustering services. JBoss also defines a second cluster group named *Tomcat-Cluster*. This cluster group is defined in /deploy/tc-cluster-service.xml. This cluster group provides session replication for the Tomcat server that runs inside JBoss.

The Identity Manager user application uses a third cluster group. This cluster group uses a UUID name to minimize the risk of conflicts with other cluster groups that users might add to their servers. By default, the cluster group is named `c373e901aba5e8ee9966444553544200`. This cluster isn't configured using a JBoss service file. Instead, the configuration settings are located in the directory and can be configured using the user application administration features. If you are familiar with JGroups and JBoss clustering, you can adjust the user application cluster configuration using this interface. Changes to the cluster configuration only take effect for a server node when that node is restarted.

The user application cluster group is used solely to coordinate user application caches in a clustered environment. It is independent of the two JBoss cluster groups and does not interact with them in any way. By default, the user application cluster group and the two JBoss groups use different group names, multicast addresses and multicast ports so no reconfiguration is necessary.

User application cluster group settings are shared by any Identity Manager 3 application that shares the directory configuration. The purpose of the local settings option in the user application administration interface is to allow an administrator to remove a node from a cluster, or change the membership of servers in a cluster. For example, you can disable clustering globally, then enable it locally for a subset of your servers sharing the directory configuration.

Application Farming

JBoss allows you to hot-deploy across the a cluster by copying an application EAR, WAR, or JAR into the farm directory of one clustered JBoss instance. Hot-deploying on one machine causes that component to be automatically deployed on all instances within the cluster, while the cluster is running.

This form of application deployment is not recommended with the release of JBoss Application Server (4.0.2) that was included with the user application installation program at the time that this document was written, as there are unresolved problems related to its use. However, we have provided the basic steps that you must perform (see “Deploying the User Application to a Cluster

Using JBoss Farming" on page 53) to successfully deploy the user application using JBoss farming technology, as improvements to this technology can be expected after the publication of this document.

MySQL Database

The user application installation program either installs the MySQL database manager and creates a database for use with the user application, or it uses an existing Oracle, Microsoft SQL Server, or MySQL database. The database is responsible for data persistence. All nodes in the JBoss cluster must access the same database instance. The user application uses standard JDBC calls to access and update the database. The user application uses a JDBC data source bound to the JNDI tree to open a connection to the database. If you create the JBoss cluster by using the user application installation program, the data source will be installed for you. If you choose to set up the JBoss cluster manually, you will need to copy the data source file (IDM-ds.xml) to the deploy directory on all nodes in your cluster. Also, if you are using MySQL, you need to copy the MySQL JDBC driver (*mysql-connector-java-3.1.10-utf8-clob-fix-bin.jar*), located in the JBoss /server/IDM/lib directory, to the JBoss server/IDM/lib directory.

Logging

To enable logging for clusters you need to edit the log4j.xml configuration file, located in the \conf directory for the JBoss server configuration (for example, \server\IDM\conf), and uncomment the section at the bottom that looks like this:

```
<!--  Clustering logging
  -->
- <!--
 Uncomment the following to redirect the org.jgroups and
      org.jboss.ha categories to a cluster.log file.
   <appender name="CLUSTER"
class="org.jboss.logging.appender.RollingFileAppender">
     <errorHandler
class="org.jboss.logging.util.OnlyOnceErrorHandler"/>
     <param name="File" value="${jboss.server.home.dir}/log
cluster.log"/>
     <param name="Append" value="false"/>
     <param name="MaxFileSize" value="500KB"/>
     <param name="MaxBackupIndex" value="1"/>
     <layout class="org.apache.log4j.PatternLayout">
       <param name="ConversionPattern" value="%d %-5p [%c] %m%n"/>
     </layout>
   </appender>
   <category name="org.jgroups">
     <priority value="DEBUG" />
     <appender-ref ref="CLUSTER"/>
   </category>
   <category name="org.jboss.ha">
     <priority value="DEBUG" />
     <appender-ref ref="CLUSTER"/>
   </category>
  -->
```

You can find the *cluster.log* file in the *log* directory for the JBoss server configuration (for example, *\server\IDM\log*).

2.4.2 Installing the User Application to a JBoss Cluster

The recommended method of installing the user application to a cluster is to use the user application installation program to install the user application to each node in a cluster. Although we do not recommend deploying the user application to a cluster using JBoss farming, we have included a procedure that you can follow as an alternative method.

Using the User Application Installation Program on Each Node in the Cluster

JBoss comes with three different ready-to-use server configurations: *minimal*, *default* and *all*. Clustering is only enabled in the *all* configuration. A cluster-service.xml file in the /deploy folder describes the configuration for the default cluster partition. When you install the user application and indicate to the installation program that you want to install into a cluster, the installation program makes a copy of the *all* configuration, names the copy *IDM* (by default; the installation program allows you to change the name), and installs the user application into the this configuration.

To install the user application to each node in a cluster using the user application Installation Program:

1. Perform a complete install of the user application (MySQL, JBoss, and the user application) on the first JBoss node. For information about using the user application installation program, see the *Identity Manager 3 Installation Guide*.

   - If you are using MySQL as your database for the user application, the user application installer creates a new installation of MySQL. Make note of the MySQL root user password that you specify; you will need this information when you install the user application on the rest of the nodes in the cluster.
   - In the installation program *IDM Configuration* screen, select the "*clustering (all)*" option.
   - Select other installation options as appropriate for your environment.

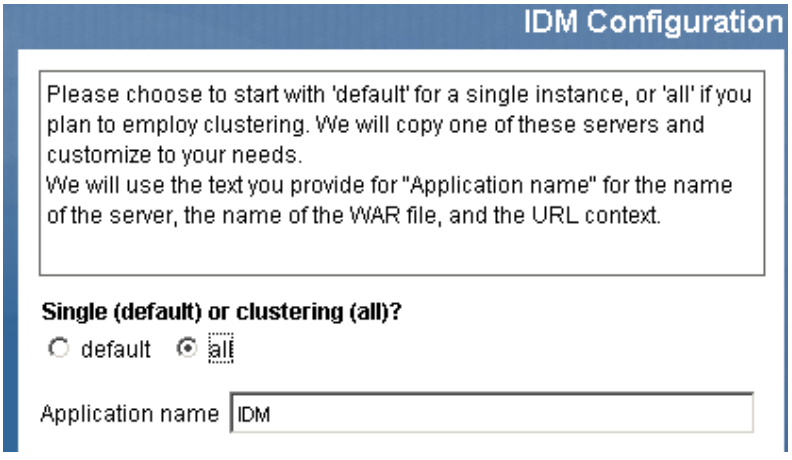
2. If MySQL isn't running already, start MySQL using the *start-mysql.bat* file located in the /IDM/mysql directory.

   NOTE: On Linux, the following shell command will be helpful in determining whether the MySQL daemon is running:

   ```
   ps -A | grep mysqld
   ```

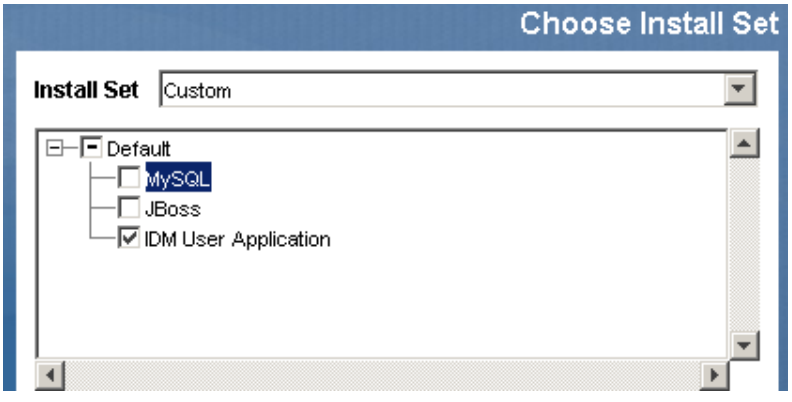
   If this command returns several lines ending in `mysqld`, then the daemon is running.

3. Start JBoss and the user application using the *start-jboss.bat* (Windows) or *start-jboss.sh* (Linux) file, located in the *IDM* directory.



4 Perform a custom install of the user application on each additional node in the JBoss cluster.

- Select only the user application for installation:



- Specify the IP address or host name of the server on which the database for the user application is installed.
- Specify the database user name and password for the user application database. If you are using MySQL, the user name is `root`, and the password is whatever you specified during the installation process in Step 1.
- In the installation program *IDM Configuration* screen, select the "*clustering (all)*" option.
- Select other installation options as appropriate for your environment.

5 Start each node in the JBoss cluster using the *start-jboss.bat* (Windows) or *start-jboss.sh* (Linux), located in the *IDM* directory.

Deploying the User Application to a Cluster Using JBoss Farming

Do not use JBoss farming with JBoss version 4.0.2 or earlier, as you may experience problems (see http://jira.jboss.com/jira/browse/JBAS-1899 (http://jira.jboss.com/jira/browse/JBAS-1899)). We recommend that you install the user application, using the user application installation program, on each node in the cluster (see "Using the User Application Installation Program on Each Node in the Cluster" on page 52 in this Chapter). However, if you want to use farming to deploy the user application to a JBoss cluster using JBoss 4.0.3 or higher, follow the steps below.

NOTE: These steps are for customers who wish to use JBoss 4.0.3 on their own, experimentally. The officially supported version is 4.0.2.

To deploy the user application to a cluster using JBoss farming:

1. Perform a custom install of the user application to one of the JBoss cluster nodes, selecting the user application and MySQL (if you are using MySQL; otherwise, install just the user application) for installation. You can perform the installation with all clusters in the node running, but the node on which you install the user application should be the first node in the cluster to start.
2. Copy the JDBC driver file (for example, if you are using MySQL, the JDBC driver is *mysql-connector-java-3.1.10-utf8-clob-fix-bin.jar*), located in the /server/IDM/lib directory, to the corresponding directory on each node in the cluster.
3. Copy the *cacerts* file from the /lib/security directory of the JRE that was installed with the user application to the JRE /lib/security directory of each node in the cluster.
4. Move the IDM.war file and the IDM-ds.xml data source file from the /deploy directory in the server configuration directory to the /farm directory in the server configuration directory. You must actually move the files. Do not leave the originals in the /deploy directory.
5. Start the database for the user application (if you are using the supplied MySQL, start MySQL using the *start-mysql.bat* file located in the /IDM/mysql directory).
6. Start JBoss and the user application using the *start-jboss.bat* (Windows) or *start-jboss.sh* (Linux), located in the *IDM* directory on the node to which you installed the user application and user application database.
7. Start the other nodes in the cluster.

2.4.3 Configuring the User Application Cluster Group Caching Configuration

Users who are familiar with JGroups and JBoss clustering can modify the cluster group caching configuration, using the user application administration user interface (see Section 13.3.5, “Cache settings for clusters,” on page 213). Changes to the cluster configuration only take effect for a server node when the server node is restarted.

2.4.4 Configuring Workflows for Clustering

Workflow engine clustering works independently of the user application cache framework. There are several steps that you must perform to ensure that the workflow engine works correctly in a cluster environment.

- All servers in the cluster need to be pointing to the same database. If you install the user application to the cluster using the recommended method (see “Using the User Application Installation Program on Each Node in the Cluster” on page 52), you accomplish this by specifying, during the installation process, the IP address or host name of the server on which the database for the user application is installed. If you use farming to deploy the user application to cluster nodes (see “Deploying the User Application to a Cluster Using JBoss Farming” on page 53), you accomplish this by moving the data source file (IDM-ds.xml) from the /deploy directory to the /farm directory on node on which the user application was first installed. This causes the data source to be deployed to all nodes in the cluster.

- Each server in the cluster needs to be started with a unique engine-id. This can done by setting the *com.novell.afw.wf.engine-id* system property at server startup. For example, if you wanted to start JBoss and assign the engine id `ENGINE1` to the workflow engine for that server, you would use the following command:

  run.sh -Dcom.novell.afw.wf.engine-id=ENGINE1 (Linux)

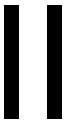
  run.bat -Dcom.novell.afw.wf.engine-id=ENGINE1 (Windows)

Once started by an workflow engine running on a particular server, a workflow process instance can only run and complete on that server. This ensures that the workflow process executes safely. However, it does not provide process instance failover support. If a server in the cluster crashes, the process instance will not be restarted until an engine with the same ID is restarted.

If a server computer cannot be restarted because of a serious hardware or software failure, you can start the application server on a new computer, using the same workflow engine ID that was used on the unrecoverable machine. Since the engine ID is a logical name, not a direct mapping to the physical computer on which the engine was running, the interrupted process instance will complete successfully on the new computer.

Process instances are owned by the engine that started the process. However, a user may log on to any user application in a cluster to view process detail, retract processes, or complete tasks assigned to them. Processes that are retracted or tasks that are completed on an engine that does not own the process enter a pending state and resume execution once they are discovered by the engine that owns them.

Configuring the User Application Environment



These chapters tell you how to configure various aspects of the Identity Manager user application environment to meet the needs of your organization.

Configuring the User Application Driver

3

3.1 About the User Application driver

The user application driver is responsible for starting provisioning workflows and for notifying the user application of changes in the Identity Vault (for example, when you make changes to the directory abstraction layer using the Designer for Identity Manager). Only the Subscriber channel is used in this driver. The driver processes messages from the Identity Vault to the user application running on an application server. While there are events that occur in the user application that are reported back to the Identity Vault, these events do not flow through the Publisher channel of the user application driver.

When the application server is started, the driver establishes a session with the application server. The driver sends messages to the user application running on the application server (for example, “retrieve a new set of virtual directory definitions”).

The source components of the driver include:

- ComposerDriverShim.jar – the Composer Driver Shim. It is installed in the *lib* directory *\Novell\NDS\lib* in Windows or the *classes* directory */usr/lib/dirxml/classes* in Linux
- srvprvUAD.jar – The Application Driver Shim. It is installed in the *lib* directory *\Novell\NDS\lib* in Windows or the *classes* directory */usr/lib/dirxml/classes* in Linux
- UserApplicationDriver.xml - A file that contains preconfiguration data for setting up the new driver. It is installed in the *DirXML.Drivers* directory *\Tomcat\webapps\nps\DirXML.Drivers* in Windows or */usr/lib/dirxml/rules/DirXML.Drivers* in Linux

The user application driver components are installed when you install Identity Manager 3. Before you can run the Identity Manager 3 user application, you must add the user application driver to a new or existing driver set, and activate the driver.

Depending on your work environment, very little configuration of the user application driver may be required, or you may want to implement a complex set of business rules in the driver policies. The user application driver provides the same flexible mechanisms for data synchronization as other Identity Manager drivers.

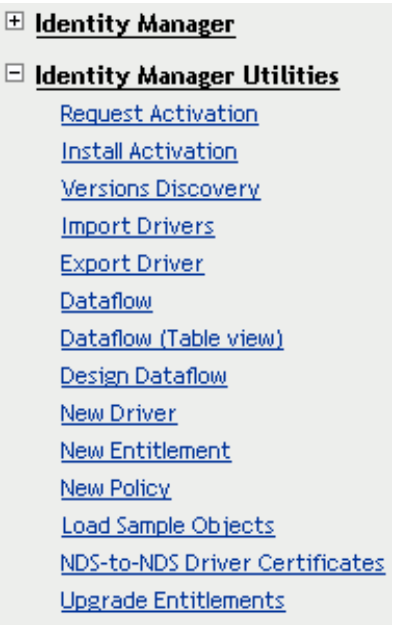
This Chapter describes how to create, configure, and start a user application driver, and how to configure the driver to automatically trigger a workflow based on an event in the Identity Vault. It contains these sections:

- Section 3.2, “Creating the User Application driver,” on page 60
- Section 3.3, “Starting the User Application driver,” on page 65
- Section 3.4, “Setting up Workflows to start automatically,” on page 66

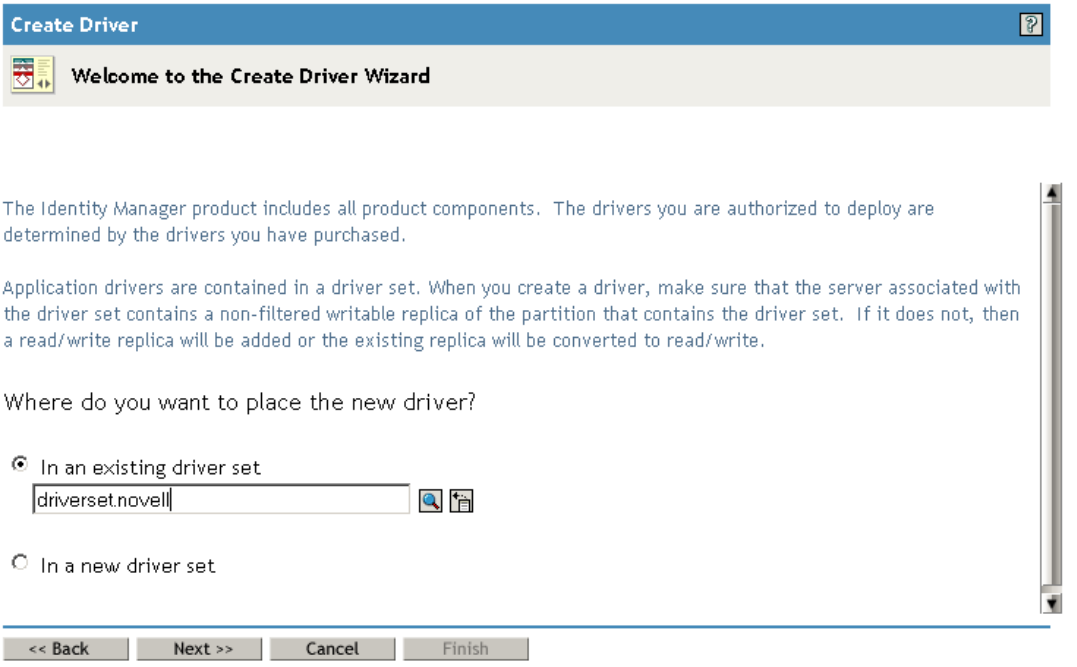
3.2 Creating the User Application driver

To create the driver:

1. Log into the instance of iManager that manages your Identity Vault.
2. Open the *Identity Manager Utilities* node in the iManager navigation frame.



3. Click *New Driver*. The Create Driver Wizard is displayed:

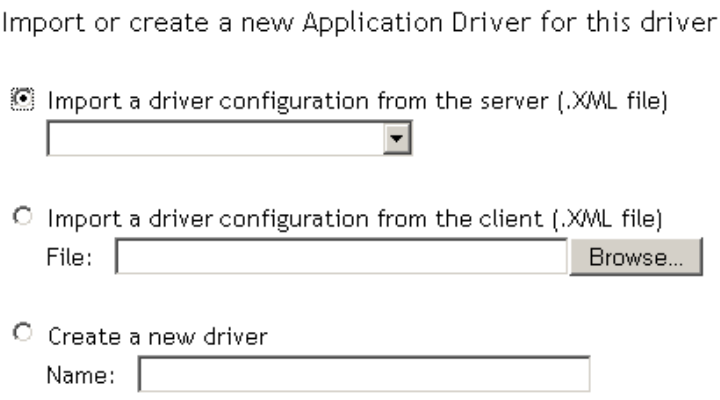


The next step is to select where you would like to create the new driver. You can create the driver in an existing driver set, or create a new driver set.

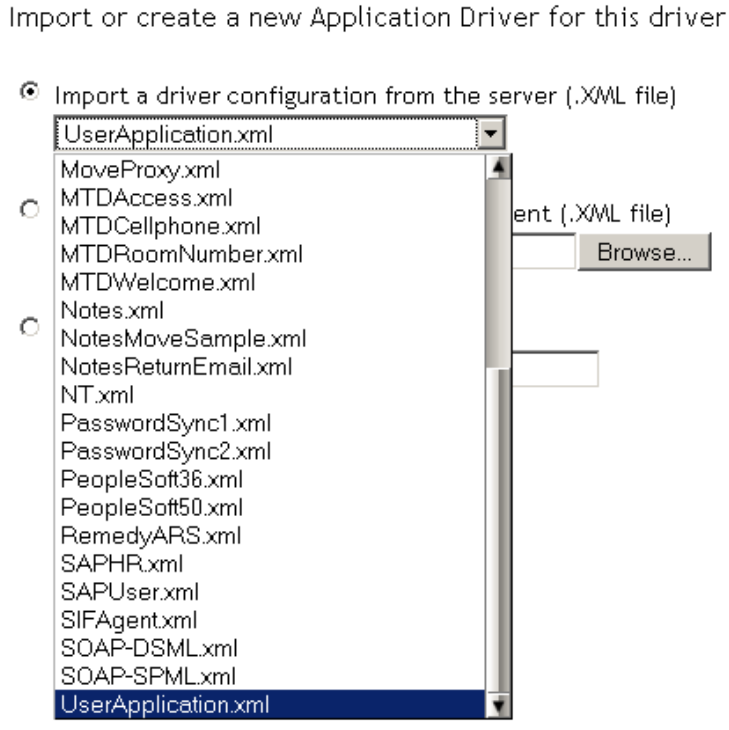
4 If you select *In an existing driver set*, a wizard is displayed that you use to browse the Identity Vault to locate the driver set. Select the existing driver set, then select *Next*.

If you select *In a new driver set*, a screen is displayed that you use to define the properties for the new driver set. Specify a name, a tree context, and a server for the driver set, then select *Next*.

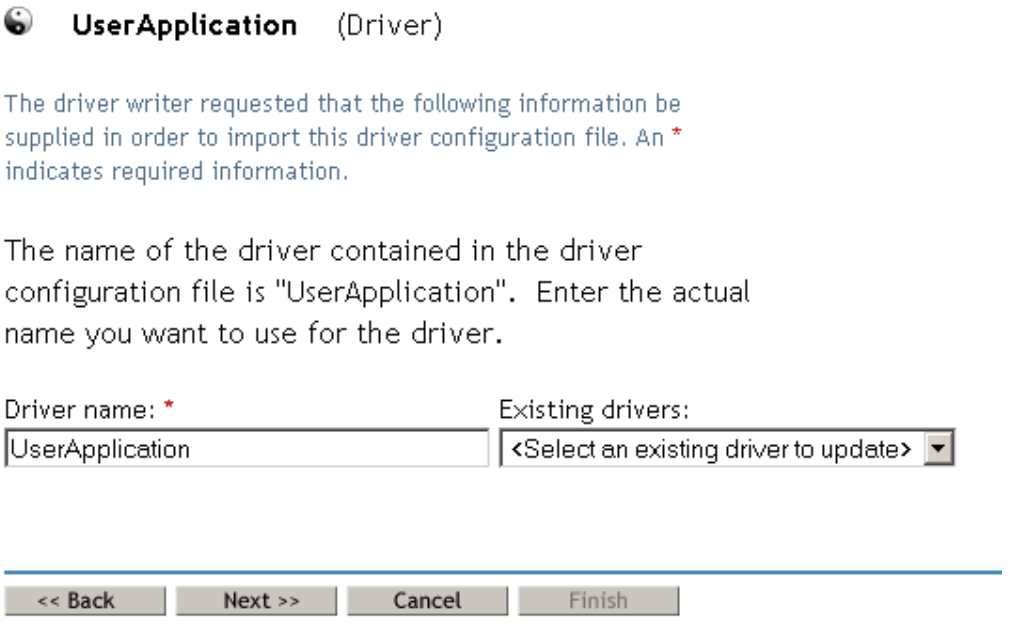
The next screen in the *Create Driver Wizard* is displayed:



5 Click the *Import a driver configuration from the server* option, then select *UserApplication.xml* from the list of XML files:



6 Click *Next*. The *Create Driver Wizard* displays a page that you use to name and configure the driver:



The default name of the driver is UserApplication. While you can use the default name, you may want to choose a more meaningful name for your project.

7 If desired, type a new name for the driver in the *Driver name* field.

8 In the *Authentication ID* field, specify the DN of the user application administrator (see Section 1.1.2, “User Application Administrator,” on page 22 for a description of the user application administrator), using the dot format (for example, admin.orgunit.novell).

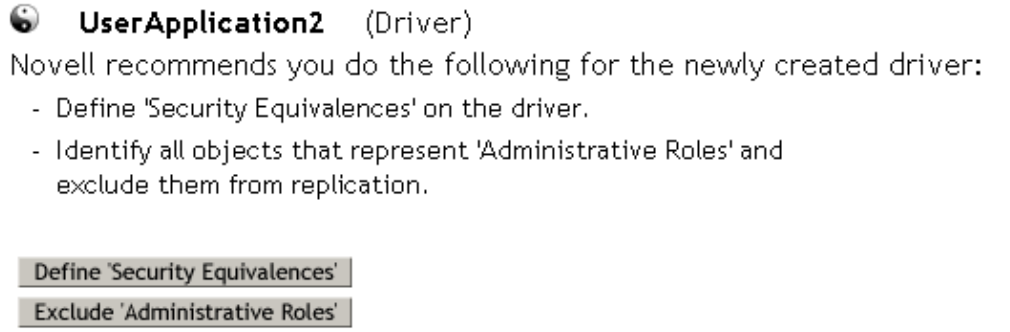
9 In the *Application Password* and *Reenter the password* fields, specify the password for the user application administrator identified in the *Authentication ID* field.

10 In the *Application Context* field, type the application name that was specified when the user application was installed. The default name is `IDM`.

11 In the *Host* field, specify the host name or IP address of the application server on which the user application runs.

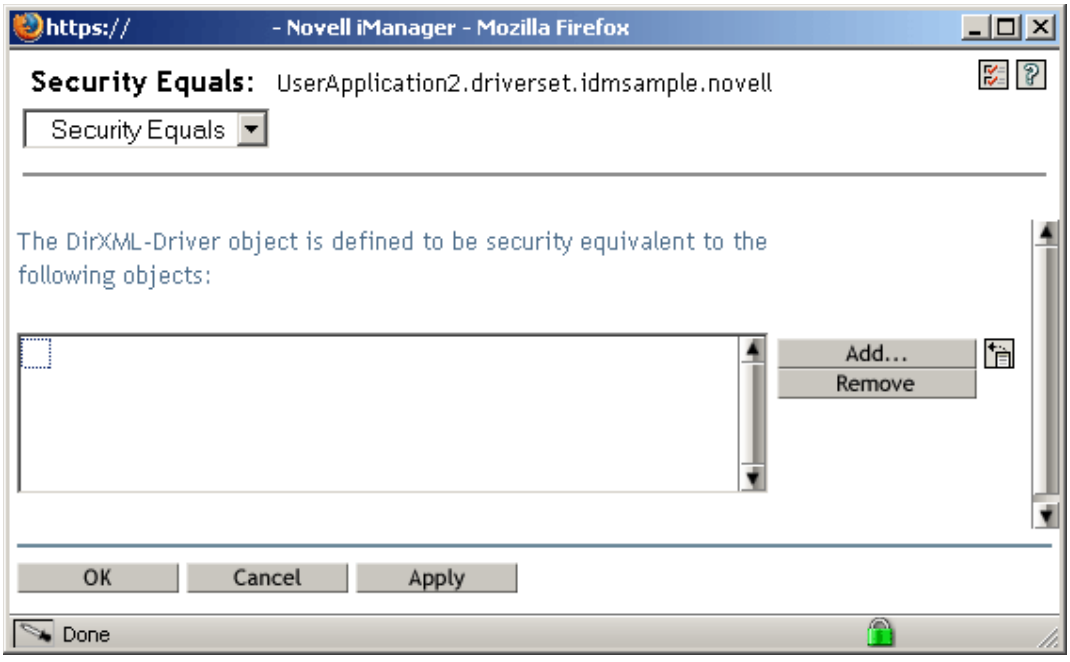
12 In the *Port* field, specify the port on which the driver will communicate with the user application running on the application server (for example, 8080).

13 Click *Next*. A message indicating that the driver configuration is being imported is displayed, then the next page of the *Create Driver* wizard is displayed:

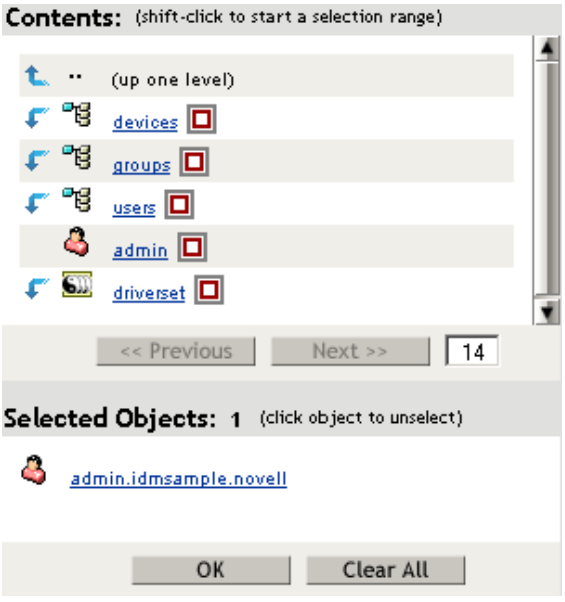


The driver object must be granted sufficient Identity Vault rights to any object that it reads or writes. You do this by granting *Security Equivalences* to the driver object. The driver must have Read/Write access to users, post offices, resources, and distribution lists, and Create, Read, and Write rights to the post office container. Normally, the driver should be given security equal to Admin.

14 Click *Define Security Equivalences*. A new window is displayed:



15 Click *Add*. A window is displayed that you use to select an object in the tree that has the appropriate level of rights that you would like to assign to this driver (for example, `admin`):



16 Select an object that has the desired level of Identity Vault rights from the tree, then click *OK*. You are returned to the previous window.

17 Click *OK*. You are returned to the *Create Driver* wizard.

18 Click *Exclude Administrative Roles*. The *Excluded Users* window is displayed. You use this feature to prevent an admin from being locked out of the user application driver if the administrator password changes in another Identity Vault that replicates back to the tree to which this driver belongs.

19 Click *Add*. A window is displayed that you use to browse the directory tree for users who should be excluded from having their data passed to the driver. Normally, you would exclude admin objects, since replicating their data across a driver connection is not good practice in most cases.

20 Select the administrative roles that you want to exclude, then click *OK*. You are returned to the previous window.

21 Click *OK*. You are returned to the *Create Driver* wizard.

22 Click *Next.* A driver summary page is displayed.

23 Click *Finish with Overview*. A graphical representation of the driver in the Identity Vault is displayed:

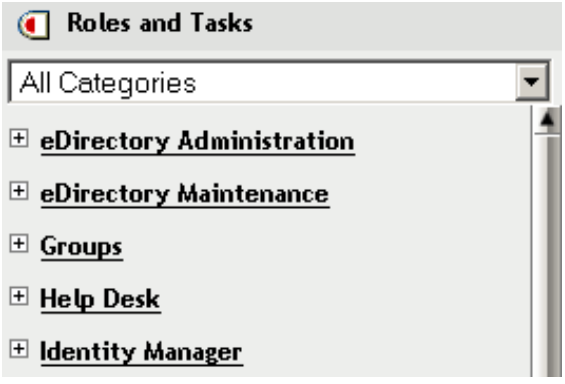
NOTE: You can view this screen again at any time by using the *Identity Manager Overview* link under *Identity Manager* in the iManager navigation tree.

The new driver appears as a large icon connected to the Identity Vault trunk.

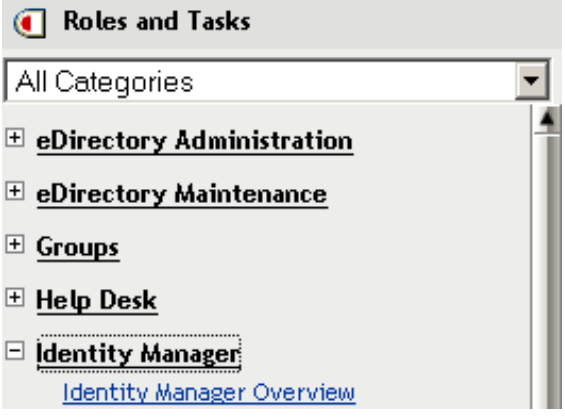
3.3 Starting the User Application driver

To start the user application driver:

1. Click the *Identity Manager* link in the iManager navigation tree to see the commands available in the Identity Manager category:



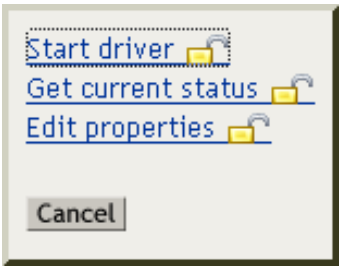
2. Click the *Identity Manager Overview* link under the *Identity Manager* link in the iManager navigation tree:



   A wizard is displayed that you use to browse the system to locate the driver set that contains the driver that you want to activate.

3. Select the driver set, then click *Next.* The *Identity Manager Overview* page is displayed.
4. Click the round status indicator in the upper right corner of the driver icon:

   A menu that lists commands for starting and stopping the driver, and editing driver properties, is displayed:



5 Click *Start driver*.

3.4 Setting up Workflows to start automatically

When the provisioning module is installed, workflows are automatically started when a user starts a provisioning request by requesting a resource. In addition, the Identity Manager user application driver listens for events in the Identity Vault and, when configured to do so, responds to events by starting the appropriate provisioning workflows. For example, you can configure the user application driver to automatically start a provisioning workflow if a new user is added to the Identity Vault. You configure the user application driver to automatically start workflows using Identity Manager policies and rules.

3.4.1 About policies

You can use filters and policies with the user application driver in the same way that you can with other Identity Manager drivers. When an event occurs in the Identity Vault, Identity Manager creates an XML document that describes the event. The XML document is passed along the channel to the connected system (in this case, the connected system is the user application). Filters and policies associated with a driver allow you to define how to respond to the event, and in the process transform that XML document to the format that is expected by the connected system. Identity Manager provides several categories of policies (for example, Event Transformation, Command Transformation, Schema Mapping, Output Transformation) that you can apply, in a prescribed order, to transform the XML document. In this section we provide an example of starting a workflow based on events in the Identity Vault. While any of the policies can be used to trigger a workflow, this example demonstrates the easiest and most useful method.

When you create a user application driver, an Event Transformation Policy is created for use by the driver. The Event Transformation Policy is responsible for creating the XML document that will be processed by the remaining Subscriber channel policies.

NOTE: Do not change the Event Transformation policy that was created when the user application driver was created. The DN of this policy begins with `Manage.Modify.Subscriber`. Changing this policy may cause the workflow process to fail.

An empty Schema Mapping Policy is also created. You can use this policy as a starting point for triggering a workflow based on events in the Identity Vault.

3.4.2 Setting up a workflow to start based on an event in the Identity Vaul

The simplest method of starting a workflow automatically is accomplished using the Schema Mapping Policy Editor, and the user application driver provides an empty policy for you to edit for this purpose.

You use the Schema Mapping Policy Editor to map Identity Vault attributes (including the eDirectory *trigger* attribute that, when it changes, starts the workflow) to the runtime data of a target workflow. The runtime data is determined by the workflow definition template (see Chapter 22, "Configuring Provisioning Request Definitions," on page 309 for information about workflow definition templates). The runtime data is needed for a workflow to complete successfully. When a workflow is created, a number of *global attributes* are created in the Identity Vault that can be used to customize the behavior of the user application driver. A global attribute is an attribute that does not belong to any Identity Vault object class. These attributes are called <workflowName>_StartWorkflow, <workflowName>_recipient, and <workflowName>_reason. The are also two other attributes that always exist named AllWorkflows:reason and AllWorkflows:recipient. The _StartWorkflow attribute is used to start a workflow. The _recipient and _reason attributes are used for accepting runtime data needed by the workflow from the Identity Vault.

Before you perform this procedure, you should know the name of the Identity Vault attribute that you want to use as a trigger for the workflow. You also need to know the name of the workflow that you want to start. All workflows include a special attribute named `<workflowName>_StartApprovalFlow`. You configure a workflow to start automatically based on an event in the Identity Vault by mapping the desired eDirectory attribute to the <workflowName>_StartApprovalFlow attribute for the workflow.

To Set up a workflow to start based on an event in the Identity Vault:

1. In iManager, click the *Identity Manager Overview* link under Identity Manager in the iManager navigation tree.

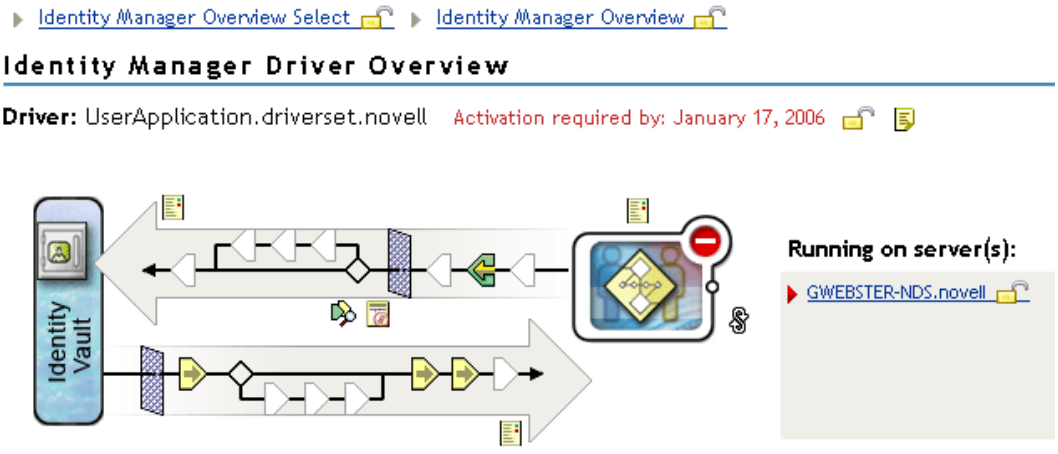
   The *Identity Manager Overview* page is displayed. This page prompts you to select a driver set.

2. Click *Search Entire Tree*; then click *Search*. The *Identity Manager Overview* page is displayed, with a graphic that depicts the drivers in the currently selected driver set.

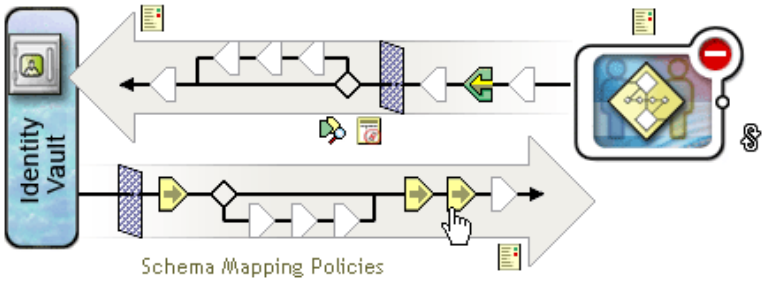
3. Click the large driver icon for the user application driver:

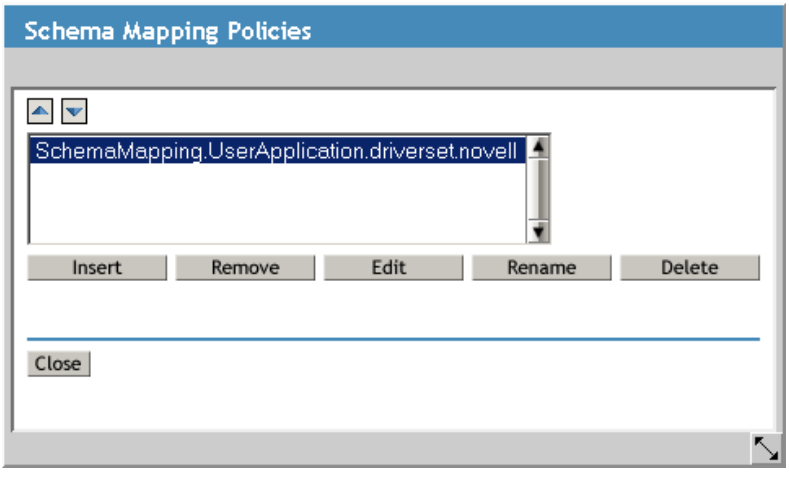
   The *Identity Manager Driver Overview* is displayed:



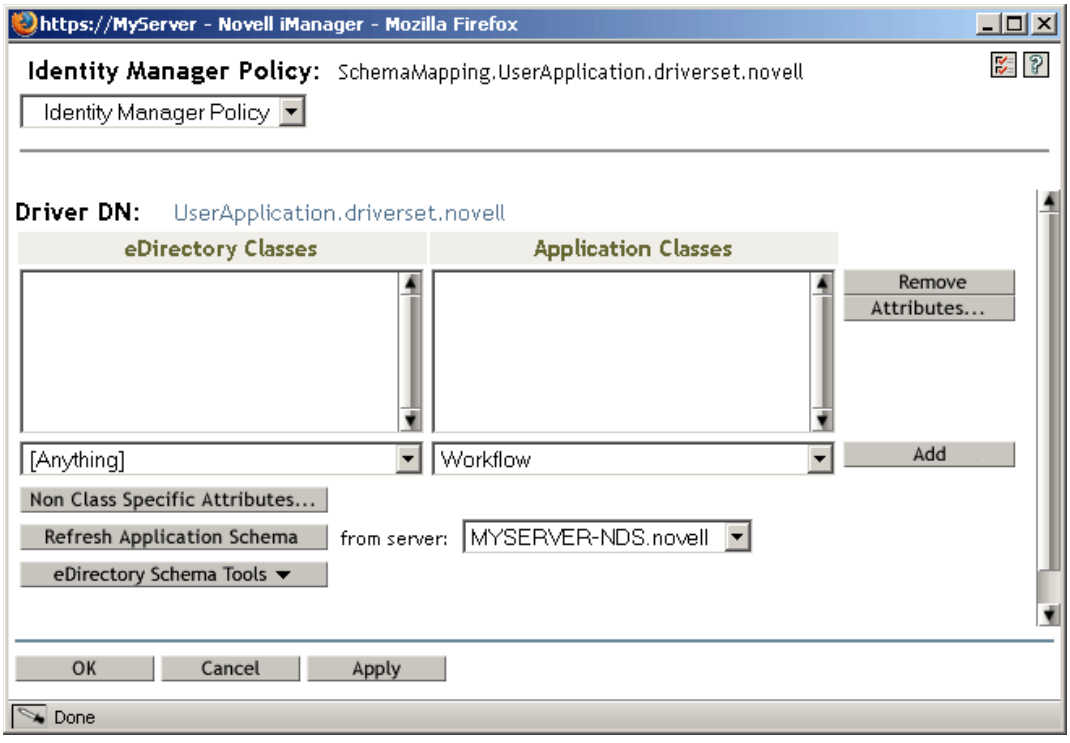
The top horizontal arrow represents the Publisher channel (which is not used in the user application driver) and the bottom horizontal arrow represents the Subscriber channel. As you pass the mouse pointer over an object in the graphic, a description of the object is displayed:



4 Click the *Schema Mapping Policies* icon for the Subscriber channel. The *Schema Mapping Policies* dialog box is displayed, with the name of the default schema mapping policy highlighted:



5 Click *Edit*. The *Identity Manager Policy* dialog box is displayed. This dialog box is used to map Identity Vault classes to application classes. This procedure does not make use of this feature. Instead we will be mapping eDirectory attributes to global user application attributes.

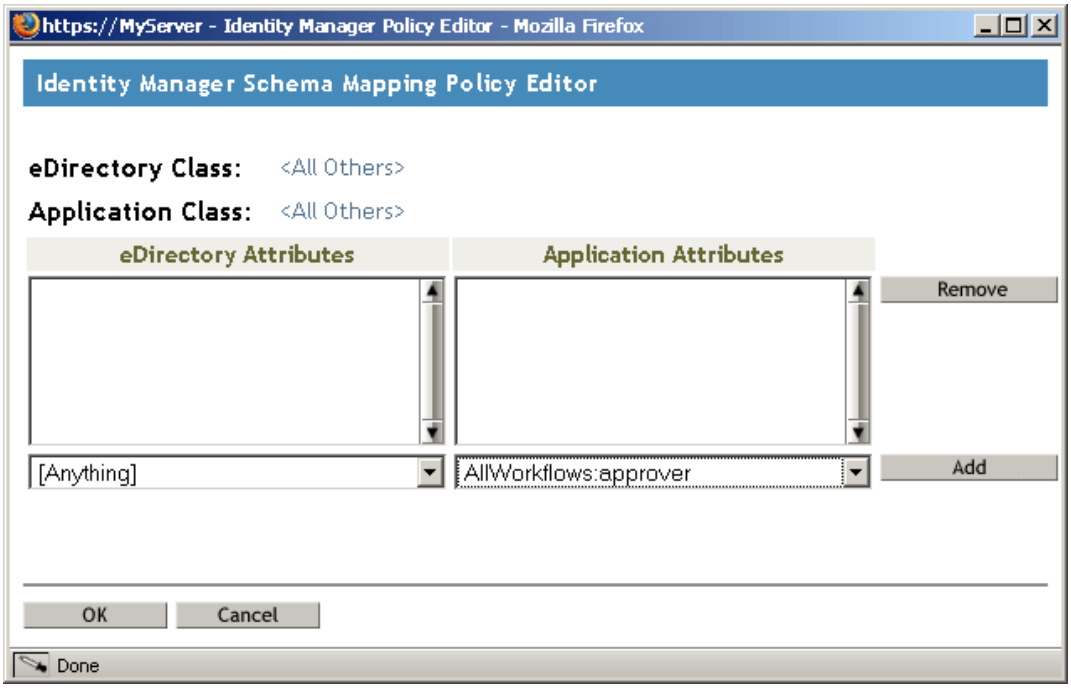


6 Click *Refresh Application Schema*. A message is displayed informing you that the driver must be stopped to read the schema, then restarted. It may take about 60 seconds to refresh the schema. This step reads the latest set of workflow information in preparation for the following step, which specifies the information to move from the Identity Vault to the workflow that will be started.

7 Click *OK* to refresh the schema. A message is displayed when the schema refresh is completed.

8 Click *OK* to close the schema refresh message. You are returned to the *Identity Manager Policy* dialog box.

9 Click *Non Class Specific Attributes*. The *Identity Manager Schema Mapping Policy Editor* is displayed.



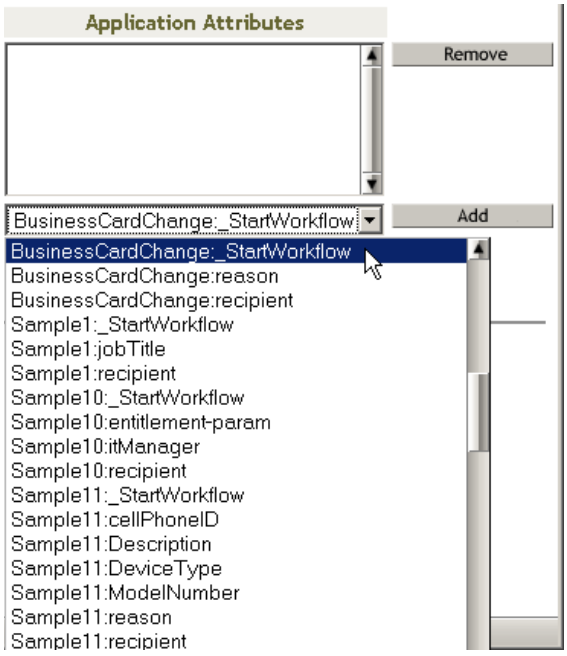
The *eDirectory Attributes* dropdown list contains all eDirectory attributes.

The *Application Attributes* dropdown list contains the attributes in all active Workflows. Attributes in the list are prefaced with either `AllWorkflows` (meaning that the attribute applies to all workflows), or the name of a specific workflow. If you want the same eDirectory attribute (for example `manager`) to be mapped to the `manager` attribute for all workflows, you would map `manager` to `Allworkflows:manager`. If you wanted a different eDirectory attribute (for example, `HRmanager`) to be used for a specific workflow, you would map the eDirectory attribute to the specific workflow attribute (for example `BusinessCardChange:manager`).

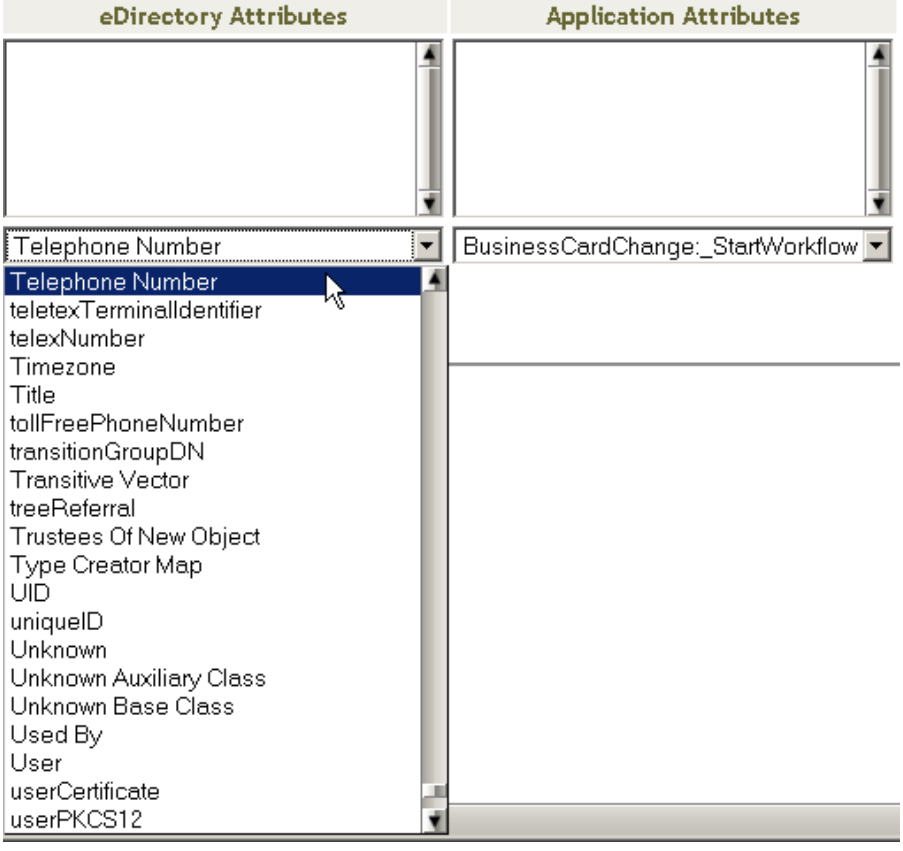
Attributes that have been mapped are displayed side-by-side in the *eDirectory Attributes* and *Application Attributes* columns.

In the following steps, we will map the eDirectory attribute that we want to use to start the workflow to the _StartWorkflow attribute for that workflow. If additional eDirectory attributes are expected by the workflow, you should also map those attributes. For example, if an eDirectory `Address` attribute is the trigger for a workflow, the workflow may also require attributes like `City` and `State`. Alternatively, these attributes may be mapped in policies.

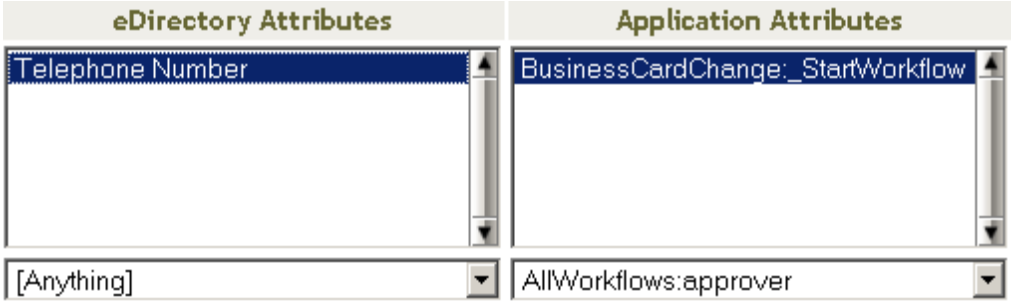
10 In the *Application Attributes* list, select the _StartWorkflow attribute for the Workflow that you want to configure. The following example shows the _StartWorkflow attribute for a BusinessCardChange workflow (BusinessCardChange_StartWorkflow).



11 In the *eDirectory Attributes* list, select the eDirectory attribute that you want to use to start the workflow when that attribute changes. In the following example, the Telephone attribute is selected. This means that the BusinessCardChange workflow will start whenever an employee's telephone number changes.



12 Click *Add*. The eDirectory attribute is mapped to the Application attribute.



13 If there are additional eDirectory attributes that are needed by the workflow, repeat Step 10 through Step 12 until you have mapped all of the attributes that you need to map.

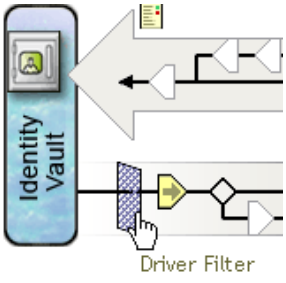
The workflow will start automatically when a change occurs in the eDirectory attribute that is mapped to an application _StartApprovalFlow attribute. However, the eDirectory attribute will only reach the Schema Mapping policy if the eDirectory attribute is included in the Subscriber channel Driver Filter. In the following steps we will add the eDirectory attribute to the Subscriber channel Driver Filter

14 Click *OK* to close the *Identity Manager Schema Mapping Policy Editor.*

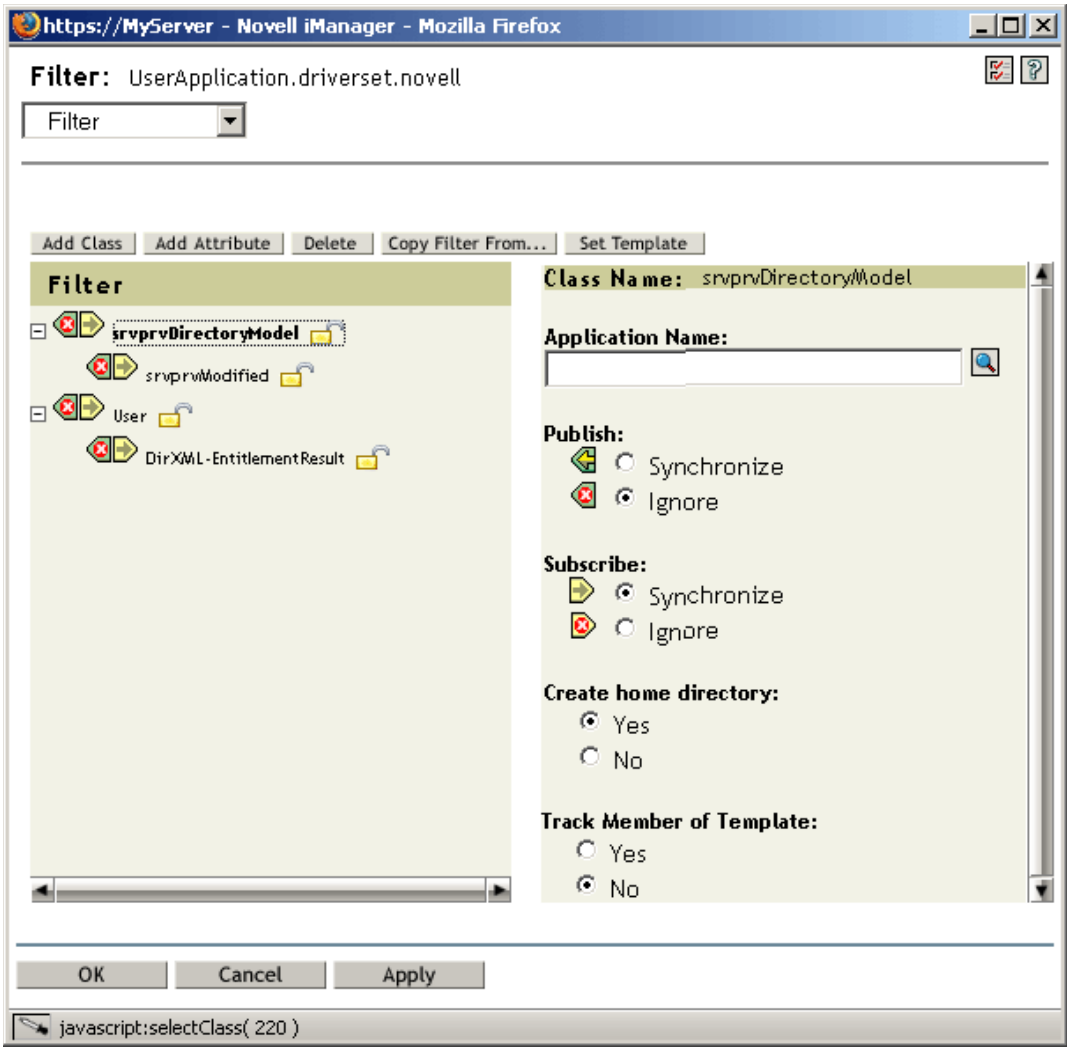
15 Click *OK* to close the *Identity Manager Policy* dialog box.

16 Click *Close* to close the Schema Mapping Policies dialog box.

17 Click the *Driver Filter* icon for the Subscriber channel.



The filter window is displayed:



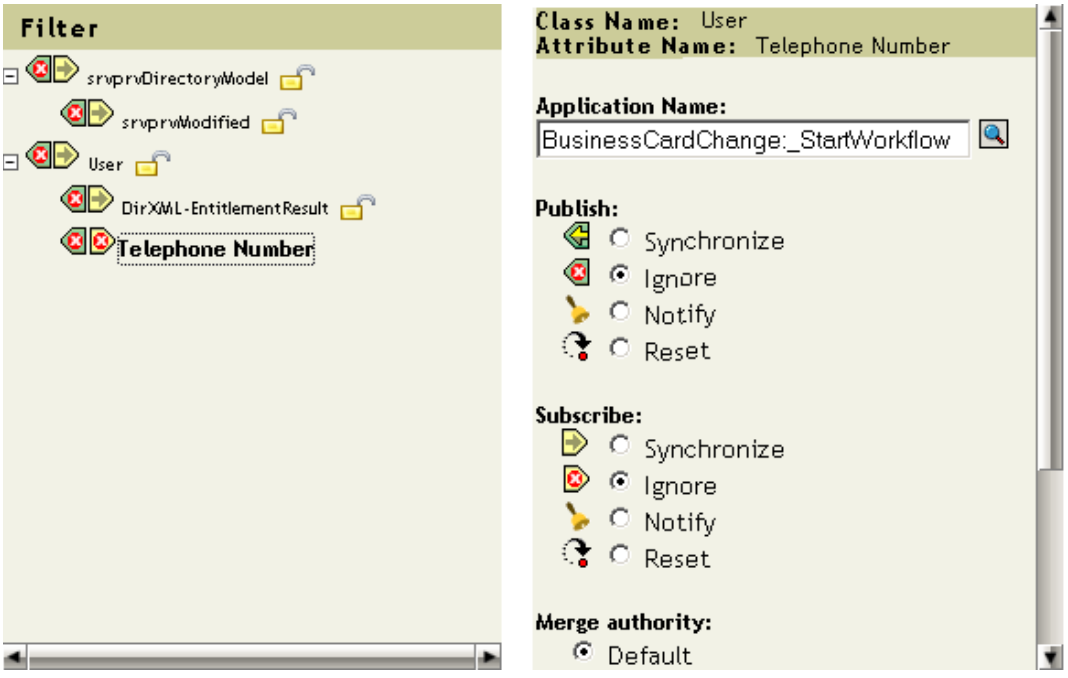
Event filters specify the object classes and the attributes for which the Identity Manager engine processes events. The read-only *Filter* list on the left shows the attributes of the class. The *Class Name* list on the right displays options associated with the target object.

18 Click the name of the class to which the attribute that you want to add to the filter belongs (for example, User).

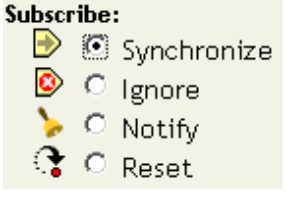
19 Click *Add Attribute*. A list of attributes is displayed.

20 Select an attribute, then click *OK*. The attribute is added to the *Filter* list.

21 Click on the attribute name. The synchronization options for the attribute are displayed on the panel on the right.



22 Under *Subscribe*, click *Synchronize*.



23 Specify any other attributes for the filter. Select *Synchronize* for an attribute if you want changes to attribute values to be reported and synchronized. Select *Ignore* if you do not want changes to attribute values to be reported and synchronized.

24 Click *OK*. A message is displayed asking you if you'd like the driver to be restarted to put the changes into effect.

25 Click *OK*. You are returned to the *Identity Manager Driver Overview* page.

Configuring the Directory Abstraction Layer

4

This chapter describes how to use the directory abstraction layer editor to define the directory abstraction layer data definitions used by the Identity Manager user application. Topics include:

4.1 About directory abstraction layer definitions

The *directory abstraction layer* is a set of data definitions that provide a logical view of an identity vault. The directory abstraction layer defines:

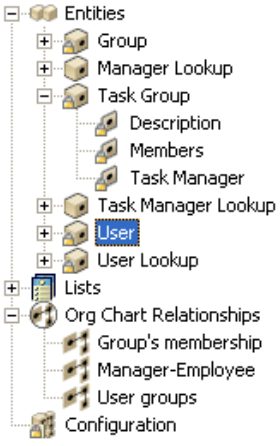
- The identity vault objects and attributes that can be used in the Identity Manager user application.
- How identity vault data is displayed in the user interface.
- The relationships available to the Org Chart portlet.

You will use the *directory abstraction layer editor* to change these data definitions if you want to modify the user application appearance or function. You can change it by:

- Adding other identity vault objects
- Changing the set of attributes available for an identity vault object
- Changing the contents of lists
- Showing different relationships among identity vault objects

The Identity Manager user application installation procedure installs and deploys the base set of abstraction layer definitions that are needed for the user application to function properly. This install also creates eDirectory schema extensions that are used by the user application driver and the user application. You can learn more about these schema extensions in Appendix A, “Schema Extensions,” on page 347. This same base set of files are created on the local file system when you create a new User Application Driver instance via Designer for Identity Manager.

Required data abstraction layer data definitions As you begin to customize your own Identity Manager user application, you’ll want to make changes to the directory abstraction layer objects. However, certain identity vault objects (called entities), attributes, relationships, and lists cannot be removed or changed or the user application will not function properly. The definitions that cannot be removed are identified by a padlock icon. From this example, you can see that Task Group entity and all of its attributes are locked.



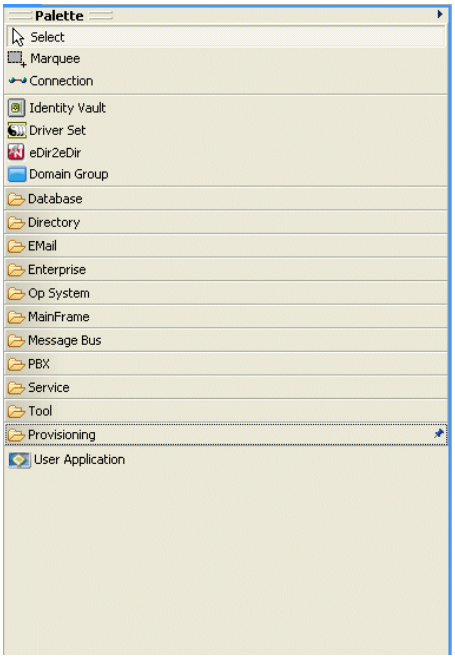
Where directory abstraction layer definitions are stored *Directory abstraction layer* definitions are XML files that are:

- *Stored* locally in the file system of the designer machine in the provisioning project's Provisioning\AppConfig\DirectoryModel subdirectory. If you have more than one User Application in your project, the directory names are numbered. For example, AppConfig1, AppConfig2, and so on.
- *Deployed* to the User Application Driver's AppConfig.DirectoryModel container. The XML files are stored in the XMLData attribute on the corresponding directory abstraction layer definition object. Each entity, relationship, and list is a unique object instance contained in the User Application Driver's AppConfig.DirectoryModel container.
- *Cached* on the application server where the user application is deployed.

4.2 Getting started

You'll use the features of the Designer for Identity Manager Provisioning View and the directory abstraction layer editor to define the contents of the directory abstraction layer. Follow these steps to get started:

Step	Task	Description
1	Create an Identity Manager project	This includes: • Configuring the Identity Vault • Specifying the Driver Set properties See the Identity Manager documentation.

Step	Task	Description
2	Add a User Application driver to the Modeler	You can find the Identity Manager user application driver in the Provisioning folder of the Modeler Palette. 
3	Complete the User Application driver configuration	See the procedure Section 4.2.1, “Completing the User Application driver configuration,” on page 77.
4	Access the Provisioning View	See Section 4.2.2, “Accessing the Provisioning View,” on page 80.
5	Start the directory abstraction layer editor	See “To open the directory abstraction layer editor:” on page 82.

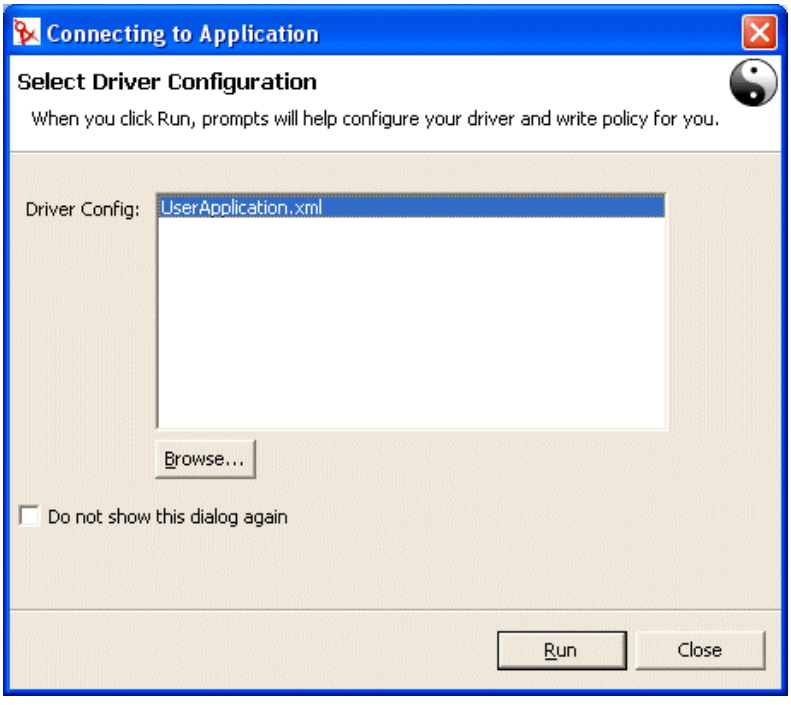
4.2.1 Completing the User Application driver configuration

Follow these steps once you have an Identity Manager project created.

To complete the User Application driver configuration:

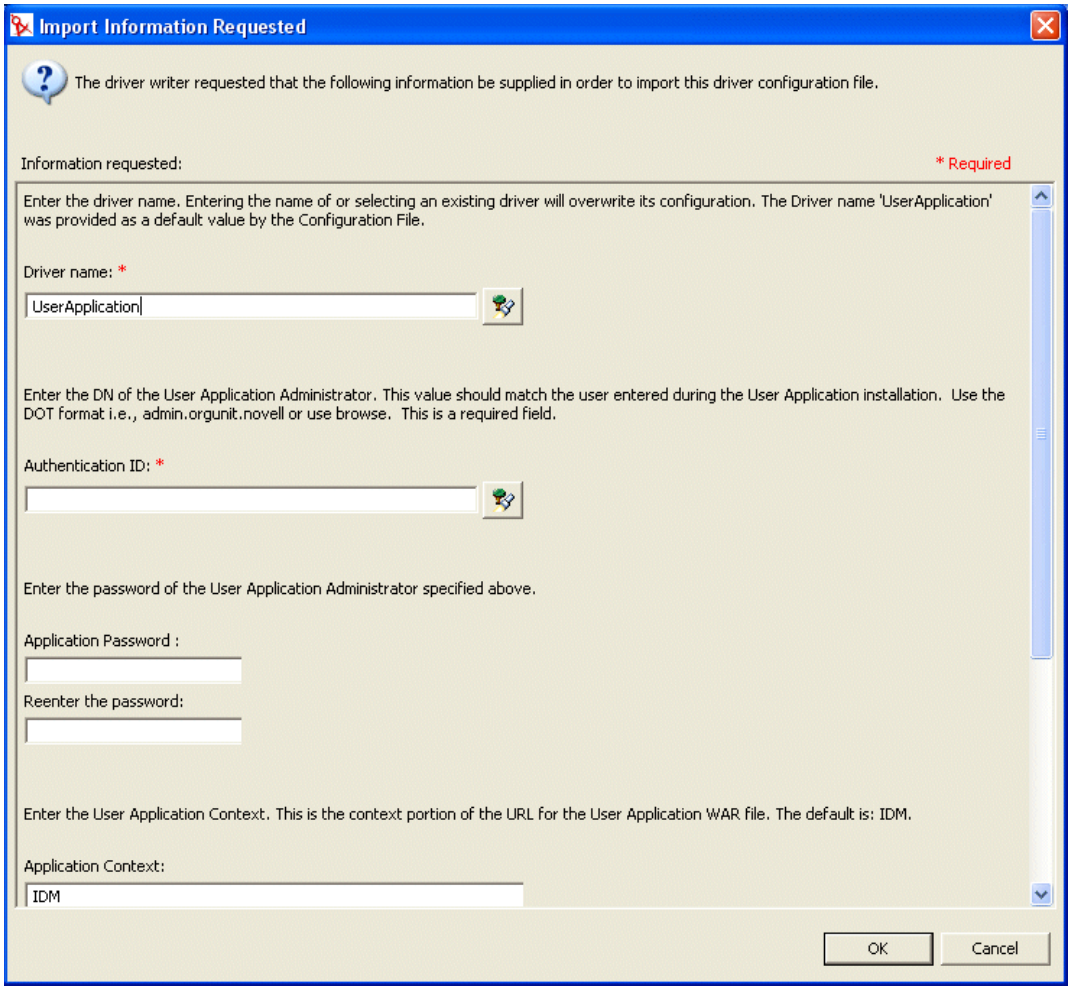
1. Drop a *User Application* driver icon on the canvas.

   You are prompted for a driver configuration.



2 Select *UserApplication.xml* (the default), then click *Run*.

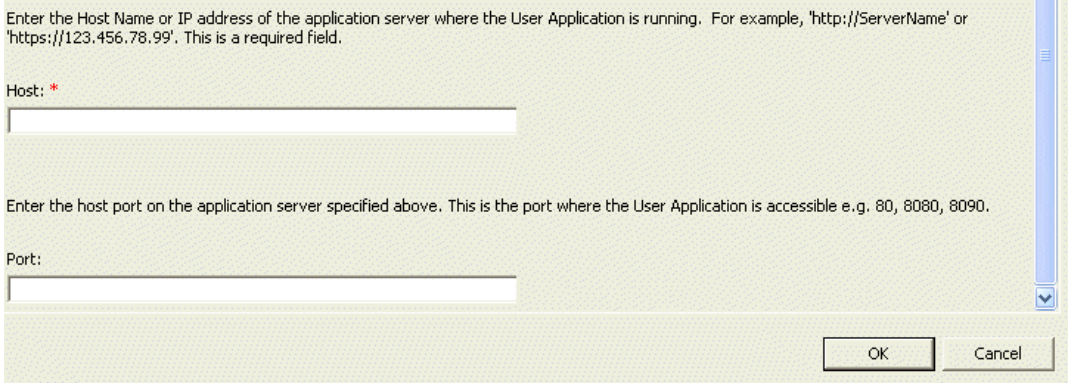
3 Specify how the wizard should handle validation of your entries by clicking Yes or No.



.

.

.



4 Complete the panel as follows:

Property	What to specify
Driver Name	• The name of an existing driver (the driver within the driver set that was specified during the user application installation). • The name of a new driver.
Authentication ID	The DN of the User Application Administrator.
Application password/Reenter password	The password for the User Application Administrator (above).
Application context	The name of the user application context (specified at install, for example, IDM).
Host	The host name or IP address of the application server where the Identity Manager user application is deployed. This information is used: • To trigger workflows on the application server to connect to access workflows (terminate, retract, and so on). • To update cached data definitions.
Port	The port for the Host above.

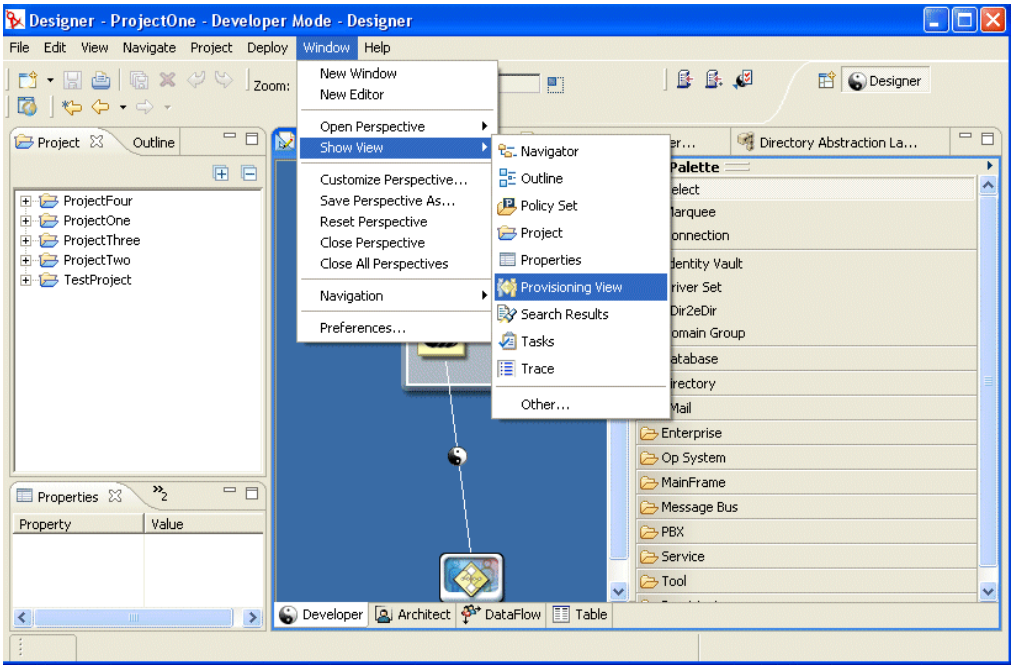
5 Click *OK*.

4.2.2 Accessing the Provisioning View

To access the Provisioning View:

1 Choose one of these ways:

- Select *Window>Show View>Provisioning View*.



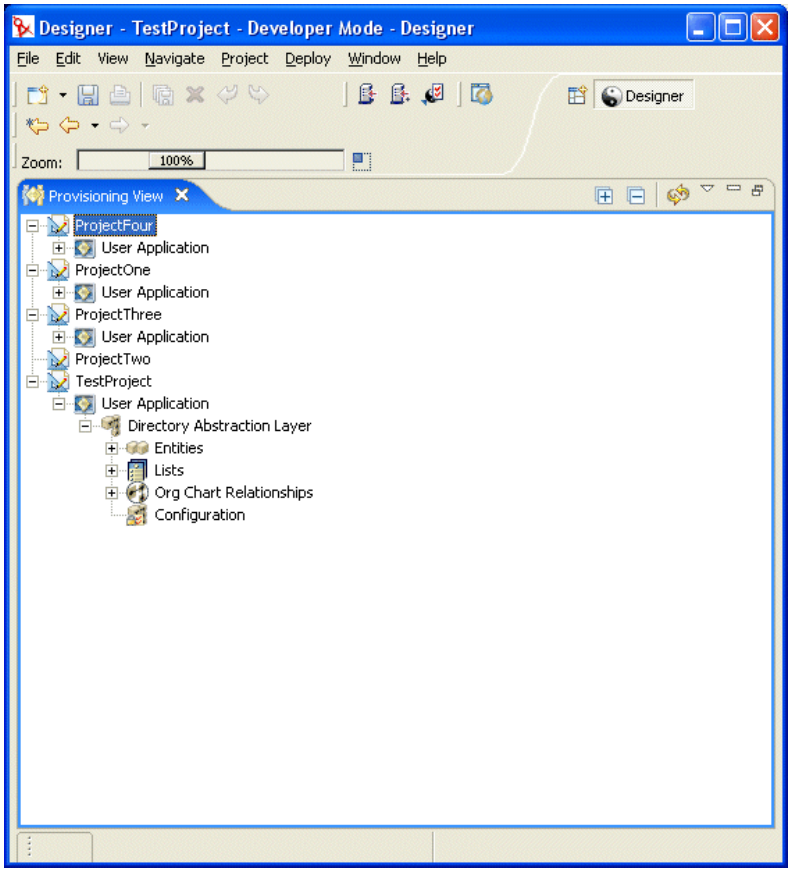
- Open the *Provisioning* folder and select *Provisioning View*.
- Click *OK*.

or

- Select the User Application icon, right-mouse and select *Application>Show Provisioning View*.

In the Provisioning View, you'll see the project you just created along with any other provisioning projects located in the same workspace.

TIP: If you do not see the applications that you expect in the view, it *might* be because the project is corrupt. If your project is corrupt, you must recreate it.



About the Provisioning View

The Provisioning View provides persistent access to the provisioning features. Double-clicking an item from the Provisioning View opens the editor for that item. You'll use the provisioning view to perform the following actions with the directory abstraction layer definitions:

- *Import* one or more object definitions from the identity vault.
- *Validate* the structure of the data definitions.
- *Deploy* your definitions to the identity vault specified in the project.
- *Create and delete* directory abstraction layer definitions.

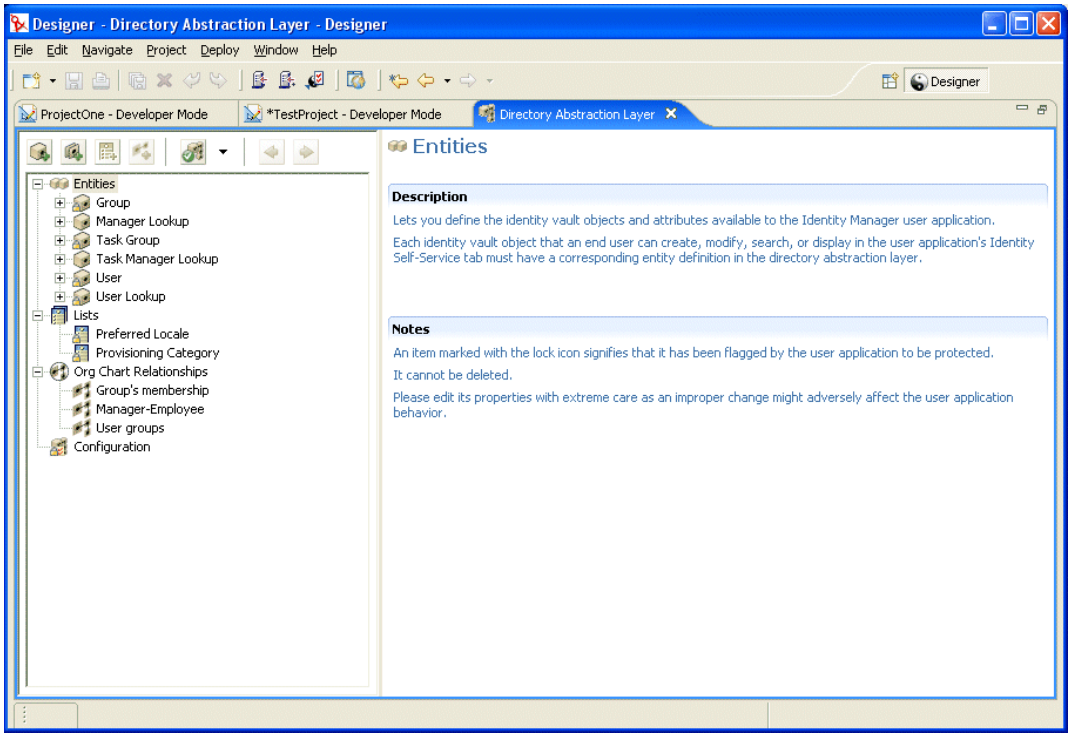
For more information, see Section 4.8, "Importing, validating, and deploying directory abstraction layer definitions," on page 109.

4.2.3 Starting the directory abstraction layer editor

To open the directory abstraction layer editor:

1. With the *Provisioning View* open navigate to the Directory Abstraction Layer node.
2. Double-click the Directory Abstraction Layer node.

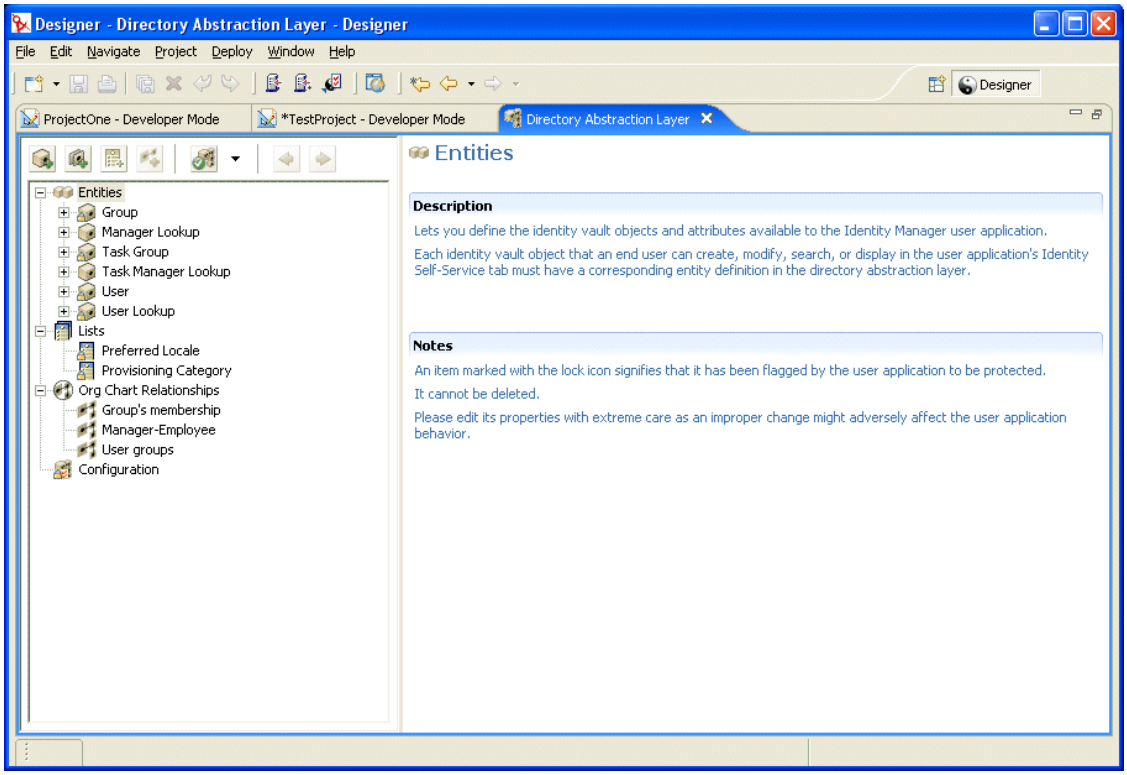
   You'll see a tree containing Entities, Lists, Org Chart Relationships, and Configuration.



About the directory abstraction layer editor

The directory abstraction layer editor provides a graphical way to define the set of XML files that comprise the directory abstraction layer. The directory abstraction layer editor is an Eclipse-based tool that you can access from the *Provisioning View* of an Identity Manager project.

When you open the directory abstraction layer editor the first time, you'll see a base set of abstraction layer objects that are created automatically each time you create a new provisioning project:



The nodes of the directory abstraction layer editor include:

Element	Description
Entities	Entities represent the identity vault objects configured for this project and available to the user application. There are two types of entities: • **Entities that are mapped from schema**. These entities represent objects that exist in the identity vault that are directly exposed to users via the user application. Users can typically create, search, and modify the attributes of these types of objects. • **Entities that represent LDAP relationships**. Also called DNLookups. These entities represent indexed searches and are used to support particular types of attributes that you want to expose. DNLookup entities provide information about relationships between LDAP objects. DNLookup entities are used by the: • The Org Chart portlet to determine relationships. • The Search List, Create, and Detail portlets to provide pop-up selection lists and DN contexts. For more information, see Section 4.3.3, “Defining entities,” on page 87.

Element	Description
Lists	Lets you define the contents of global lists. Global lists are: • Associated with an attribute. When the attribute is displayed in the user application, it is displayed as a dropdown list. • Used to display categories used by the Provisioning Request Configuration Plug-in to iManager. For more information, see Section 4.4, “Working with lists,” on page 101.
Org Chart Relationships	Used by the Organization Chart action of the Identity Self-Service tab of the user application. Lets you map hierarchical relationships among schema-based entities. For more information, see Section 4.5, “Working with Org Chart relationships,” on page 104.
Configuration	General configuration parameters. For more information, see Section 4.6, “Working with configuration settings,” on page 107.

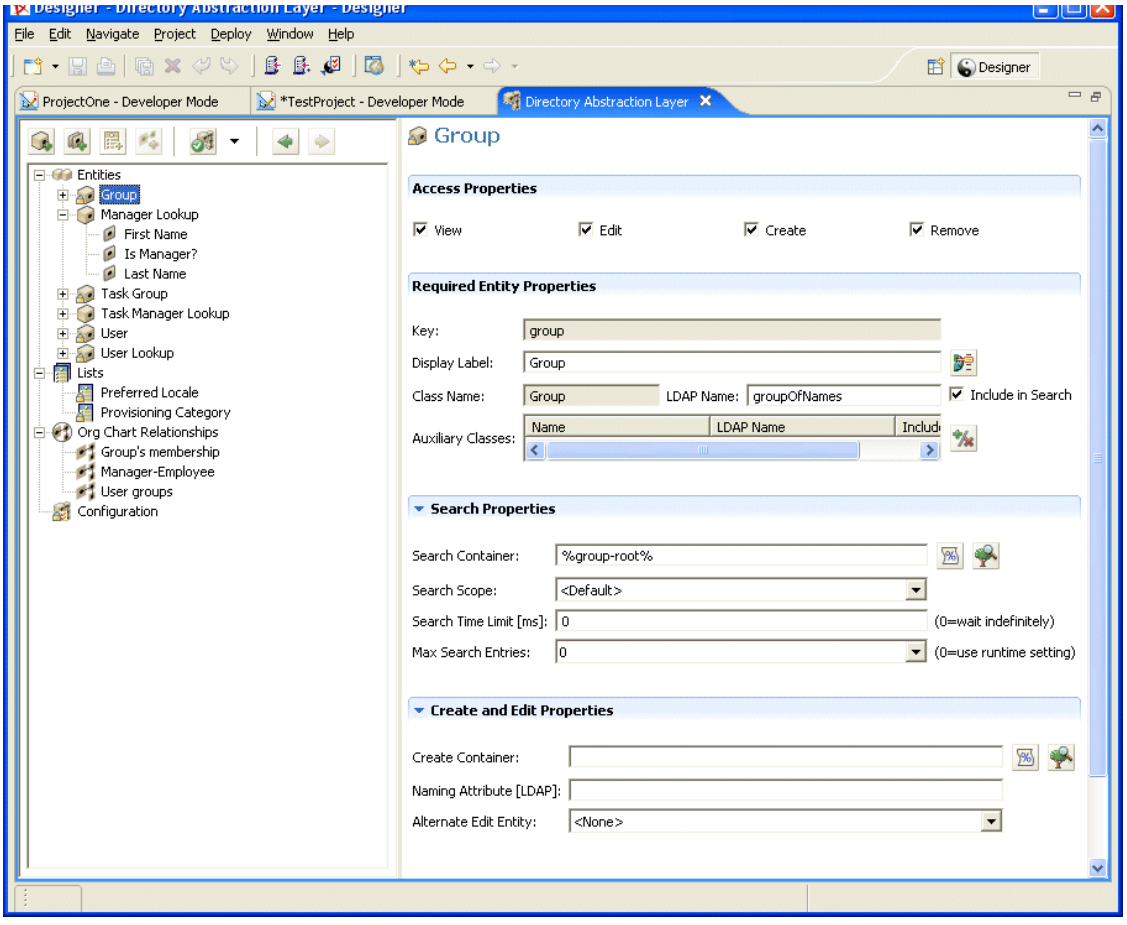
Where the XML files are stored locally The directory abstraction layer editor generates a single XML file for each entity, list, or relationship. The files are stored in the project’s Provisioning\AppConfig\DirectoryModel folder. The file name is based on the object’s key. They include:

Directory	Description
ChoiceDefs	Contains the files that define global lists. Files have the .choice extension.
EntityDefs	Contains the files that define the entities and attributes. Files have the .entity extension.
RelationshipDefs	Contains the files that define the relationships available to the Org Chart portlet. These files have the .relation extension.

You’ll use the features of the directory abstraction layer editor to add new definitions that model your own identity vault schema. You’ll use the features of the *Provisioning View* to deploy the new definitions to the identity vault.

Using the directory abstraction layer editor

The directory abstraction layer editor is divided into two panes. The left pane provides a view of the directory abstraction layer contents. When you select an item in the left pane, the right pane displays the attributes and settings for the selected item.



4.3 Working with entities and attributes

Any identity vault object that you want users to search, display, or edit in the Identity Manager user application must be defined as an *entity* in the directory abstraction layer. For example, to use the inetOrgPerson identity vault object in the user application, you must create an entity definition for it.

4.3.1 Steps for adding entities

Follow these steps to add entities to the directory abstraction layer:

Step	Task	For more information
1	Decide what identity vault objects you want to use in the user application	Section 4.3.2, “Analyzing your data needs,” on page 87
2	Use the directory abstraction layer editor to define the identity vault objects in the directory abstraction layer	Section 4.3.3, “Defining entities,” on page 87
3	Use the Provisioning View to validate the data definitions	Section 4.8, “Importing, validating, and deploying directory abstraction layer definitions,” on page 109

Step	Task	For more information
4	Deploy the definitions to the identity vault	Section 4.8.3, “About deploying,” on page 111
5	Update the application server’s cache to include the new abstraction layer definitions	Chapter 13, “Caching Configuration,” on page 207
6	Test the Identity Manager user application to ensure that your changes display properly	

4.3.2 Analyzing your data needs

To model your identity vault data in the directory abstraction layer, you’ll need to know:

- The *parts of the directory* you want to make available to the Identity Manager user application.

For example, the list of objects that the user can search and display. Check this list against the base set of abstraction layer definitions to determine what you need to add.

- The *structure of the schema* including custom extensions and auxiliary classes
- The *structure of the data* including:
  - What is required and what is optional
  - Validation rules
  - Relationships between objects (DN references)
  - How attributes are defined (for example, an attribute that represents a phone number might be multi-valued for home, office, and cell phone numbers)
- Who will see the data

  Is this a public or private site?

Once you have this information, you can use it to map your identity vault objects to abstraction layer entities.

NOTE: The eDirectory ACLs are applicable to all abstraction layer objects. Effective rights on objects and attributes are based on the authenticated user established at application login.

4.3.3 Defining entities

Depending on what you want to expose in the user application, you’ll be defining two kinds of entities:

- *Entities that are mapped from schema*. These entities represent objects that exist in the identity vault that are directly exposed to users in the user application. When defining this type of entity, you’ll expose all of the attributes that you’ll want your users to work with. Examples of this entity type include: User, Group, and Task Group.You can also create more than one entity definition for the same object if you want to expose different sets of attributes to different kinds of users. For more information, see “Creating multiple entity definitions for a single object” on page 88.
- *Entities that represent LDAP relationships*. This type of entity is known as a DNLookup and it is used by the user application to:

- Populate a list with the results of a DN search among related entities
- Maintain referential integrity across DN referenced attributes during updates and deletes

Entities that support DNLookups are used by the Org Chart portlet to determine relationships and are also used by the Search, Create, and Detail portlets to provide pop-up selection lists and DN contexts. Examples of this kind of entity include: Manager Lookup, Task Manager Lookup, and User Lookup. For more information, see "Using DNLookup control types" on page 98.

Creating multiple entity definitions for a single object

You can create more than one entity definition that represents the same identity vault object, but provides a different view of the data. Within the entity definitions you could:

- *Define different attributes* for each entity definition

OR

- *Define the same attributes*, but specify different access properties that control how the attributes are searched, viewed, edited or hidden

---

NOTE: The entity definitions can optionally include a filter to hide certain entities from the result set.

You could then use these different entity definitions in different parts of the user interface. For example, suppose that you wanted to create a directory of employees; one for a public site and one for an internal site. On the public site you wanted to supply first and last names and a phone number, but on the internal site, you wanted to list additional information like title, managers, and so on. Here's how you could accomplish this:

1. Create two entity definitions (with different keys).

   Both entity definitions expose the same identity vault object, but one entity definition key is public-staff-information, and the other is internal-staff-information.
2. Within each entity definition define a different set of attributes: one for public-staff-information, the other for internal-staff-information.
3. Use the Portal Administration tab of the Identity Manager user application to create a portlet instance for the public page, and another one for the internal page.

   For more information about creating portlet instances, see Chapter 9, "Portlet Administration," on page 171.

Procedures for creating entity definitions

When you have determined the entities and attributes that you want to expose, you can start adding them to the directory abstraction layer using the editor. You'll follow a set of steps like this:

Step	What to do	See this procedure
1.	Decide which set of files to start with.	
	• You want to add to the base set of definitions	Section 4.3.1, "Steps for adding entities," on page 86
	• You want to start with already deployed definitions	Section 4.8.1, "About importing," on page 109

Step	What to do	See this procedure
1a.	Some of the entities that you want to use are not part of the eDirectory base schema. Any extensions to the eDirectory schema will not show up automatically in the editor's list of selectable objects and attributes. This means that you have to update the designer's local schema file to include these custom objects and attributes.	"To update the list of available schema elements:" on page 89
2.	Add one or more entities to the directory abstraction layer	"Adding entities" on page 89
3.	Add attributes to the entities	"Adding attributes" on page 91

Updating the list of available schema elements

To update the list of available schema elements:

1. With the Identity Manager project open, select the Identity Vault, right-mouse and select *Live Operations>Import Schema*.
2. Choose *Import from eDirectory* and provide the specifications for the eDirectory host.
3. Click *Next*.
4. Select the classes and attributes that you want to import, and click *Finish*.

Adding entities

You can add an entity via the Add Entity Wizard (described next) or by clicking the *Add Entity* button from the editor's toolbar.

NOTE: When using the Add Entity button, you are prompted to select the object class of the entity you want to create. The editor automatically adds the required attributes to the entity. You can then use the Add Attribute dialog to complete the entity definition.

To add an entity using the Add Entity Wizard:

1. Launch the Add Entity Wizard in one of these ways:

   From the *Provisioning View*:

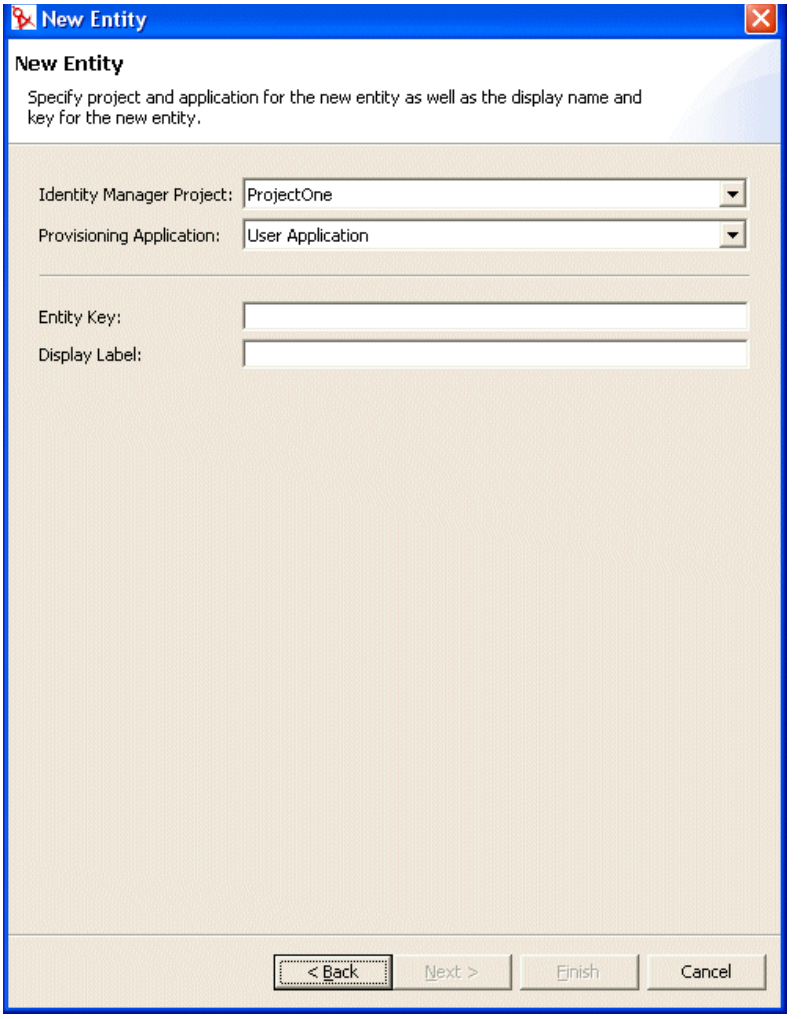
   - Select the *Entities* node, right-mouse click and choose *New*.
   - Select *File>New>Provisioning*. Choose *Directory Abstraction Layer Entity*. Click *Next*.

   From the directory abstraction layer editor:

   - Select the *Entities* node, right-mouse click and choose *New Entity-Attributes Wizard*.

     The New Entity dialog displays.

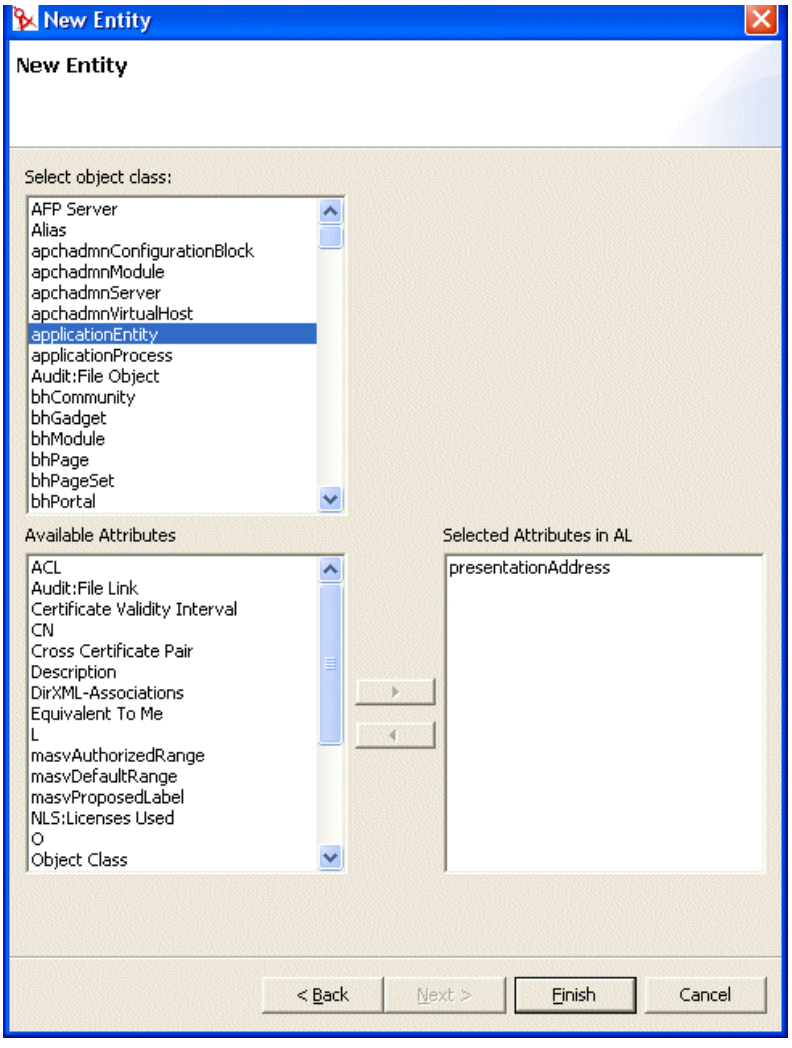
   NOTE: If launched from the File menu, the dialog contains fields not displayed when launched in either of the other ways. It is shown below.



2 Complete the panel as follows:

Field	Description
Identity Manager Project and Provisioning Application	Select the Identity Manager project and Provisioning Application where you want to add the entity and attributes. **NOTE:** These fields display when you launch the wizard from the File menu.
Entity Key	The unique identifier for the entity.
Display Label	The string displayed whenever this entity is referenced in the user interface.

3 Click *Next*. The New Entity dialog displays:



4 Choose the Object Class for the entity that you want to create, then select the attributes that you want from the Available Attributes list

---

TIP: If the object class of the entity that you want to create is not shown in the Available Object Classes list you might need to update the designer's local schema file. Follow the steps described in "To update the list of available schema elements:" on page 89.

---

5 Click *Finish*.

The property sheet is displayed for editing.

For more information, see "Entity property reference" on page 93.

NOTE: To make the attribute available to the user application, you must deploy the entity that contains the attribute.

Adding attributes

To add an attribute:

1 Select an entity.

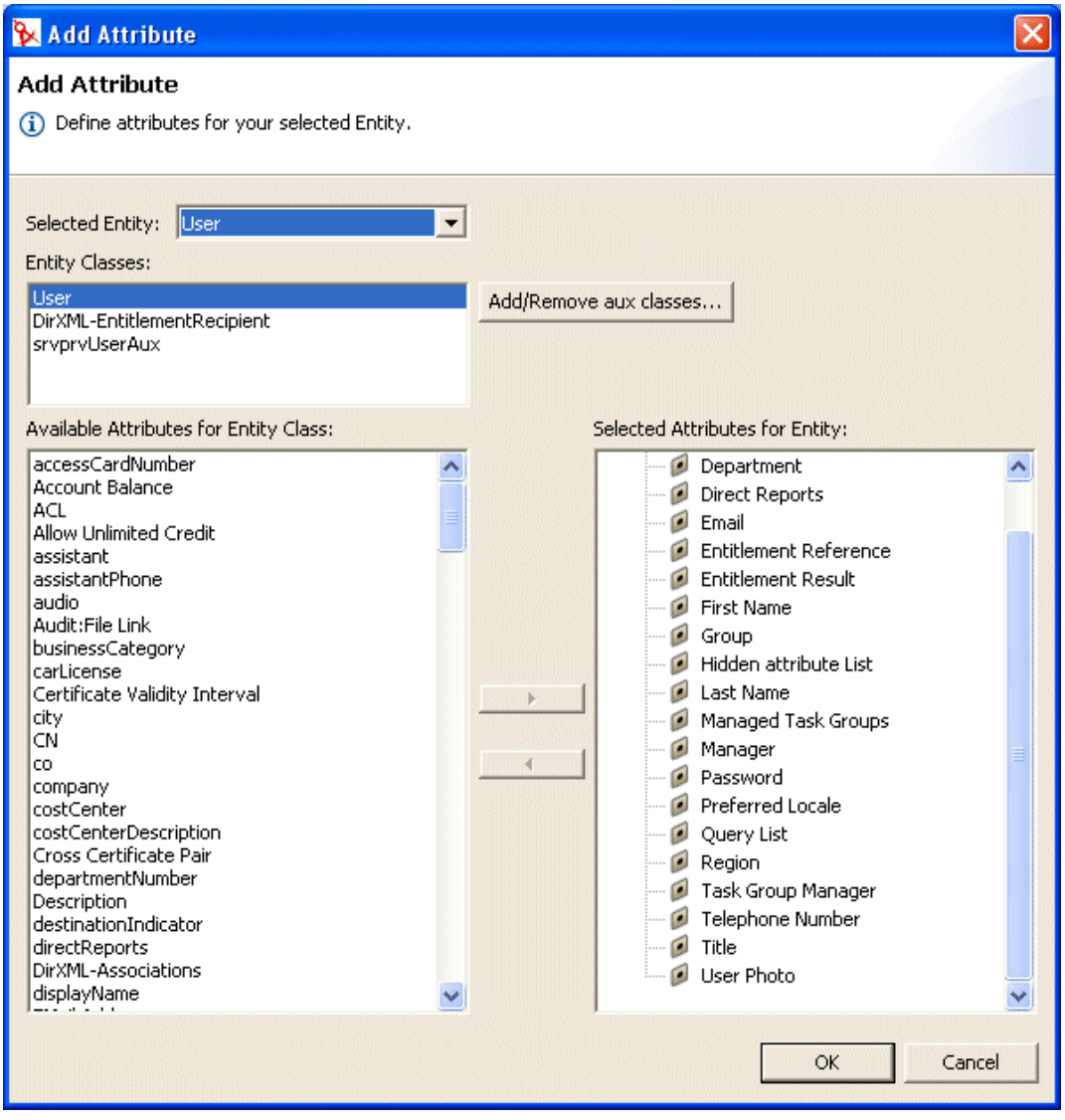
2 Add an attribute by:

- Right-clicking and selecting *Add Attribute*.

or

- Clicking the *Add Attribute* icon.

You are prompted as follows:



3 Choose the attribute from the *Available Attributes for Entity Class* list and add it to the *Selected Attributes for Entity* list.

TIP: If the attribute that you want to create is not shown in the Available Attributes from Entity Class list you might need to update the designer's local schema file. Follow the steps described in "To update the list of available schema elements:" on page 89.

---

4 Click *OK*.

The property sheet is displayed for editing.

For more information, see "Attribute property reference" on page 95.

NOTE: To make the attribute available to the user application, you must deploy it.

Entity property reference

You can set the following kinds of properties on entities:

Entity access properties

The *Access Properties* control how the user application interacts with the entity. They include:

Property	Description
Create	**Selected**—This object can be created by the user application.
Edit	**Deselected**—This object is not changeable by the user application regardless of the underlying ACLs. **Selected**—This object might be changeable, but the identity vault ACLs are used to determine this.
View	**Selected**—This object can be displayed by the user application.
Remove	**Selected**—This object can be deleted by the user application.

Entity required properties

The *Required* entity properties are:

Property name	Description
Key	The unique identifier for this entity. It defines the way the user application will reference this object.
Display Label	Defines how the object is shown in the user interface.
Class name	The Novell Directory Service (NDS) class name.
LDAP name	The LDAP object class name.
Search	**Selected**—This entity is searchable. Entities used in queries by identity portlets (such as Entity Search List or Entity Org Chart) must be selected (true).
Auxiliary Classes	A list of zero or more auxiliary classes for this entity. If adding auxiliary classes, you must specify the auxiliary class LDAP Name, NDS Name, and whether or not it can be searched.

Entity search properties

The *entity Search properties* are:

Property name	Description
Search container	The distinguished name of the LDAP node or container where searching starts (the search root). For example: `ou=sample,o=ourOrg` You can browse the identity vault to select the container, or you can use one of the predefined parameters described in “Using predefined parameters” on page 95.
Search scope	Specifies where the search occurs in relation to the search root. Values are: **<Default>**—This search scope is the same as selecting Containers and subcontainers. **Container**—Search occurs in the search root DN and all entries at the search root level. **Container and subcontainers**—Search occurs in the search root DN and all subcontainers. This is the same as selecting **<Default>.** **Object**—Limits the search to the object specified. This search is used to verify the existence of the specified object.
Search Time Limit [ms]	Specify a value in milliseconds or specify 0 for no time limit.
Max Search Entries	Specify the maximum number of search result entries you want returned for a search. Specify 0 if you want to use the runtime setting. Recommendations: Set **between 100 and 200** for greatest efficiency Do **not** set over 1000

Entity create and edit properties

The *entity Create and Edit Properties* are:

Property name	Definition
Create Container	The name of the container where a new entity of this type is created. You can browse the identity vault to select the container, or you can use one of the predefined parameters described in “Using predefined parameters” on page 95. If this value is not specified, then the Create portlet will prompt the user to specify a container for the new object. The portlet will use the search-root specified in the entity definition as the base and allow the user to drill down from there. If there is no search-root specified in the entity definition then it will use the root DN specified during the user application installation.

Property name	Definition
Naming Attribute	The naming attribute of the entity (the Relative Distinguished Name (RDN)). This value is only necessary for entities where the access parameter Create is selected.
Alternate Edit Entity	The attributes of the Edit Entity are displayed in the edit mode of the Detail portlet. Choose an entity from the dropdown or <None> if this entity is not displayed by the Detail portlet.

Password Management properties

The *Password Management Properties* are:

Property name	Definition
Password Attribute	Choose the attribute where the password for this entity will be stored.
Password required when attribute is created	**Selected**—Means a password is required when this entity is created.

Using predefined parameters

The directory abstraction layer editor allows you to use predefined parameters for certain values. The parameters are:

Predefined parameter	Description
%driver-root%	Represents the Provisioning Driver DN. This value is specified during the user application configuration during installation or a later configuration. It is stored in the user application's realm configuration.
%user-root%	Represents the User Container DN. This value is specified during the user application configuration during installation or a later configuration. It is stored in the user application's realm configuration.
%group-root%	Represents the Group Container DN.This value is specified during the user application configuration during installation or a later configuration. It is stored in the user application's realm configuration.

Attribute property reference

You can set the following kinds of properties on attributes:

- "Attribute access properties" on page 95
- "Attribute required properties" on page 96
- "Attribute filter and format properties" on page 97
- "Attribute UI control properties" on page 97

Attribute access properties

The *attribute access properties* are:

Name	Description
Edit	**Selected**—This attribute can be edited/modified by the user application. Even if it is selected (true), the attribute might still not be editable if the underlying identity vault ACLs/effective rights prevent it.
Enable	**Deselected**—This attribute cannot be used by the user application. It is the same as removing the entry from the file.
Hide	Controls whether the Hide check box in the user application is enabled or disabled. The Hide check box allows users to control whether an attribute (such as their photo) is displayed by the application. **Deselected**—The Hide check box is disabled for this attribute, so the user cannot choose to hide this attribute. **Selected**—The Hide check box can be enabled in the user application. However, the following must also be true of the logged-in user. They: • Are either the owner of the attribute or a User Application Administrator. • Have Trustee rights to update the srvprvHideAttributes attribute on the identity vault. If these requirements are not met, then the Hide check box is disabled in the user interface even if this setting is selected (true). **TIP:** When a user hides an attribute that contains an image, users who have viewed the image might continue to see it until their browser cache is refreshed.
Multivalue	Specifies whether this attribute can be multivalued, for example, a phone number. **Selected**—the attribute can be multivalued.
Read	**Selected**—The user application can query this attribute. For most attributes this should be selected (true), but for some attributes, like password, it should be deselected.
Require	**Selected**—the attribute must be supplied.
Search	**Selected**—The user application can search on this attribute. Attributes that will be used in queries by identity portlets (such as Entity Search List or Entity Org Chart) must be selected. **TIP:** If an attribute used in a search is also indexed in eDirectory the search will be faster.
View	**Selected**—The user application can display this attribute. In most cases this would be true, but for some attributes, like password, it would probably be deselected.

Attribute required properties

Name	Description
Key	The unique identifier for the attribute.
Display Label	The label that is displayed in the user application.
Attribute Name	The NDS name for this attribute.

Name	Description
LDAP Name	The LDAP name for this attribute.

Attribute filter and format properties

Name	Description
Filter: WHERE Attribute	Lets you specify an LDAP filter on the identity vault search for this attribute.
Enable	**Selected**—Enables the filter.

Attribute UI control properties

Name	Description
Data Type	Choose a data type from the following list: • Binary • Boolean • DN • Integer • LocalizedString • String • Time
Format Type	Used by the user application to format data. Format types include: • None • AOL IM • Email • Groupwise IM • Image • Phone Number • Yahoo IM • Image URL • Date • DateTime The Format Types are dependent on the data type. For example, a Time data type can only be associated with Date and DateTime formats.

Name	Description
Control Type	Types include: **DNLookup**—Defines that this attribute contains a DN reference. Use when you want to: • Populate a list with the results of a DN search among related entities • Maintain referential integrity across DN referenced attributes during updates and deletes The user application use this information to generate special user interface elements, and to perform optimized searches based on the DNLookup definition. For more information, see "Using DNLookup control types" on page 98 **Global List**—Display this attribute as a dropdown list whose contents are defined in a file outside of this attribute definition. For more information, see Section 4.4, "Working with lists," on page 101. **Local List**—Display this attribute as a dropdown list whose contents are defined with this attribute. To define a local list: 1. With the attribute selected, set the control type to Local List.  2. Click the Add button to add more values. Use the up and down arrow buttons to change the position of the item in the list. In the Value column, type the value to write to the identity vault. It can only include lowercase letters, numbers, and underscore (_) characters. 3. In the Labels column, type the text you want displayed in the user interface. **Range**—Use the Range control type with Integer data types to restrict user input to a sequential range of values. You'll supply the range's start and end values.

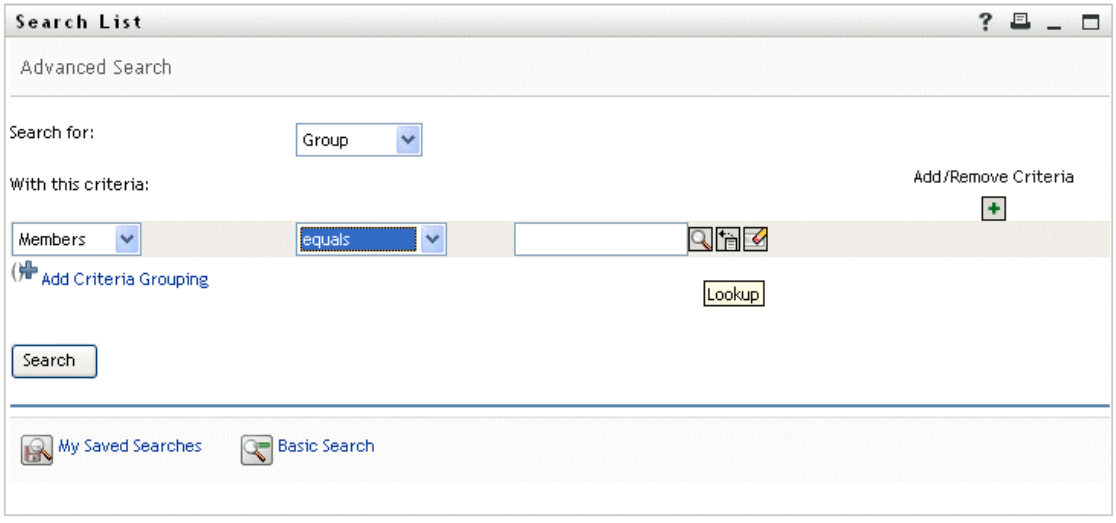
Using DNLookup control types

When you define a control type as a DNLookup, it means that:

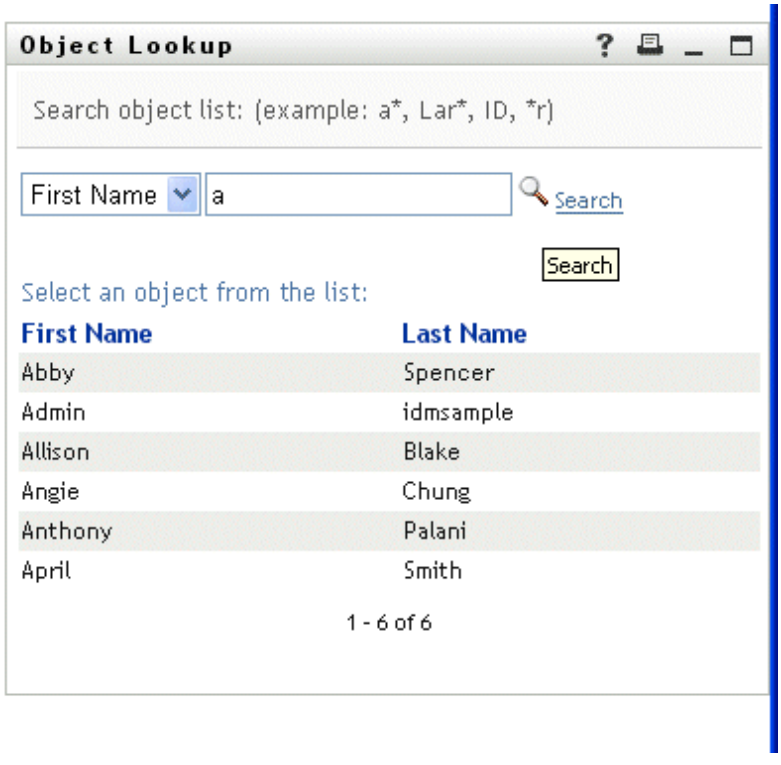
- Users can select from a list of possible values when searching on this attribute.
- When this attribute is created, populated, or deleted an attribute on a related entity will be updated appropriately depending on the user action (create, delete, update) to maintain referential integrity.

DNLookups for selection lists

The installed user application contains entity definitions for Users and Groups. The Users entity definition contains an attribute called Group which is defined as a DNLookup control type. This enables any identity portlet to provide a selection list of groups for a particular user. For example, a user chooses to do a Directory Search. They want to find a user in a group, but they do not know the group name. They would select User as the object to search for and include Group as a search criteria as shown here:



Because Group is defined as a DNLookup control type for the User entity, the Lookup icon displays. If the user selects it, then a list of possible groups displays:



The user can select a group from the list.

DNLookups for referential integrity

DNLookups for updates and synchronization are important because LDAP allows group relationships to map in both directions. For example, your data might be set up so that:

- User object contains a group attribute. The group attribute:
- Is multi-valued
- Lists all of the groups to which a user belongs
- Group object contains a user attribute. The user attribute:
- Is multi-valued
- Lists all of the users that belong to the group

This means that you can have an attribute on the user object that shows all the groups a user belongs to, and on the Group object you have a DN attribute that includes all the members of that group.

When the user requests an update, the user application must honor the relationships and ensure that the target and source attributes are synchronized. In the DNLookup, you'll specify both attributes that must be synchronized. You can use this technique to provide synchronization between any objects that are related not just group structural objects. You create this kind of DNLookup control type by specifying the advanced DNLookup properties described in the *DNLookup Relational Integrity properties* reference.

DNLookup property reference

The DNLookup Display properties are:

Field	Definition
Lookup Entity	The name of the entity to search, for example, the Task Group entity contains an attribute for Task Manager. To populate that field, you'd need to know which users are Task Managers.
Detail entity	The key of the entity whose details you want displayed if the user requests more information by clicking a hypertext link in the user application. When you define a DNLookup the identity portlets are able to provide a hypertext link that allows users to display the details of the linked object.
Attributes to display	Choose one or more attributes to display when the search is complete.
Perform Automatic Query	Defines how the **Attributes to display** (above) are displayed. • **Selected**—Performs an automatic query of the entity and presents the results in a selectable list. You might not want to choose this option if the data returned will be a large number because it will force the user to scroll through a large result set. • **Deselected**—allows the user to specify the search criteria for the entity query, then presents the results in a selectable list.

DNLookup Relational Integrity properties—These properties are used for synchronizing data between two objects such as groups and group members.

Property	Definition
Source attributes to update	Name of the attribute to update. The attribute must contain a DN reference to the **Target attributes to update**. This is required to synchronize attributes on two different objects.
Target attributes to update	Name of the attribute that must be updated along with the **Source attributes to update**.This is an LDAP attribute name. This is required to synchronize attributes on two different objects. The attribute must contain a DN reference.
Target auxiliary classes, if any	Name of the auxiliary class that contains the **Target attributes to update**.

4.4 Working with lists

The lists node lets you define the contents of global lists. Global lists are used by the Identity Manager user application to:

- Provide a list of values for an attribute. When the attribute is displayed for editing in the user interface, the possible values are displayed as a dropdown list.
- Used to define the categories available to the Provisioning Request Configuration plug-in to iManager. This is a special list. For details, see Section 4.4.2, “About the Provisioning Category list,” on page 103.

To create a new global list:

1 Launch the New List Wizard in one of these ways:

From the *Provisioning View*:

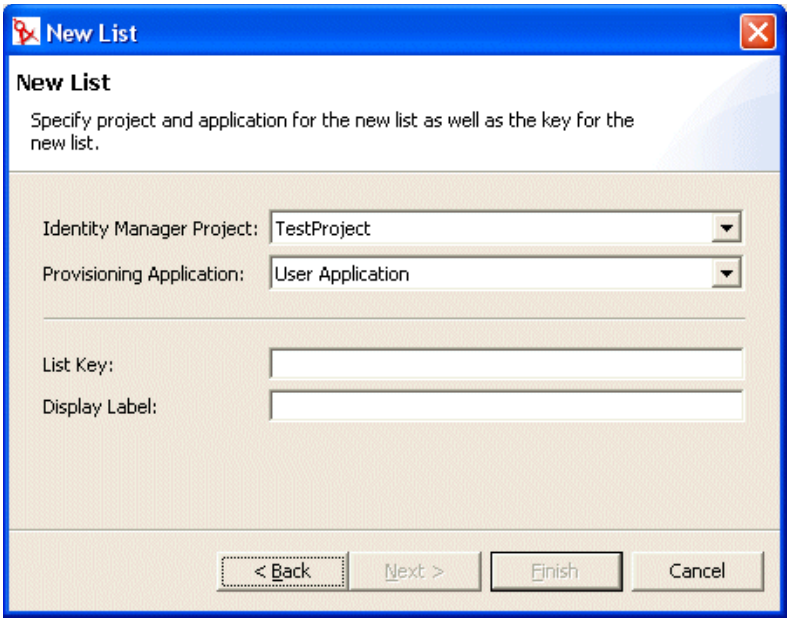
- Select *File>New>Provisioning*. Choose *Directory Abstraction Layer List*. Click *Next*.
- Select the *Lists* node, right-mouse click and choose *New*.

From the *directory abstraction layer editor*:

- Click the *New List* button.
- Select the Lists node, right-mouse click and choose *Add List*.

The New List dialog displays.

NOTE: If launched from the File menu, the dialog contains fields not displayed when launched in either of the other ways.



2 Complete the panel as follows:

Field	Description
Identity Manager Project and Provisioning Application	Select the Identity Manager project and provisioning application where you want to add the entity and attributes. **NOTE:** These fields display when you launch the wizard from the File menu.
List Key	The unique identifier for the list.
Display Label	The string used whenever this list is referenced in the user interface.

3 Click *Finish*. The Global Lists property sheet displays.

4 Complete these fields:

Field	Description
Display Label	The name of this list as shown in the designer.
Labels	The text for the list item that you want displayed in the user interface.
Values	The value for the list item that you want stored in the identity vault. It can only include lowercase letters, numbers, and underscore (_) characters.

The list is now available in the design environment.

5 Save the project.

NOTE: To make the list available to the runtime environment, you must deploy it.

4.4.1 About the Preferred Locale list

The Preferred Locale list represents the default language that will be used in the event that the browser language is not one of the supported languages. The contents of this list are displayed by the default configuration of the Edit User action in the user application.

4.4.2 About the Provisioning Category list

The Provisioning Category list defines the set of categories that help you organize Provisioned Resources (Entitlements) and Provisioning Requests. The categories in this list display in:

- *iManager*—Provisioning Request Configuration plug-in
- *user application*—Requests and Approvals tab

You cannot change the Provisioning Request list key, but you can add more items to the list or change the existing category values and labels.

To modify the contents of the Provisioning Category List:

1. Make sure the correct project is open in the editor.
2. Click the Lists Node.
3. Select *Provisioning Category*.
4. Use the global list property pane to make your modifications.

   NOTE: The Values field is used to populate the category key. The Values field restricts you to lowercase letters, numbers and underscore (_)characters because these are the only valid characters in the category key. The category key is used internally as an identifier for the category.

5. Save then deploy your changes. Remember to update the application server's cache.

   Once your changes are deployed, they are reflected in the user application and the iManager plug-in.

4.5 Working with Org Chart relationships

The Org Chart Relationships node allows you to define hierarchical relationships between entities defined in the directory abstraction layer. The relationship can be between like entities (such as user/user) or unlike entities (such as user/device).

The following relationships are defined for the user application:

- Group's membership
- Manager-Employee
- User groups

To successfully deploy a relationship, all of the components (entities and attributes) of the relationship must already be deployed.

To create a new relationship:

1. You can create a new relationship in any of these ways:

   From the *Provisioning View*:

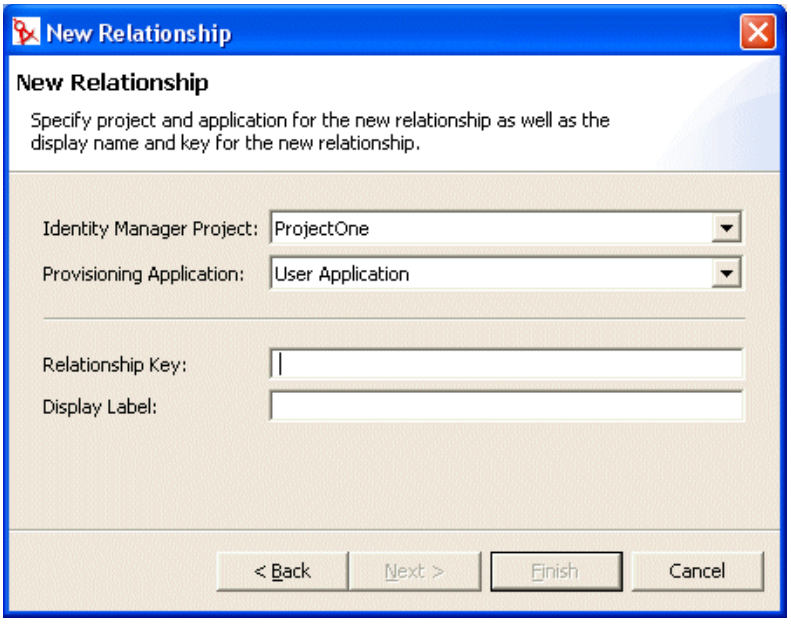
   - Select *File>New>Provisioning*. Choose *Directory Abstraction Layer Relationship* and click *Next*.
   - Select the *Org Chart Relationships* node right-mouse, and choose *Add*.

   From the *directory abstraction layer editor*:

   - Click the *Add Relationship* button.
   - Select the *Org Chart Relationships* node right-mouse and choose *Add Relationship*.

   The New Relationship dialog displays.

   NOTE: When launched from the File menu, the dialog contains fields not displayed when launched in either of the other ways.

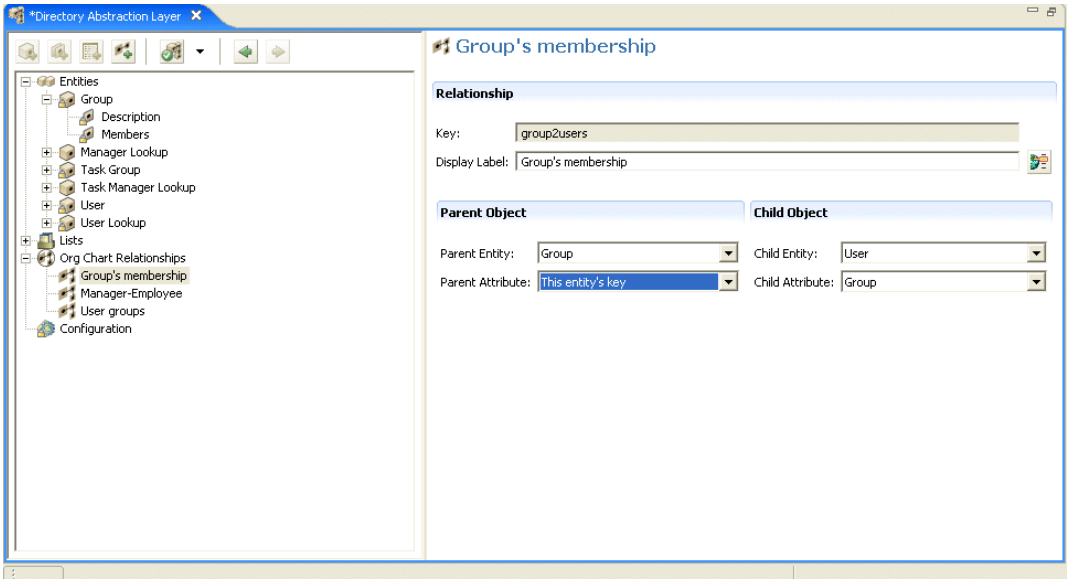


2 Complete the panel as follows:

Field	What to do
Identity Manager Project and Provisioning Application	Make sure the correct Identity Manager project and Provisioning Applications are selected. **NOTE:** This field displays when you create relationships from the File menu.
Relationship Key	Type a unique value for the relationship key.
Display Label	Type the string that you want displayed any time the relationship is shown in the Identity Manager user interface.

3 Click *Finish*.

The relationship is created and its property sheet is opened for editing.



4.5.1 Relationship properties reference

Field	Description
Key	The read-only unique identifier for the relationship. **TIP:** You'll specify this value in the Org Chart Portlet preference sheet.
Display Label	Specify a name to display when this relationship is referenced by other identity portlets. For example, this value is displayed when users click the Choose Org Chart icon from the Detail portlet. Click **Localize** to provide the translation for the display label text.
Parent entity	Choose an entity from the dropdown list. The entity that you choose becomes the parent object in the organization chart hierarchy. For example, in a Manager-Employee relationship, the Parent Entity would be User. For a Group-Member relationship, the Parent Entity would be Group. **Directory abstraction layer requirements**—The entities in this list are a subset of the entities defined in the directory abstraction layer. Parent entities must have the view access property selected (true)
Parent attribute	Choose an attribute from the dropdown list. This attribute is used to find matching child entities. When the value of this attribute matches a corresponding value on an attribute of the child entity (see Child attribute below), then a relationship can be established. **Directory abstraction layer requirements**—This list of attributes is populated using the selected Parent Entity's attributes. It includes only the attributes defined as the DNLookup control type

Field	Description
Child entity	Choose the entity that will be the child object in the hierarchy. For instance in a Manager-Employee relationship it would be user. For an Employee-Resources relationship it would be Devices. This entity must contain the attribute that is related to the Parent attribute.
Child attribute	Choose the attribute that matches the Parent Attribute. This specifies the attribute on the child entity to use to find matching parent entities. When the value of this attribute matches a corresponding value on an attribute of the parent entity (see Parent Attribute above), then a relationship can be established.

NOTE: Dynamic groups are not fully supported by the Org Chart portlet. You cannot define a dynamic group as the Parent entity in a relationship, but you can define a dynamic group as the child entity in a relationship.

To delete a relationship:

1. Select the relationship you want to delete.
2. Right-mouse click and choose *Delete*.

4.6 Working with configuration settings

The Configuration node allows you to set general configuration properties for the user application. They include:

Property	Description
Default 'My Profile' Entity	Defines the entity to display when the user clicks **My Profile** in the user interface. This field is restricted to show only entities whose object class is user (or LDAP inetOrgPerson).
Default Locale	Defines the default language that will be used for the display labels in the user application. If the browser is set to an unsupported language this locale is used instead. **NOTE:** The browser locale will override the default locale for the supported languages.
Container Classes	This provides the Create User or Group action with the contents of a selection list of container classes. The user selects a container from the selection list as the location where the newly created object will reside.

4.7 Localizing display text

The directory abstraction layer editor provides an easy way to localize the display text for:

- Entity and attribute display labels

- Org chart relationship names
- Global and local list items

4.7.1 Supported languages

You can localize the display text in one or more of these languages:

- English
- French
- German
- Italian
- Japanese
- Korean
- Portuguese
- Russian
- Simplified Chinese
- Spanish
- Traditional Chinese

4.7.2 Localizing text

The directory abstraction layer editor provides several different ways to localize abstraction layer definitions. You can access the localization dialogs in these ways:

To define the localization text for	Action
Every localizable item in the directory abstraction layer	• Click **Set Global Localization (**on the directory abstraction layer editor toolbar). Make sure to select the Target Language before entering the localized text in the target field.
A specific entity, relationship or list	• From the directory abstraction layer editor tree view, select the object to localize. • Right-mouse and select **Localize**. Make sure to select the Target Language before entering the localized text in the target field.
A single display label	• Select a specific entity or attribute. • Click **Localize Display Label** (beside the Display Label field in the property pane).

The dialogs might look a little different, but each contains these fields:

- *Origin*—This is typically the object type (such as entity, list, or relationship) and key
- *Source*—The text to translate (display label)
- *Target Language*—One of the supported languages
- *Target*—The translation text

4.8 Importing, validating, and deploying directory abstraction layer definitions

Importing, validating, and deploying directory abstraction layer definitions are actions performed from the Provisioning View of the designer.

4.8.1 About importing

The import feature lets you import a set of existing definitions. You’ll want to use import when:

- You want to begin a new project based on a deployed project.
- You want to share definitions with other developers working on the same project. For example, another developer adds an attribute to the user entity, or adds a new global list. If the developer deploys the new definition to the identity vault, you can import it, and ensure that you are both working with identical definitions.

To import existing definitions:

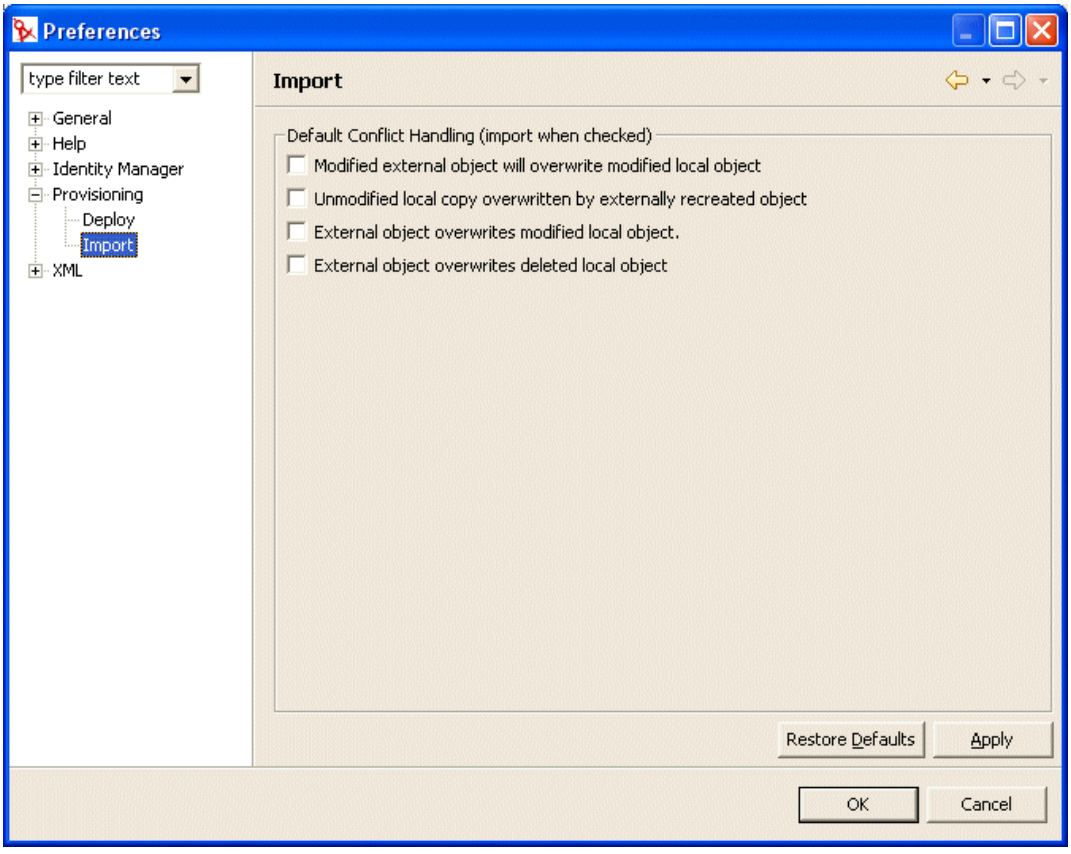
1. Open the *Provisioning View*.
2. Determine whether you want to import:
   - A complete set of definitions
   - A set of one definition type, such as all entities or all relationships.
   - A specific object (such as the User entity)
3. To import:
   - A specific object, select it from the list, right-mouse and select *Import Object*.
   - A complete set of definitions, select the Directory Abstraction Layer node, right-mouse and select *Import All* or *Import Object*.
4. Click the eDirectory Browse icon and navigate to the DirectoryModel node and select the object(s) to import then click *OK*.
   - If the objects match, then you are notified that there are no differences and the import does not proceed.
   - If the objects do not match, you are able to confirm which object(s) to import. Review the items selected for import, make any changes needed then click *OK*.

Setting import preferences

Import preferences let you specify how you want the designer to resolve conflicts between the data in the identity vault and your local directory abstraction layer files. These conflicts can arise because different users and tools have access to the identity vault’s directory abstraction layer definitions. The definitions can be changed by other administrators or developers using iManager tools or their own local designer-based project. When conflicts arise between the definitions on your local file system and the identity vault, these preferences allow you to specify how the conflicts are handled.

To set Import preferences:

1. Choose *Window>Preferences*.
2. Open the Provisioning node of the tree and click *Import*.



3. Choose the preferences:

Preference	Description
Modified external object will overwrite modified local object	Both the local file and the identity vault definitions contain changes. The local changes have not yet been deployed. Select this option if you want the identity vault object to overwrite the changes that you've made to the local file.
Unmodified local copy overwritten by externally recreated object	The identity vault object was deleted and then recreated. The local file set includes the original definition with no changes. Select this option if you want the import to overwrite the local copy.
External object overwrites modified local object	The local file contains changes not deployed to the identity vault. Select this option if you want the local files to be overwritten on import.

Preference	Description
External object overwrites deleted local object	You have deleted a definition locally, but have not deployed the changes. This means that the object still exists in the identity vault. Select this option if you want identity vault objects to be copied to the local file system. If you choose this option, you will lose your undeployed changes.

4.8.2 About validation

You can validate the directory abstraction layer data definitions on the local files system before you attempt to deploy them. The validation:

- Verifies that the XML is well-formed and complies with the schema that defines the elements needed for entities, attributes, lists, relationships, and so on.
- Checks every entity to ensure that references to other entities and global lists are valid.

For example when validating an entity and its attributes, the validator checks that all references to other entities via the *Edit Entity*, *DN Lookup*, and *Detail Entity* fields reference entities that actually exist.

- Ensures that every entity has at least one attribute defined.
- Ensures that every local and global list contains at least one item.

You can selectively validate definitions from the *Provisioning View*. To validate:

- All of the items within a node, select the node, right-mouse and select *Validate*.
- A single object within a node, select the object, right-mouse and select *Validate*.

You can validate all of the definitions by clicking the *Validate Abstraction Layer* button from the directory abstraction layer toolbar.

NOTE: The validation does not check the identity vault for the existence of any objects.

4.8.3 About deploying

You must deploy your definitions to an identity vault before you will see the resulting changes within the Identity Manager user application.

To deploy a set of definitions to an identity vault:

1. Save all of the changes that you've made using the directory abstraction layer editor.

   If you do not save your changes before attempting the deploy, the editor display a dialog that shows the definitions that have not been saved. It prompts you to save the most recent changes. If you do not to save the changes, the object is still deployed to the server but what is deployed does not include the unsaved changes. Choosing not to save the changes does not cancel the deployment.
2. Open the *Provisioning View*.

3 Decide if you want to deploy all of the objects defined using the directory abstraction layer editor or a subset.

- To deploy all:

  Select the root node, right-mouse and choose *Deploy All*

- To deploy a specific entity, relationships, list, or configuration setting:

  Select it, right-mouse and choose *Deploy object*.

You might be prompted for identity vault credentials. The editor performs a validation and displays any validation messages in a dialog. Respond to the validation messages by selecting/deselecting the items to deploy. After you've made your deployment selections and submitted them, you are notified on the deployment's success or failure.

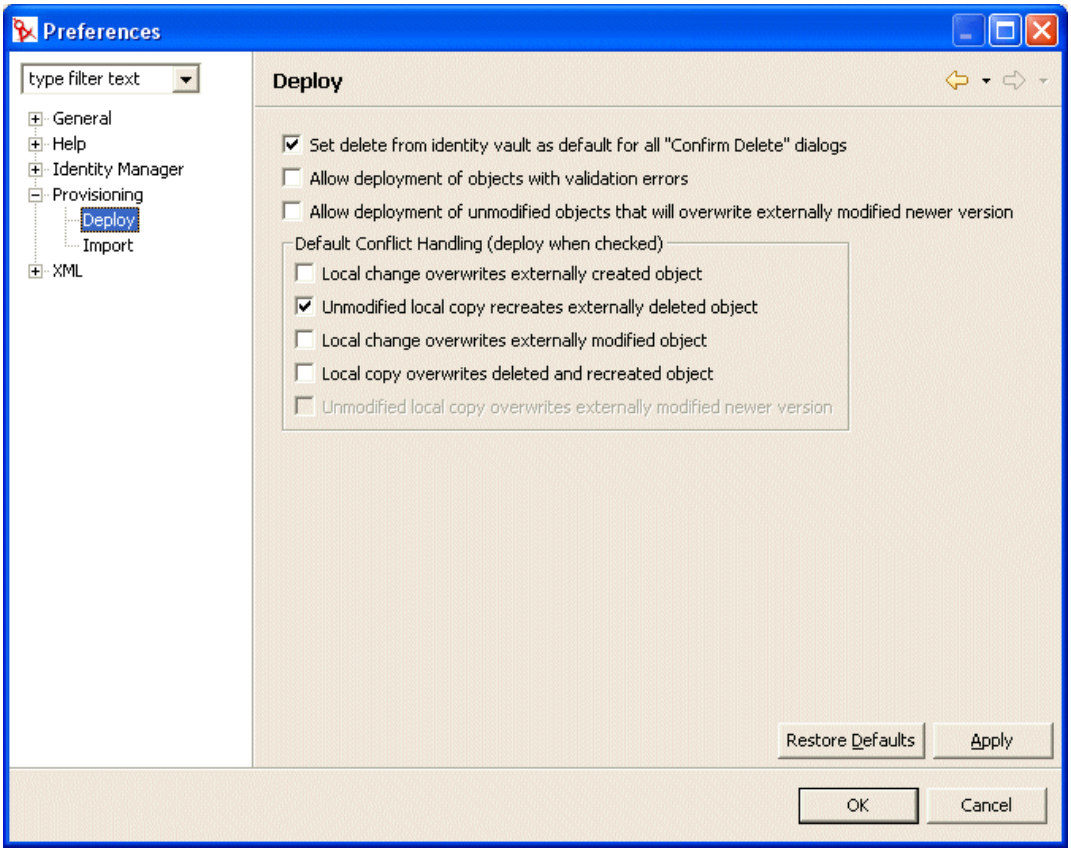
Setting deployment preferences

Deploy preferences let you specify how you want the designer to resolve conflicts between the data in the identity vault and your local directory abstraction layer files. Conflicts might arise because other users have deployed changes to the identity vault and these changes are not reflected in the definitions on your local file system. To ensure that conflicts are handled the way you want them to be, you can set preferences specifying the conflict resolution.

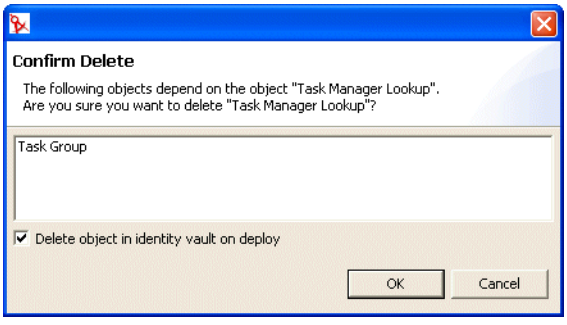
To set Deploy preferences:

1 Choose *Window>Preferences*.

2 Open the Provisioning node of the tree and click *Deploy*.



3 Specify general deploy preferences:

Preference	Description
Set delete from identity vault as default for all “Confirm Delete” dialogs	If you attempt to delete an object in the Provisioning view or the directory abstraction layer editor, you are prompted to confirm the deletion with a dialog like this:  This preference determines whether the delete confirmation dialog check box labeled **Delete object in identity vault on deploy** is selected by default. To select this preference means the default is to always delete the identity vault object. The local object is always deleted.
Allow deployment of objects with validation errors	**Select**—Select this option if you want to deploy objects that fail validation. At deploy, the designer validates the definitions that are being deployed following the validation rules outlined in Section 4.8, “Importing, validating, and deploying directory abstraction layer definitions,” on page 109. **Deselect**—To prevent deployment of definitions that fail validation.
Allow deployment of unmodified objects that will overwrite externally modified newer version	**Select**—If your local files have not been changed, but the identity vault objects have. Do you want the local files to overwrite the identity vault files? If so, then select this preference. **Deselect**—If you want to keep the newer identity vault versions. When selected, you can set this as the default behavior by also selecting the conflict resolution preference **Unmodified local copy overwrites externally modified newer version**.

4 Specify conflict resolution preferences:

Preference	Description
Local change overwrites externally created object	**Select**—If you want the object you are deploying to overwrite the object that is in the identity vault. **Deselect**—Deploy does not occur when this conflict occurs.

Preference	Description
Unmodified local copy recreates externally deleted object	**Select—**If you want the local object you are deploying to create an object that was already deleted from the identity vault. **Deselect**—Deploy does not occur when this conflict occurs.
Local change overwrites externally modified object	**Select—**If you want the local definition to always be deployed even if the identity vault has been changed by another user. **Deselect**—Deploy does not occur when this conflict occurs.
Local copy overwrites deleted and recreated object	**Select—**If you want the local object to always be deployed even if the identity vault object has been deleted or has been deleted and recreated. **Deselect**—Deploy does not occur when this conflict occurs.
Unmodified local copy overwrites externally modified newer version	This preference can only be set when the general deploy preference **Allow deployment of unmodified objects that will overwrite externally modified newer version** is selected. **Select**—If your local files have not been changed, but the identity vault objects have changed and you **always** want the local files to overwrite the identity vault files as the default behavior. **Deselect**—If you want to keep the newer identity vault versions.

Setting up Logging

5

This chapter includes the following:

5.1 About event logging

The Identity Manager user application implements logging by using *log4j*, an open-source logging package distributed by The Apache Software Foundation. By default, event messages are logged to the *system console* and to the application server’s log file at logging level INFO and above. You can also configure the user application to log to Novell Audit. Events are logged to *all* activated loggers.

IMPORTANT: If you are logging to Novell Audit, it is recommended that you review the Novell Audit documentation (http://www.novell.com/documentation/nsureaudit).

5.1.1 About the log level settings

Console logging involves synchronized writes. This means that logging can become a processor usage issue as well as a concurrency impedance. You can change the priority value default setting to ERROR, by modifying the setting in the <*installdir*>/jboss/server/IDMProv/conf/log4j.xml. Locate the root node that looks like this:

```
<root>
      <appender-ref ref="CONSOLE"/>
      <appender-ref ref="FILE"/>
   </root>
```

Change the priority value to:

```
<root>
      <priority value="ERROR"/>
      <appender-ref ref="FILE"/>
</root>
```

Assigning a value to the root ensures that any appenders that do not explicitly have a level assigned inherit the root's level. By default, the file appender does not have a threshold level assigned so it assumes the root's. Any appender included in the root that does have a level threshold should be ERROR or WARN. Error level settings at more than WARN will impact performance.

5.2 Logging to a Novell Audit server

To log to an Novell Audit server, follow these steps:

Step	What to do	For more information
1	Add the Identity Manager application schema to the Novell Audit server as a log application	Section 5.2.1, “Adding the Identity Manager application schema to your Novell Audit server as a log application,” on page 116
2	Configure the Novell Audit **platform agent** on your application server	The Platform Agent is required on any client that reports events to Novell Audit. You configure the platform agent through the **logevent configuration** file. This file provides the configuration information that the platform agent needs to communicate with the Novell Audit server. The default location for this file, on the application server, is: • Linux—/etc/logevent.conf • Windows—/<WindowsDir>/logevent.cfg (Usually c:\windows) Make sure to specify the **IP address or DNS name of your Novell Audit server** in the **LogHost** setting. For example: `LogHost=xxx.xxx.xxx.xxx` Specify any other settings needed for your environment. **IMPORTANT:** After you create or modify the logevent configuration file, you must restart the JBoss application server before those changes will take effect. For more information about the structure of the logevent configuration file, see the section on configuring platform agents (http://www.novell.com/documentation/nsureaudit) in the chapter on the logging system in the Novell Audit Administration Guide.
3	Enable Novell Audit logging	Section 5.2.2, “Enabling Audit logging,” on page 117

5.2.1 Adding the Identity Manager application schema to your Novell Audit server as a log application

To configure Audit to use the Identity Manager user application as a log application, follow these steps:

1. Locate the following file:

   `DirXML.lsc`

Platform	Location
Linux	Post installation: `/opt/novell/naudit/logschema/dirxml.lsc`
Windows	On the installation media: `/nt/dirxml/nsure_audit/nauditextensions/lsc/dirxml.lsc`

2. Use a Web browser to access *iManager*, and log in as an *administrator*.
3. Go to *Roles and Tasks> Auditing and Logging* and select *Logging Server Options*.
4. Browse to the *Logging Services container* in your tree and select the appropriate *Audit Secure Logging Server*. Then click *OK*.
5. Go to the *Log Applications* tab, then select the appropriate *Container Name*, and click the *New Log Application* link.
6. When the New Log Application dialog displays, specify the following:

For this setting	Do this
Log Application Name	Type any name that is meaningful for your environment
Import LSC File	Use the **Browse** button to select the **DirXML.lsc** file

   Then click *OK*. The Log Applications tab displays the added application name.
7. Click *OK* to complete your Novell Audit server configuration.
8. Make sure the status on the Log Application is set to ON. (The circle under the status should be green. If it is red, click it to switch it to ON.)
9. *Restart* the Novell Audit server to activate the new log application settings.

5.2.2 Enabling Audit logging

To enable Novell Audit logging in your Identity Manager user application

1. Log in to the user application as the admin user.
2. Select the *Administration* tab.
3. Select the *Logging* tab.
4. Check the *Also send logging messages to Audit* check box (near the bottom of the tab).
5. To persist the changes for any subsequent application server restarts, make sure *Persist the logging changes* is selected.

5.2.3 What events get logged

The Identity Manager user application logs a set of events automatically from workflow, search, detail, and password requests. By default, the Identity Manager user application automatically logs the following events to all active logging channels:

Event ID	Process	Event	Severity
31400	Detail portlet	Delete_Entity	Info
31401		Update_Entity	Info
31410	Change Password portlet	Change_Password_Failure	Error
31411		Change_Password_Success	Info
31420	Forgot Password portlet	Forgot_Password_Change_Failure	Error
31421		Forgot_Password_Change_Success	Info
31430	Search portlet	Search_Request	Info
31431		Search_Saved	Info
31440	Create portlet	Create_Entity	Info
31520	Workflow	Workflow_Error	Error
31521		Workflow_Started	Info
31522		Workflow_Forwarded	Info
31523		Workflow_Reassigned	Info
31524		Workflow_Approved	Info
31525		Workflow_Refused	Info
31526		Workflow_Ended	Info
31527		Workflow_Claimed	Info
31528		Workflow_Unclaimed	Info
31529		Workflow_Denied	Info
3152A		Workflow_Completed	Info
3152B		Workflow_Timedout	Info
3152C		User_Message	Info
31533		Workflow_Retracted	Info
3152D	Provisioning	Provision_Error	Error
3152E		Provision_Submitted	Info
3152F		Provision_Success	Info
31530		Provision_Failure	Error
31531		Provision_Granted	Info
31532		Provision_Revoked	Info

Event ID	Process	Event	Severity
31450	Security Context	Create_Proxy_Definition_Success	Info
31451		Create_Proxy_Definition_Failure	Error
31452		Update_Proxy_Definition_Success	Info
31453		Update_Proxy_Definition_Failure	Error
31454		Delete_Proxy_Definition_Success	Info
31455		Delete_Proxy_Definition_Failure	Error
31456		Create_Delegatee_Definition_Success	Info
31457		Create_Delegatee_Definition_Failure	Error
31458		Update_Delegatee_Definition_Success	Info
31459		Update_Delegatee_Definition_Failure	Error
3145A		Delete_Delegatee_Definition_Success	Info
3145B		Delete_Delegatee_Definition_Failure	Error
3145C		Create_Availability_Success	Info
3145D		Create_Availability_Failure	Error
3145E		Delete_Availability_Success	Info
3145F		Delete_Availability_Failure	Error

5.2.4 Log reports

If you log events to the Novell Audit database channel, you can run reports on the data. There are several ways to generate reports against data logged to a Novell Audit database:

- Use the Novell Audit Report application to run your own reports or to run the predefined reports described in “Predefined log reports” on page 119 below.
- Write queries against the logged data using iManager to select *Auditing and Logging>Queries*.
- Write your own SQL queries against the logged data.

The default Novell Audit table is called NAUDITLOG.

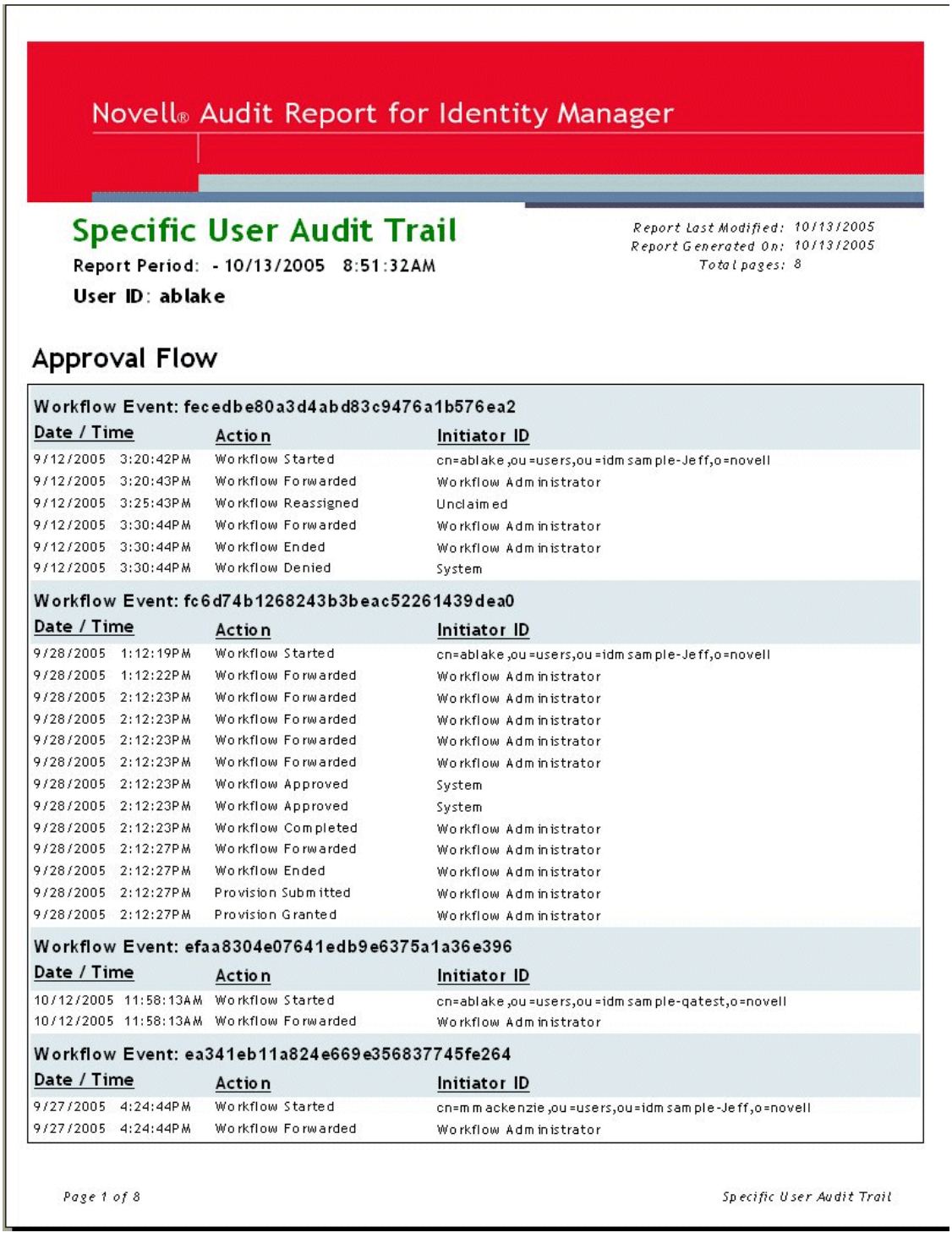
Predefined log reports

The following predefined log reports are created in Crystal Reports (*.rpt*) format for filtering data logged to the Novell Audit database:

Report Name	Description
Administrative Action Report	Shows all administrative actions initiated from the Identity Manager user application portal. This report includes the administrator who initiated the action. It excludes any administrative changes made using iManager or the Designer for IDM

Report Name	Description
Historical Approval Flow Report	Shows all approval flow activities for a specified time frame.
Resource Provisioning report	Shows all provisioning activities, sorted by resource.
Specific User Audit Trail	Shows all activity relating to a user. Activities include both provisioning and self-service activities.
Specific User Provisioning report	Shows all provisioning activities for a specific user.
User Provisioning report	Shows all provisioning activities, sorted by user.

Sample report This is an example of the Specific User Audit Trail report:



Novell® Audit Report for Identity Manager

Specific User Audit Trail

Report Period: - 10/13/2005 8:51:32AM

User ID: ablake

Report Last Modified: 10/13/2005
Report Generated On: 10/13/2005
Total pages: 8

Approval Flow

Workflow Event: fecedbe80a3d4abd83c9476a1b576ea2

Date / Time	Action	Initiator ID
9/12/2005 3:20:42PM	Workflow Started	cn=ablake,ou=users,ou=idmsample-Jeff,o=novell
9/12/2005 3:20:43PM	Workflow Forwarded	Workflow Administrator
9/12/2005 3:25:43PM	Workflow Reassigned	Unclaimed
9/12/2005 3:30:44PM	Workflow Forwarded	Workflow Administrator
9/12/2005 3:30:44PM	Workflow Ended	Workflow Administrator
9/12/2005 3:30:44PM	Workflow Denied	System

Workflow Event: fc6d74b1268243b3beac52261439dea0

Date / Time	Action	Initiator ID
9/28/2005 1:12:19PM	Workflow Started	cn=ablake,ou=users,ou=idmsample-Jeff,o=novell
9/28/2005 1:12:22PM	Workflow Forwarded	Workflow Administrator
9/28/2005 2:12:23PM	Workflow Forwarded	Workflow Administrator
9/28/2005 2:12:23PM	Workflow Forwarded	Workflow Administrator
9/28/2005 2:12:23PM	Workflow Forwarded	Workflow Administrator
9/28/2005 2:12:23PM	Workflow Forwarded	Workflow Administrator
9/28/2005 2:12:23PM	Workflow Approved	System
9/28/2005 2:12:23PM	Workflow Approved	System
9/28/2005 2:12:23PM	Workflow Completed	Workflow Administrator
9/28/2005 2:12:27PM	Workflow Forwarded	Workflow Administrator
9/28/2005 2:12:27PM	Workflow Ended	Workflow Administrator
9/28/2005 2:12:27PM	Provision Submitted	Workflow Administrator
9/28/2005 2:12:27PM	Provision Granted	Workflow Administrator

Workflow Event: efaa8304e07641edb9e6375a1a36e396

Date / Time	Action	Initiator ID
10/12/2005 11:58:13AM	Workflow Started	cn=ablake,ou=users,ou=idmsample-qatest,o=novell
10/12/2005 11:58:13AM	Workflow Forwarded	Workflow Administrator

Workflow Event: ea341eb11a824e669e356837745fe264

Date / Time	Action	Initiator ID
9/27/2005 4:24:44PM	Workflow Started	cn=mmackenzie,ou=users,ou=idmsample-Jeff,o=novell
9/27/2005 4:24:44PM	Workflow Forwarded	Workflow Administrator

Page 1 of 8

Specific User Audit Trail

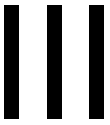
Report file location The report files are located:

Platform	Location
Windows	/nt/dirxml/reports

You can use these reports as templates for creating custom reports in the Crystal Reports Designer or you can run the reports using *Audit Report* (lreport.exe), a Windows program supplied with Novell Audit. The predefined reports query data from the default Novell Audit log database named *naudit* and a database table named *nauditlog*. If your Novell Audit log database has a different name, use the *Set Datasource Location* menu item in Crystal Reports Designer to replace the *naudit* database name with the one in your environment.

For more information, see the section on working with reports in the Novell Audit documentation (http://www.novell.com/documentation/nsureaudit).

Administering the User Application



These chapters tell you how to configure and manage the Identity Manager user application by using the Administration tab of the user interface.

Using the Administration Tab

6

This chapter introduces you to the Administration tab of the Identity Manager user interface. You'll learn how to use the Administration tab to configure and manage the Identity Manager user application. Topics include:

6.1 About the Administration tab

The Identity Manager *user interface* is primarily accessed by end users, who work with the tabs it provides for identity self-service and workflow-based provisioning (with the Provisioning Module for Identity Manager). But this browser-based user interface also provides an *Administration tab*, which administrators can access to configure various characteristics of the underlying Identity Manager *user application*.

For example, the Administration tab can be used to:

- *Change the theme* used for the look and feel of the user interface
- *Customize the identity self-service features* available to end users
- *Specify who is allowed* to perform administration actions
- *Manage other details* about the user application and how it runs

6.2 Who can use the Administration tab

The Administration tab is not visible to typical end users of the Identity Manager user interface. There are two kinds of users who can see and access this tab:

- *User Application Administrators*

  A User Application Administrator is authorized to perform all management functions related to the Identity Manager user application. This includes accessing the Administration tab of the Identity Manager user interface to perform any administration actions that it supports.

  During installation, a user is specified as User Application Administrator. After installation, that user can use the *Security* page on the Administration tab to specify other User Application Administrators, as needed.

  For details, see Chapter 11, "Security Configuration," on page 197.

- *Users permitted by User Application Administrators*

  If necessary, a User Application Administrator can assign permission for one or more end users to see and access specific pages on the Administration tab. These permissions are assigned by using the *Page Admin* page on the Administration tab.

  For details, see Chapter 7, "Page Administration," on page 131.

6.3 Accessing the Administration tab

Once you are a User Application Administrator (or other permitted user), you can access the Administration tab of the Identity Manager user interface when you need to manage the Identity Manager user application. You just need a supported Web browser.

For a list of supported Web browsers, see the *Novell Identity Manager: Installation Guide*.

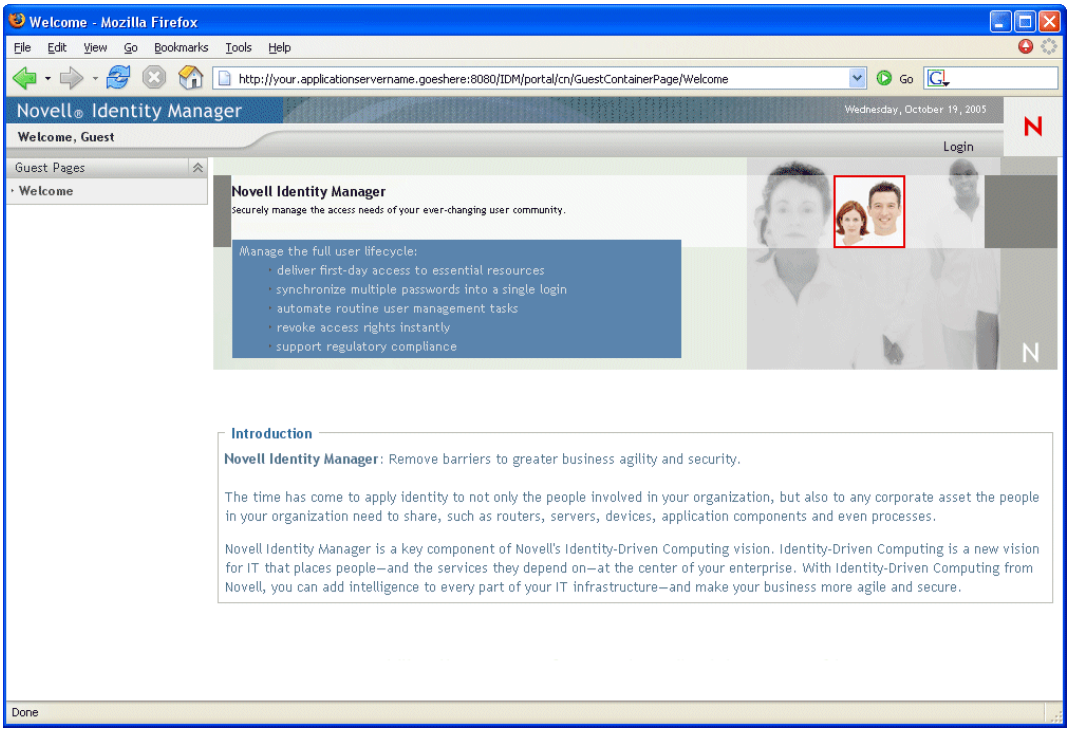
NOTE: To use the Identity Manager user interface, make sure your Web browser has *JavaScript enabled*.

To access the Administration tab:

1. In your *Web browser*, go to the URL for the Identity Manager user interface (as configured at your site). For example:

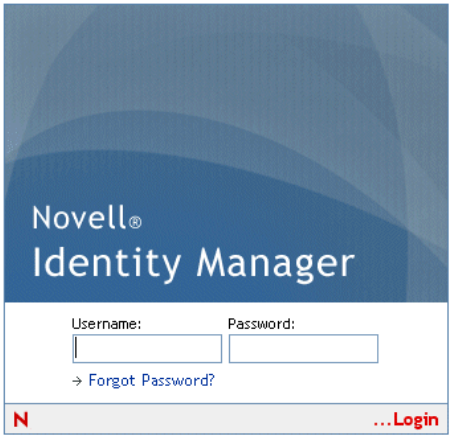
   ```
   http://myappserver:8080/IDM
   ```

   The *guest welcome page* of the user interface displays:



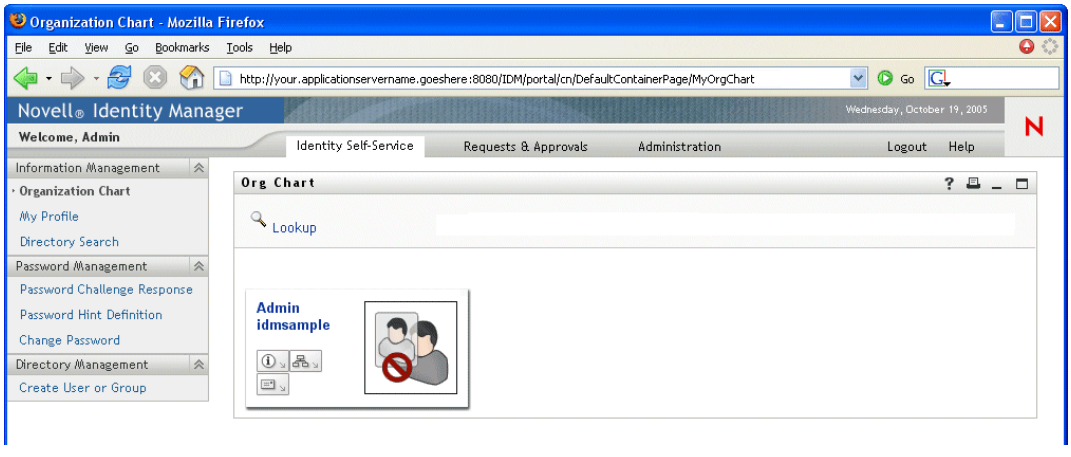
2. Click the *Login* link in the page header.

   The user interface prompts you for a user name and password:



3 Enter the user name and password of a *User Application Administrator* (or a user with some Administration tab permissions), then click *Login*.

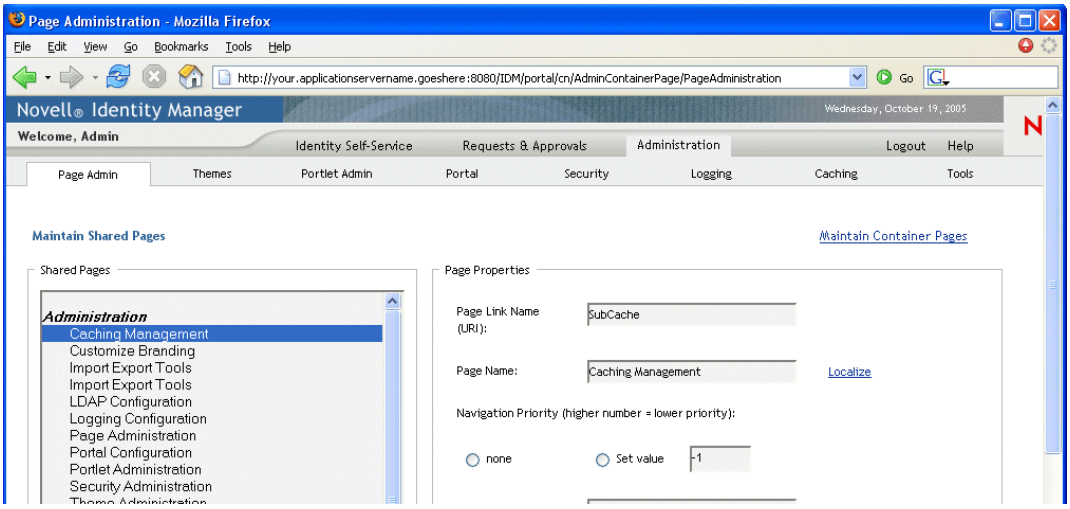
Once you're logged in, you see the appropriate user-interface content for that user. For example:



By default, you are on the *Identity Self-Service* tab.

4 Click the *Administration* tab.

The Administration tab displays a menu of the *administration actions* you can perform. Each choice shows a corresponding page of settings and controls. By default, you see the *Page Admin* page:



For more general information about accessing and working in the Identity Manager user interface, see the *Identity Manager User Application: User Guide*.

6.4 Administration actions you can perform

Once you're on the Administration tab, you can use any available actions to configure and manage the Identity Manager user application. Here's a summary:

Action	Description
Page Admin	Controls the pages displayed in the Identity Manager user interface and who has permission to access them For details, see Chapter 7, "Page Administration," on page 131.
Themes	Controls the look and feel of the Identity Manager user interface For details, see Chapter 8, "Theme Configuration," on page 165.
Portlet Admin	Controls the portlets available in the Identity Manager user interface and who has permission to access them For details, see Chapter 9, "Portlet Administration," on page 171.
Portal	Controls the portal characteristics of the Identity Manager user application and specifies how the user application connects to the identity vault (LDAP provider) For details, see Chapter 10, "Portal Configuration," on page 189.
Security	Specifies who is a User Application Administrator for the Identity Manager user application For details, see Chapter 11, "Security Configuration," on page 197.
Logging	Controls the levels of logging messages you want the Identity Manager user application to generate and specifies whether those messages are sent to Novell Audit For details, see Chapter 12, "Logging Configuration," on page 201.

Action	Description
Caching	Manages various caches maintained by the Identity Manager user application For details, see Chapter 13, “Caching Configuration,” on page 207.
Tools	Enables you to export or import portal content (pages and portlets) used in the Identity Manager user application For details, see Chapter 14, “Tools for Exporting and Importing Portal Data,” on page 217.

Page Administration 7

This chapter tells you how to use the *Page Admin* page on the *Administration tab* of the Identity Manager user interface. Topics include:

For more general information about accessing and working with the Administration tab, see Chapter 6, “Using the Administration Tab,” on page 125.

7.1 About page administration

You can use the Page Admin page to control the *pages* displayed in the Identity Manager user interface and who has *permission* to access them. The user interface includes *two types of pages*:

Type of page	Description
Container	Container pages wrap shared pages with a consistent look and feel, corporate branding, and navigation approach.
Shared	Shared pages provide a coherent set of content that is used for a specific purpose (such as updating a user’s profile). They are called shared pages because they offer services used by multiple people.

Both page types include content in the form of *portlets* (a Java standard for pluggable user-interface elements).

To learn more about portlets, see Chapter 9, “Portlet Administration,” on page 171 and Part IV, “Portlet Reference,” on page 225.

7.1.1 About container pages

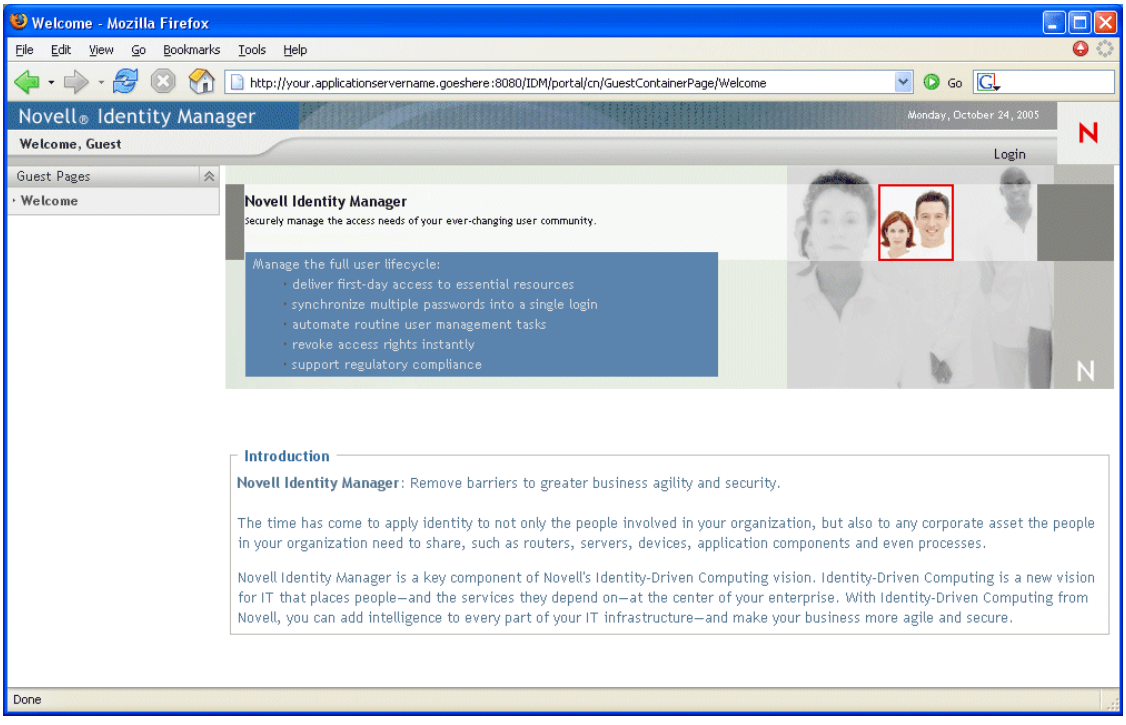
This section introduces you to some container pages that play an important role in the Identity Manager user interface:

Keep in mind that you can modify these container pages if necessary. You also have the option of adding your own container pages.

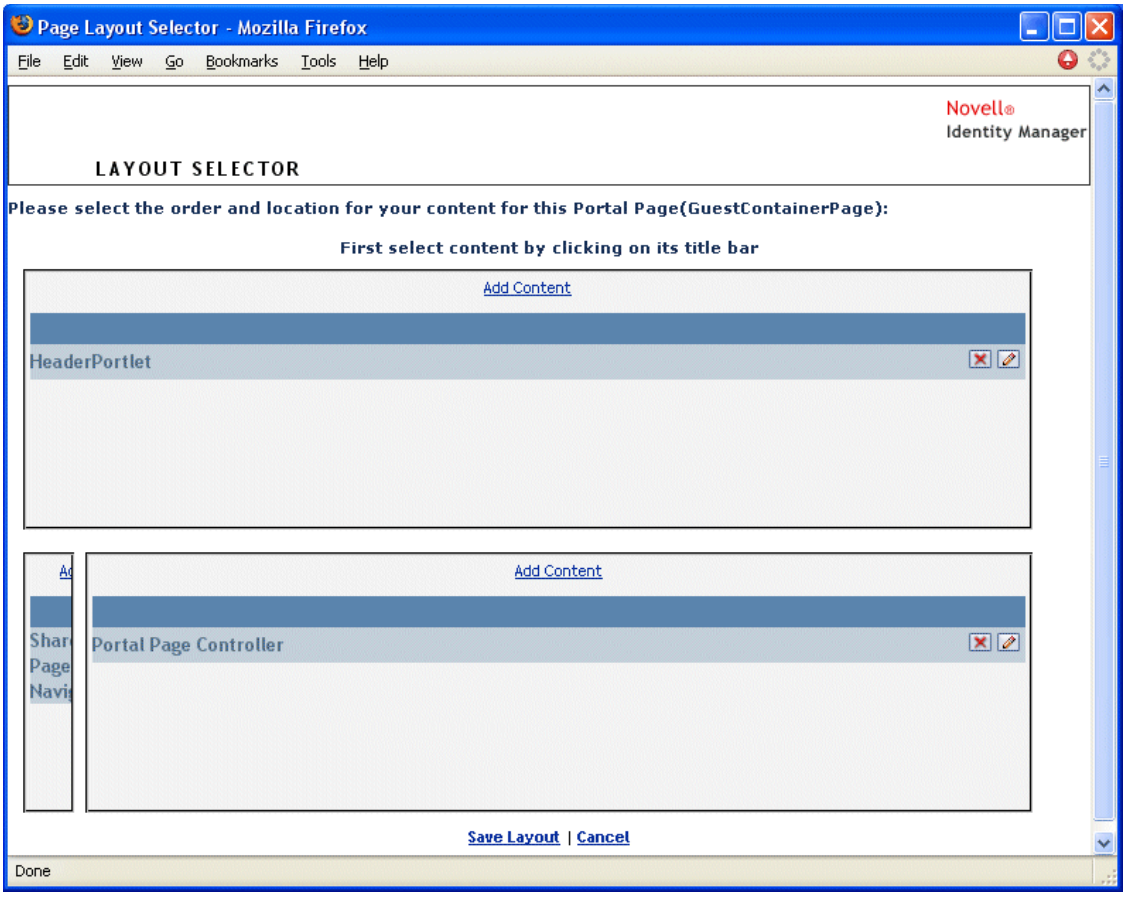
To learn about working with container pages, see Section 7.2, “Creating and maintaining container pages,” on page 138.

GuestContainerPage

By default, when users arrive at the Identity Manager user interface *prior to logging in*, they see the container page named *GuestContainerPage*. That container page displays like this:



Internally, GuestContainerPage has the following *layout*:



The GuestContainerPage layout is divided into *three regions*, which display the following portlets:

Portlet	Description
HeaderPortlet	Displays the header information and top-level tab controls for the user interface
Shared Page Navigation	Displays a vertical menu from which the user can select a shared page to display
Portal Page Controller	Displays the shared page that the user has currently selected via the Shared Page Navigation portlet

Note that, by default, users see only the following in those portlets prior to logging in:

- A single link in the header: *Login*
- A single shared page: *Welcome*

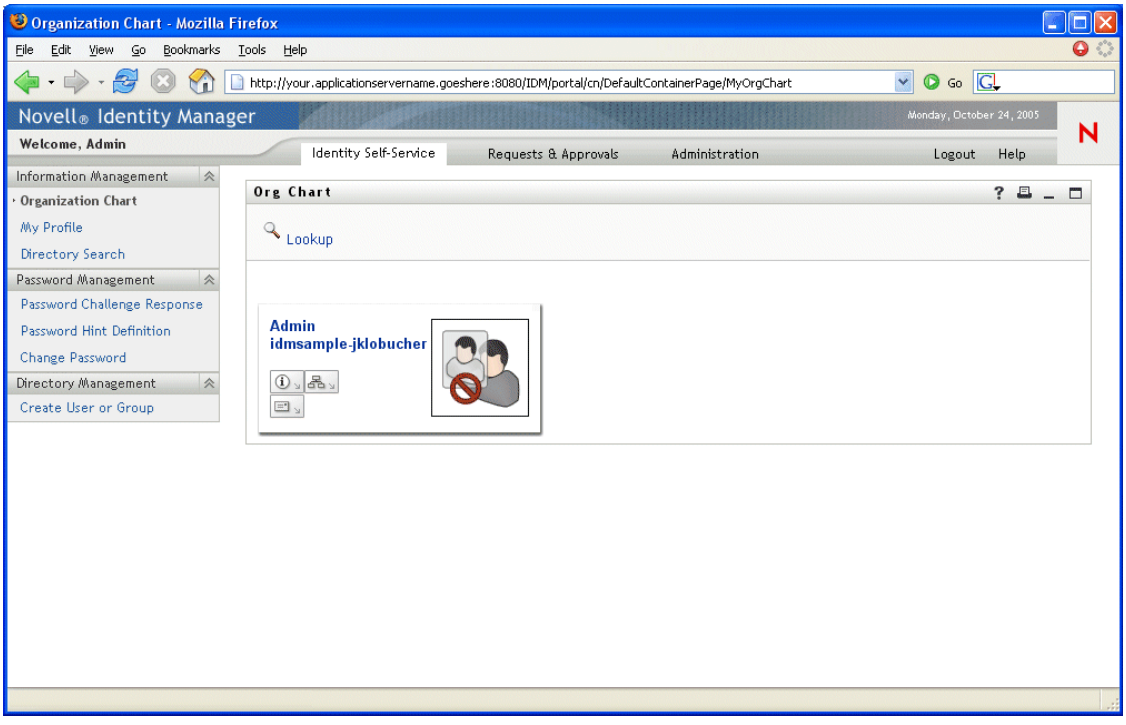
Because the user has not logged in yet, the Shared Page Navigation portlet shows only shared pages that are in the *Guest Pages* category; it filters out all other categories. By default, Welcome is the only page in the Guest Pages category.

After login, the Shared Page Navigation portlet filters out the Guest Pages category. Instead, it shows other categories of shared pages (as specified in its preferences).

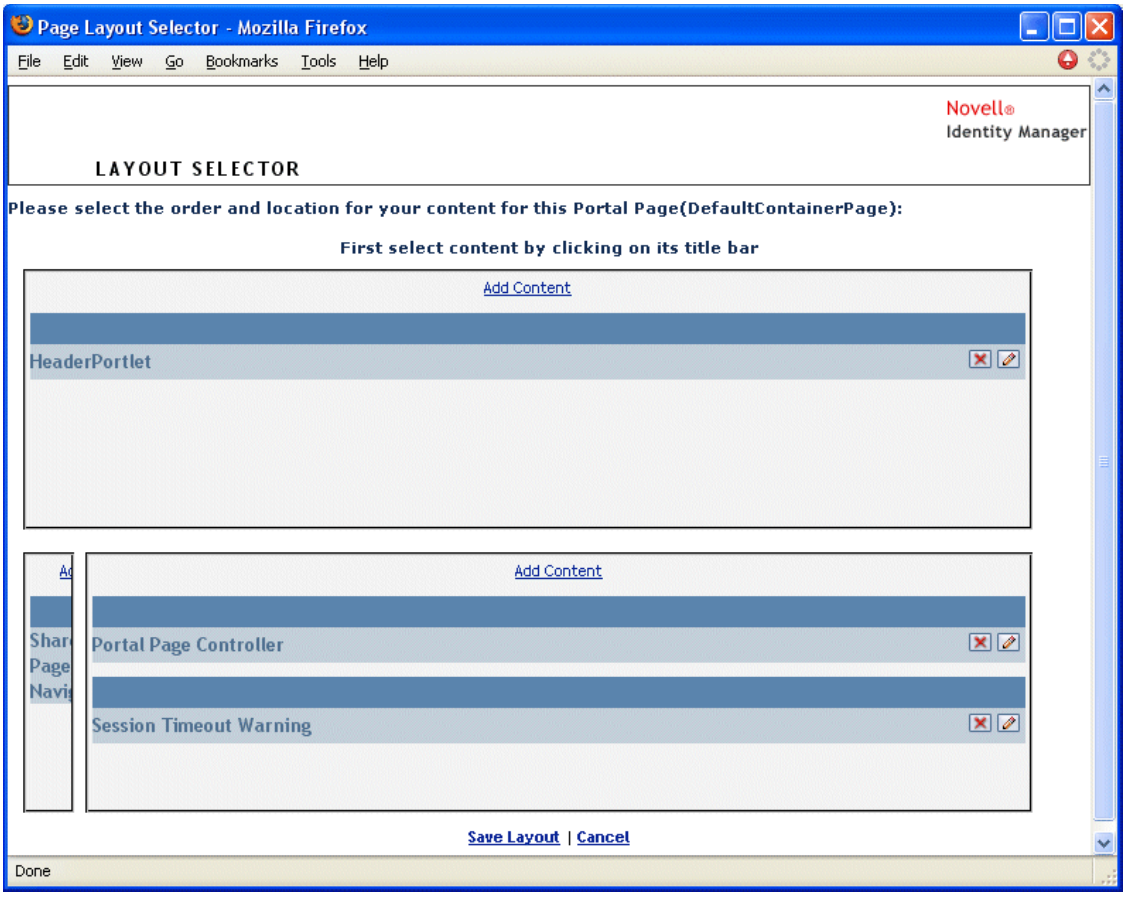
For more information on the Shared Page Navigation portlet, see Chapter 15, “About Portlets,” on page 227.

DefaultContainerPage

By default, *after users log in* to the Identity Manager user interface, they go to the container page named *DefaultContainerPage*. That container page displays like this:



Internally, DefaultContainerPage has the following *layout*:



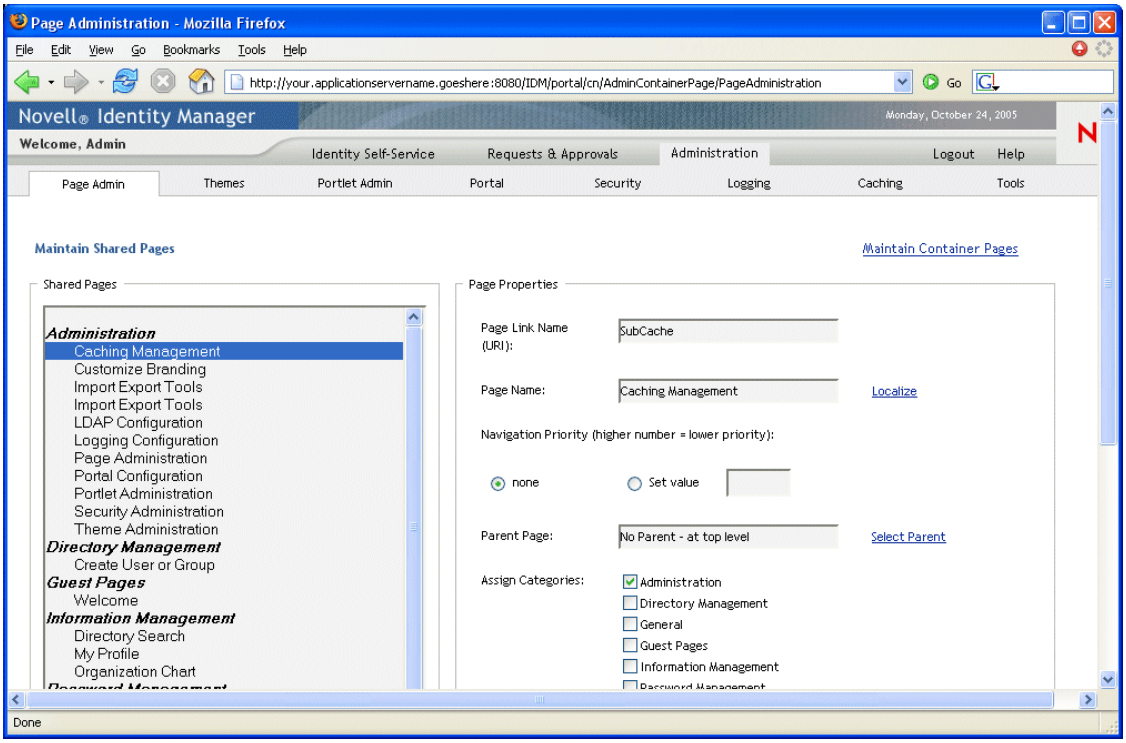
The DefaultContainerPage layout is divided into *three regions*, which display the following portlets:

Portlet	Description
HeaderPortlet	Displays the header information and top-level tab controls for the user interface
Shared Page Navigation	Displays a vertical menu from which the user can select a shared page to display
Portal Page Controller	Displays the shared page that the user has currently selected via the Shared Page Navigation portlet
Session Timeout Warning	Displays an alert message whenever a user's session is about to time out

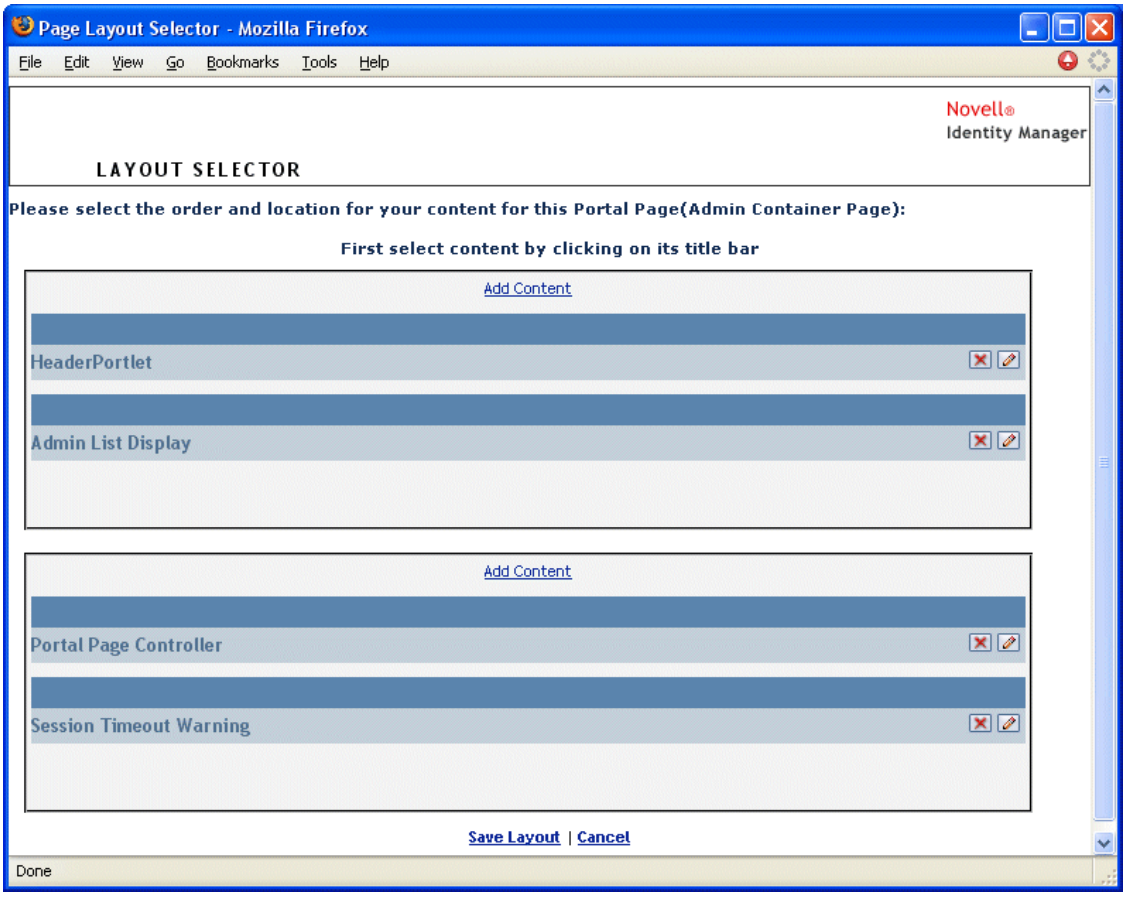
Note that, after user login, *DefaultContainerPage automatically opens the Identity Self-Service tab* in HeaderPortlet.

Admin Container Page

By default, when User Application Administrators (and other authorized users) *click the Administration tab* of the Identity Manager user interface, they go to the container page named *Admin Container Page*. That container page displays like this:



Internally, Admin Container Page has the following *layout*:



The Admin Container Page layout is divided into *two regions*, which display the following portlets:

Portlet	Description
HeaderPortlet	Displays the header information and top-level tab controls for the user interface
Admin List Display	Displays a second level of tabs from which the user can select an administration action to perform
Portal Page Controller	Displays a shared page that corresponds to the tab currently selected by the user via the Admin List Display portlet
Session Timeout Warning	Displays an alert message whenever a user's session is about to time out

7.1.2 About shared pages

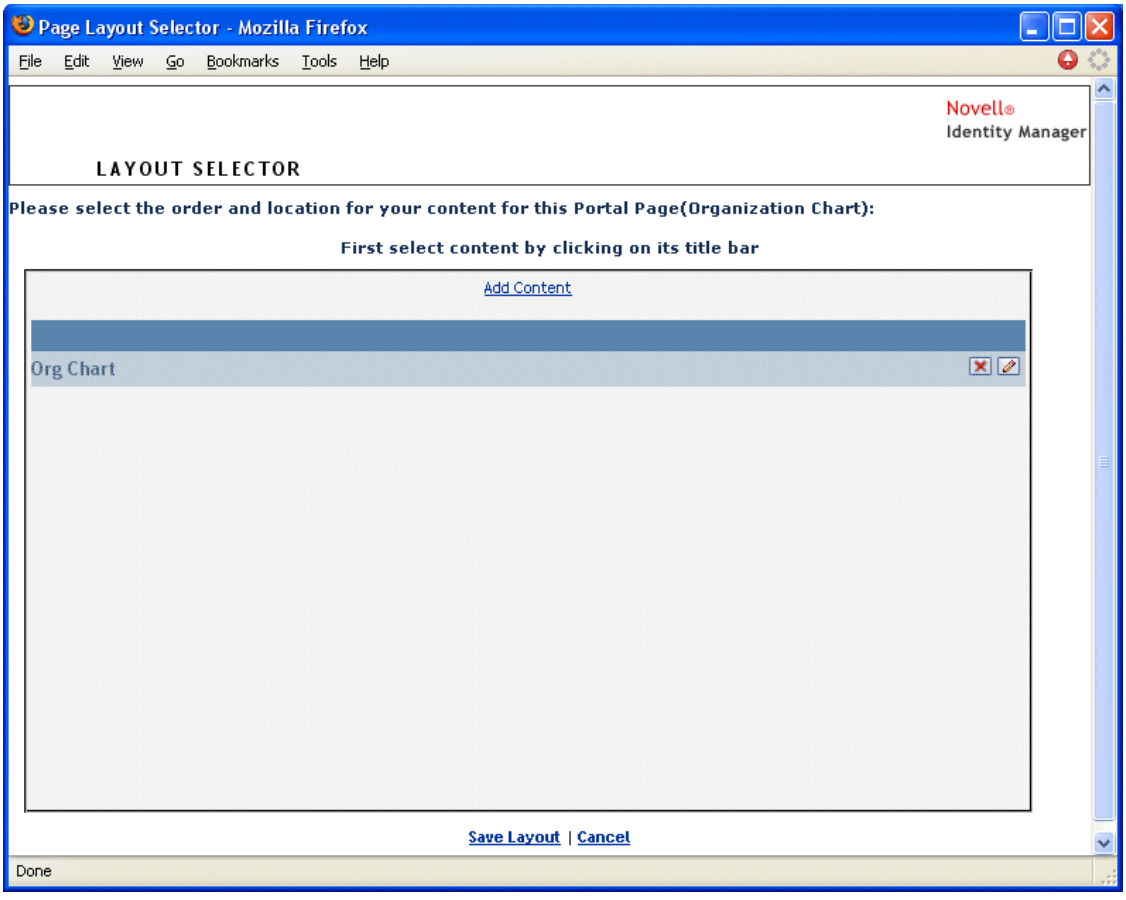
The Identity Manager user interface includes many shared pages, which provide the major content within its container pages. You can modify these shared pages if necessary. You also have the option of adding your own shared pages.

To learn about working with shared pages, see Section 7.3, "Creating and maintaining shared pages," on page 147.

A typical shared page

Let's take a look at one of these shared pages. *Organization Chart* is the *default shared page* that the DefaultContainerPage displays after users log in to the Identity Manager user interface:

Internally, Organization Chart has the following *layout*:



The Organization Chart layout consists of just *one region*, which displays just one portlet (the *Org Chart* portlet).

7.1.3 An exception to page usage

In this chapter, you've seen how these top-level tabs of the Identity Manager user interface are based on pages:

- The *Identity Self-Service tab* uses the *DefaultContainerPage*
- The *Administration tab* uses the *Admin Container Page*

But note that the *Requests & Approvals tab* is based on a different architecture and *cannot be manipulated* via Page Admin.

7.2 Creating and maintaining container pages

The process of creating and maintaining container pages involves the following steps:

1. *Create* a new container page or *select* an existing container page, as described in Section 7.2.1, "Creating container pages," on page 139.
2. *Add content* (in the form of portlets) to the page, as described in Section 7.2.2, "Adding content to a container page," on page 142.

You may also want to *delete content* from the page, as described in Section 7.2.3, “Deleting content from a container page,” on page 143.

3. *Choose a portal layout*, as described in Section 7.2.4, “Modifying the layout of a container page,” on page 144.
4. *Arrange the order and position* of content on the selected layout, as described in Section 7.2.5, “Arranging content on the container page,” on page 145.
5. *Display the new page* right away by entering the container page URL in your browser, as described in Section 7.2.6, “Displaying a container page,” on page 147.

Container pages and layouts Container pages are not tightly bound to portal layouts. That means you can switch layouts for container pages without losing any page contents. When you apply a new layout to a container page, any portlets that have been added to the page are automatically displayed using the new layout. You may need to fine-tune the content placement in the new layout.

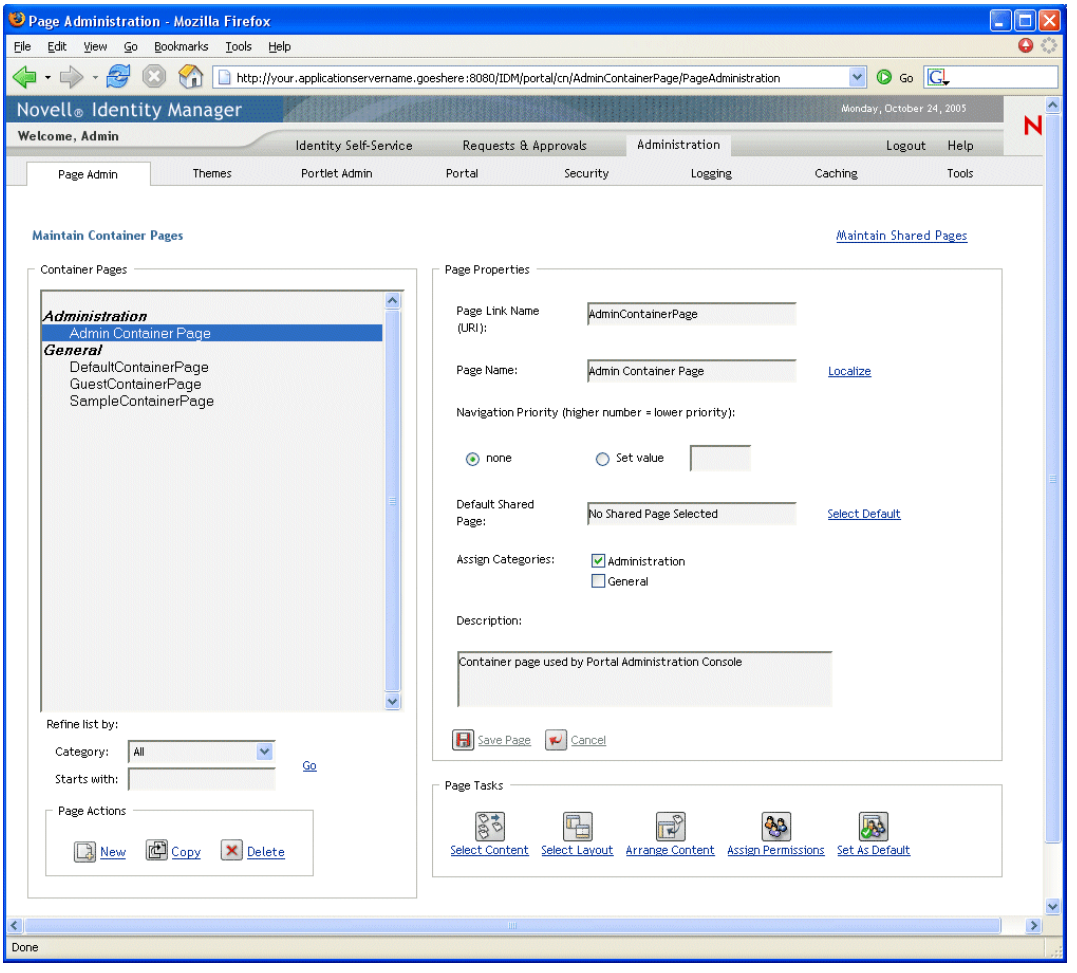
7.2.1 Creating container pages

You can create container pages from scratch or by copying existing pages. This section describes both procedures.

To create a container page from scratch:

1. On the Page Admin page, select *Maintain Container Pages*.

   The Maintain Container Pages panel displays:



2 Select the *New* page action (in the bottom-left section of the panel).

An untitled, uncategorized container page is created.

3 Specify the *page properties* of the container page:

Property	What to do
Page Link Name (URI)	Specify the URI name for the page (as it is to appear within the user interface URL). For example, if you specify the URI: `MyContainerPage` it appears within the URL like this: `http://myappserver:8080/IDM/portal/cn/`**`MyContainerPage`**

Property	What to do
Page Name	Specify the display name for the page. For example: `My Container Page` You can click **Localize** to specify localized versions of this name for other languages.
Navigation Priority	Specify one of the following: • **None** — if you don't need to assign a priority to this container page. • **Set value** — to assign a priority to this container page, relative to other container pages. The priority must be an integer between -1 and 9999, where -1 is the highest priority and 9999 is the lowest. Setting priority values is useful if you want to ensure a particular order when pages are listed by priority, or if you want to ensure a particular selection when multiple default pages exist (in the case of a user who belongs to multiple groups).
Default Shared Page	See Section 7.6, "Selecting a default shared page for a container page," on page 162.
Assign Categories	Select zero or more of the following categories in which you want the page to belong: • Administration • General Assigning categories is useful if you want to ensure proper organization when pages are listed by category, or if you want to ensure an appropriate subset when pages are filtered by category.
Description	Type text that describes the page.

4 Click *Save Page* (at the bottom of the page properties section).

To create a container page by copying an existing page:

1 On the Page Admin page, select *Maintain Container Pages*.

The Maintain Container Pages panel displays (as shown in the previous procedure).

2 In the list of container pages, *select* the page you want to copy.

TIP: If the list is long, you can *refine* it (by category or starting text) to more easily find the desired page.

3 Select the *Copy* page action (in the bottom-left section of the panel).

A new container page is created with the name *Copy of OriginalPageName*.

4 Specify the *page properties* of the container page (as described in the previous procedure).

5 Click *Save Page* (at the bottom of the page properties section).

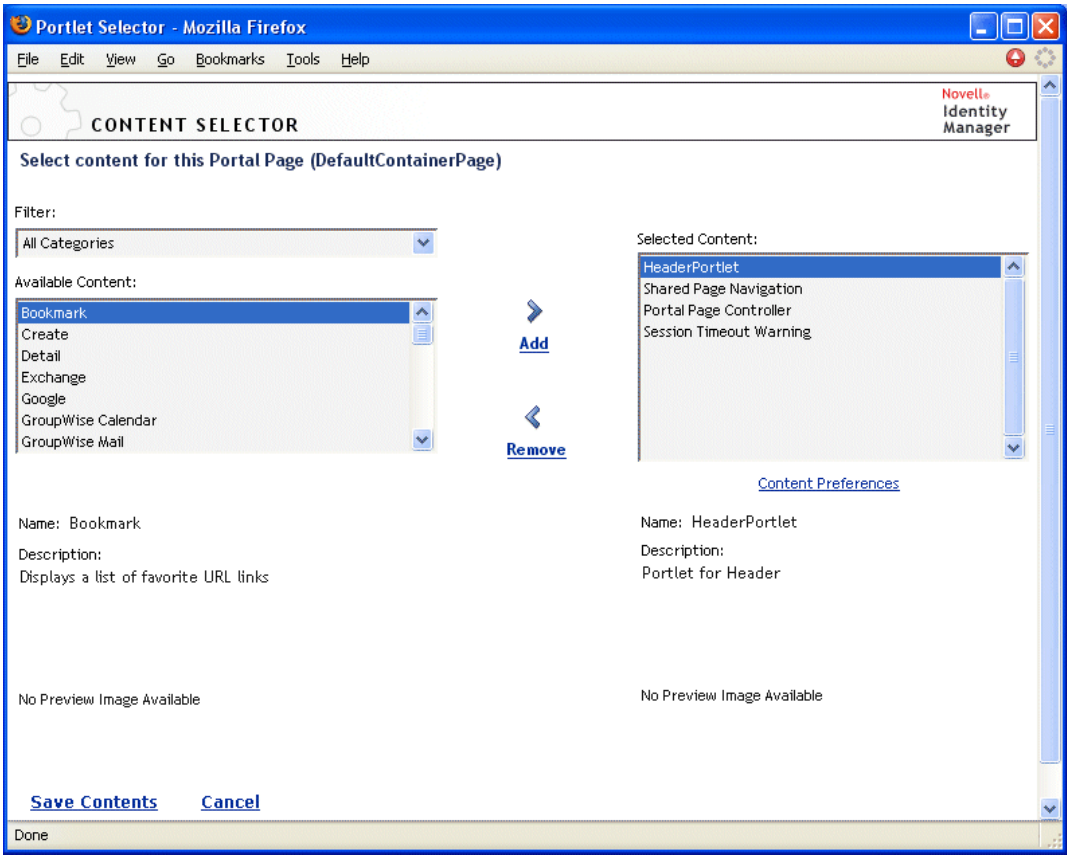
7.2.2 Adding content to a container page

After you create a container page, the next step is to add content by selecting portlets to place on the page. You can use prebuilt portlets supplied with the Identity Manager user application or other portlets you have registered.

To add content to a container page:

1. Open a new or existing page on the Maintain Container Pages panel, then click the *Select Content* page task (at the bottom of the panel).

   The *Content Selector* displays in a new browser window:



2. If you want to display a specific category of available content, select a category from the *Filter* dropdown menu.
3. Select one or more portlets from the list of *Available Content*.

   TIP: Hold down the *Control* key to select multiple non-contiguous portlets from the list; use the *Shift* key to make multiple contiguous selections.

4. Click *Add* to move your choices to the list of *Selected Content*.
5. You can click *Content Preferences* to edit the preferences of any portlet you have selected for your container page. The preference values you specify take effect for the instance of the portlet that appears on your page.
6. Click *Save Contents*.

Now that you have chosen the content for your container page, you can select a new layout as described in Section 7.2.4, “Modifying the layout of a container page,” on page 144, or arrange the content on the current layout as described in Section 7.2.5, “Arranging content on the container page,” on page 145.

7.2.3 Deleting content from a container page

In the process of creating container pages, you may want to delete content by removing portlets from a page. You can use the Content Selector or Layout Selector, as described in the following procedures.

To delete content from a container page using the Content Selector:

1. Open a page on the Maintain Container Pages panel, then click the *Select Content* page task (at the bottom of the panel).

   The *Content Selector* displays in a new browser window (as shown in the previous procedure).

2. Select a portlet you want to delete from the Selected Content list and click *Remove*.

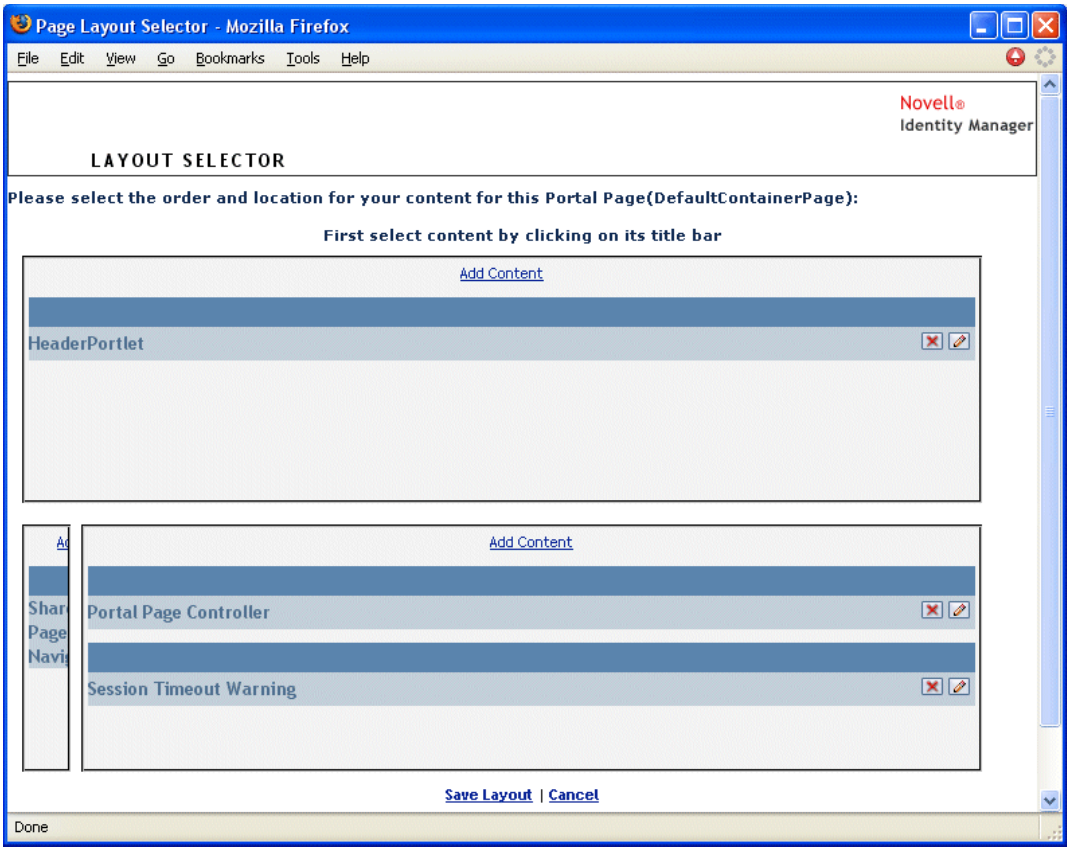
   The portlet is removed from the page.

3. Click *Save Contents*.

To delete content from a container page using the Layout Selector:

1. Open a page on the Maintain Container Pages panel, then click the *Arrange Content* page task (at the bottom of the panel).

   The *Layout Selector* displays in a new browser window, showing the portlets on that page:



2 Click the *X* button for a portlet you want to remove.

3 When you're prompted for confirmation, click *OK*.

The portlet is removed from the page.

4 Click *Save Layout*.

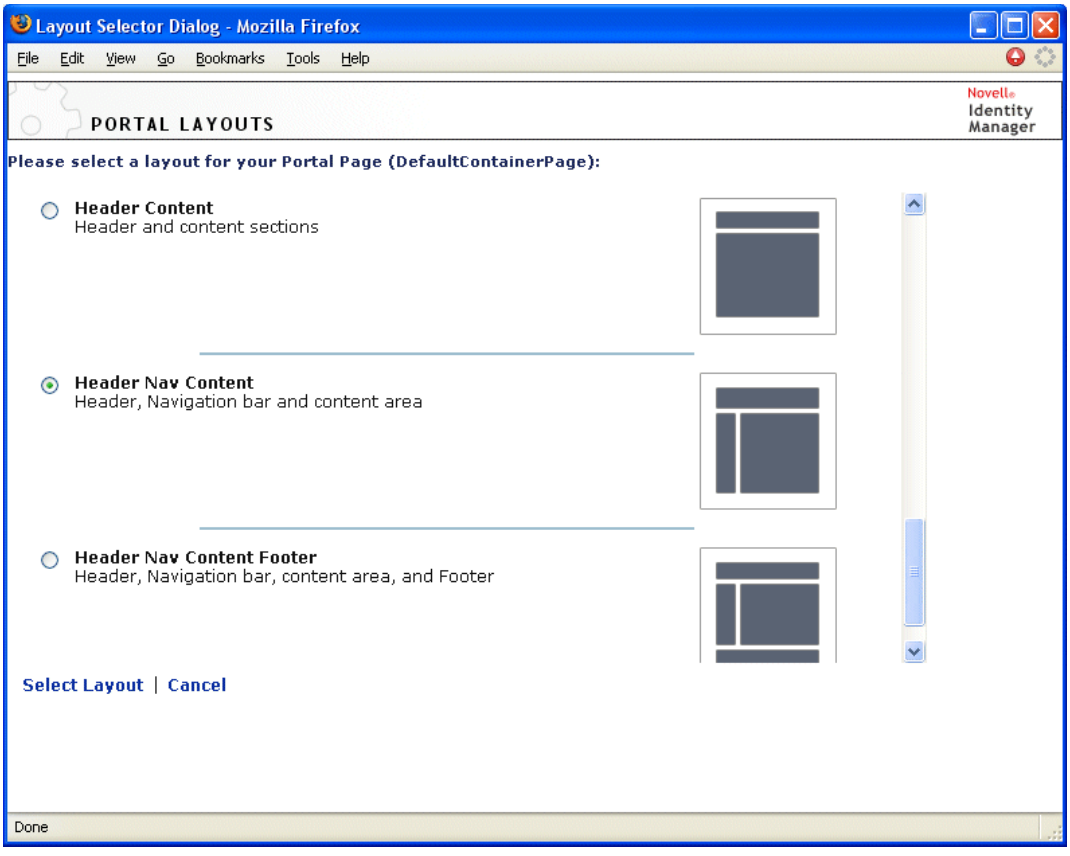
7.2.4 Modifying the layout of a container page

When you modify the layout of a container page, existing content is shifted to accommodate the new layout. In some cases, you may need to fine-tune the end result.

To modify the layout of a container page:

1 Open a page on the Maintain Container Pages panel, then click the *Select Layout* page task (at the bottom of the panel).

The *Portal Layouts* list displays in a new browser window:



2 *Scroll* through the choices and *select* the layout you want.

3 Click *Select Layout*.

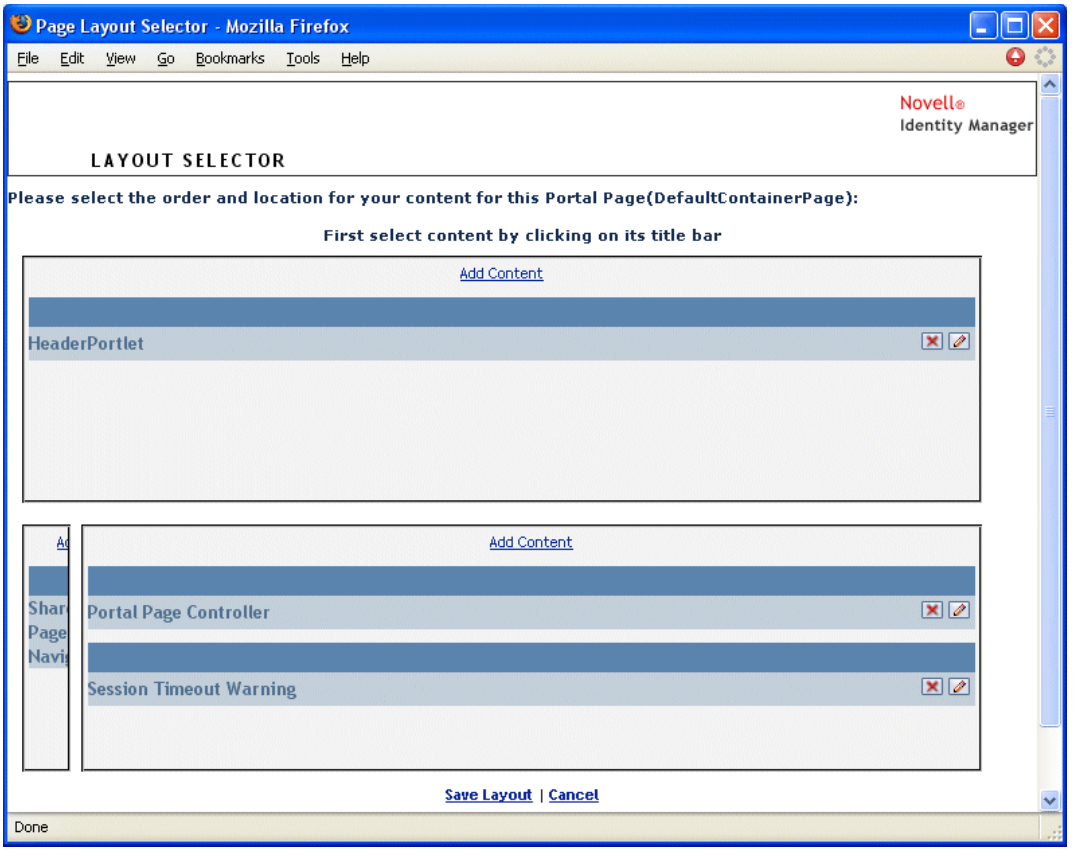
7.2.5 Arranging content on the container page

After you have designated the content and layout for your container page, you can position the content in the selected layout, add other portlets in specific locations, or delete portlets.

To arrange content on a container page:

1 Open a page on the Maintain Container Pages panel, then click the *Arrange Content* page task (at the bottom of the panel).

The *Layout Selector* displays in a new browser window, showing the portlets on that page:



2 If you want to *add a portlet* to the page, follow these steps:

2a Click *Add Content* in the desired layout frame.

The *Portlet Selector* displays in a new browser window.

2b If you want to display a specific category of available content, select a category from the *Filter* dropdown menu.

2c Select a portlet you want from the list of *Available Content*.

2d Click *Select Content*.

The Portlet Selector closes and the portlet you selected appears in the target layout frame of the Layout Selector.

3 If you want to *move a portlet* to a different location in the layout, follow these browser-specific steps:

Browser	What to do
Internet Explorer	1. Move your cursor over the title bar of the portlet until the cursor changes to a hand shape. 2. Hold down the left mouse button and drag the portlet to the desired location in the layout.
Mozilla	1. Click the portlet you want to move. 2. Click inside the destination layout frame. The portlet moves to the destination.

4 If you want to *remove a portlet* from the layout, follow these steps:

4a Click the *X* button for the portlet you want to remove.

4b When you're prompted for confirmation, click *OK*.

The portlet is removed from the layout.

5 If you want to *edit the preferences* of a portlet, follow these steps:

5a Click the *pencil* button for the portlet you want to edit.

The portlet's *Content Preferences* display in your browser.

5b *Change* preference values, as appropriate.

The preference values you specify take effect for the instance of the portlet that appears on your page.

5c Click *Save Preferences*.

6 Click *Save Layout* to record your changes and close the Layout Selector.

7.2.6 Displaying a container page

You can display your page by going to the container page URL in your browser.

To display a container page:

- In your *Web browser*, go to the following URL:

  `http://server:port/IDM-war-context/portal/cn/container-page-name`

  For example, to display the container page named *MyContainerPage*:

  ```
  http://myappserver:8080/IDM/portal/cn/MyContainerPage
  ```

7.3 Creating and maintaining shared pages

The process of creating and maintaining shared pages involves the following steps:

1 *Create* a new shared page or *select* an existing shared page, as described in Section 7.3.1, "Creating shared pages," on page 148.

2 *Add content* (in the form of portlets) to the page, as described in Section 7.3.2, "Adding content to a shared page," on page 150.

You may also want to *delete content* from the page, as described in Section 7.3.3, "Deleting content from a shared page," on page 151.

3 *Choose a portal layout*, as described in Section 7.3.4, "Modifying the layout of a shared page," on page 152.

4 *Arrange the order and position* of content on the selected layout, as described in Section 7.3.5, "Arranging content on the shared page," on page 153.

5 *Display the new page* right away by entering the shared page URL in your browser, as described in Section 7.3.6, "Displaying a shared page," on page 155.

Shared pages and layouts Shared pages are not tightly bound to portal layouts. That means you can switch layouts for shared pages without losing any page contents. When a new layout is applied, any

portlets that have been added to the page are automatically displayed using the new layout. You may need to fine-tune the content placement in the new layout.

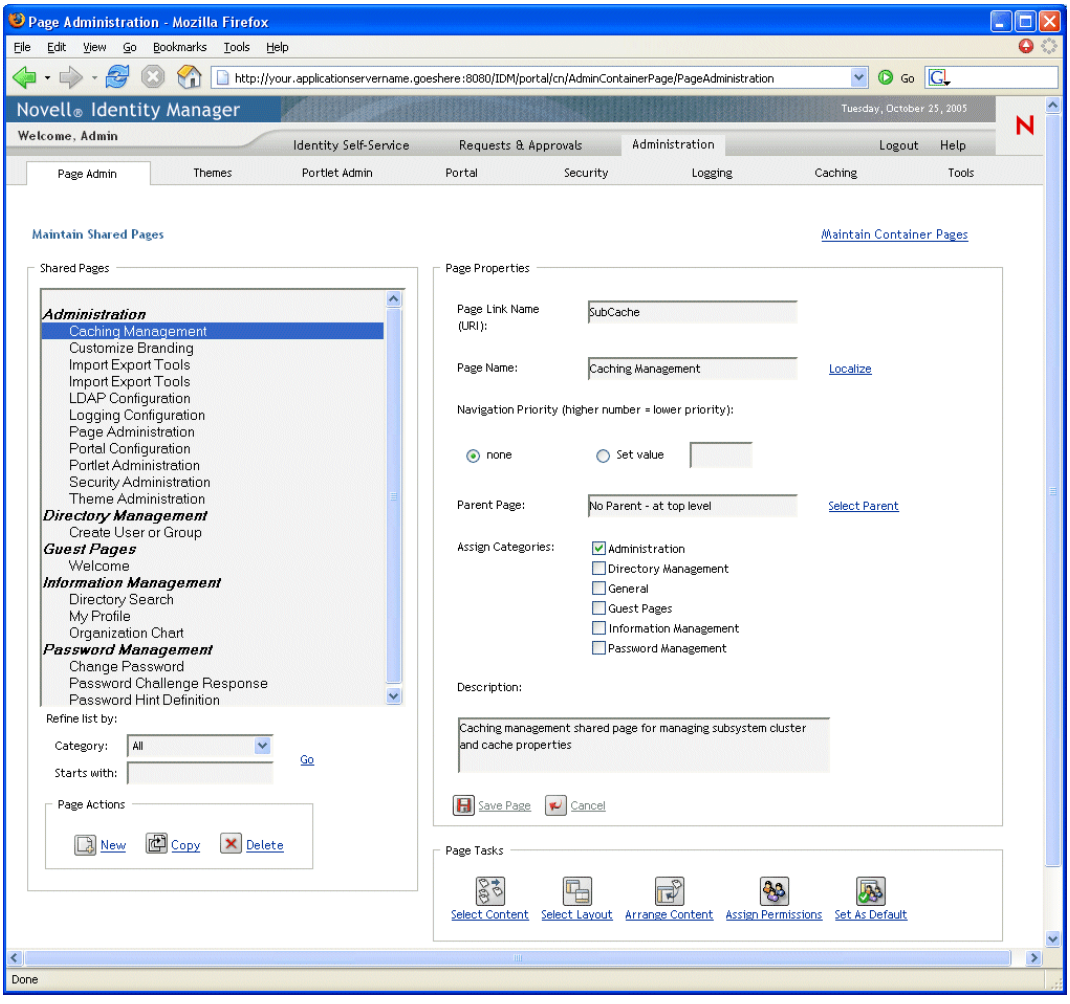
7.3.1 Creating shared pages

You can create shared pages from scratch or by copying existing pages. This section describes both procedures.

To create a shared page from scratch:

1 On the Page Admin page, select *Maintain Shared Pages*.

The Maintain Shared Pages panel displays:



2 Select the *New* page action (in the bottom-left section of the panel).

An untitled, uncategorized shared page is created.

3 Specify the *page properties* of the shared page:

Property	What to do
Page Link Name (URI)	Specify the URI name for the page (as it is to appear within the user interface URL). For example, if you specify the URI: `MySharedPage` it appears within the URL like this: `http://myappserver:8080/IDM/portal/cn/MyContainerPage/`**`MySharedPage`**
Page Name	Specify the display name for the page. For example: `My Shared Page` You can click **Localize** to specify localized versions of this name for other languages.
Navigation Priority	Specify one of the following: • **None** — if you don't need to assign a priority to this shared page. • **Set value** — to assign a priority to this shared page, relative to other shared pages. The priority must be an integer between -1 and 9999, where -1 is the highest priority and 9999 is the lowest. Setting priority values is useful if you want to ensure a particular order when pages are listed by priority, or if you want to ensure a particular selection when multiple default pages exist (in the case of a user who belongs to multiple groups).
Parent Page	If you want this shared page to be the child of another shared page, click **Select Parent**. Make sure that both the parent and child pages belong to the **same categories** (to prevent display problems). At runtime, the end user will see this relationship when using the Shared Page Navigation portlet. When displaying the list of shared pages, it shows children indented under their parents. (Note that child pages do not inherit content, preferences, or settings from their parent pages. Conversely, parent pages do not automatically display the content of child pages along with their own content.)

Property	What to do
Assign Categories	Select zero or more of the following categories in which you want the page to belong: • Administration • Directory Management • General • Guest Pages • Information Management • Password Management Assigning categories is useful if you want to ensure proper organization when pages are listed by category, or if you want to ensure an appropriate subset when pages are filtered by category. **NOTE: Guest Pages** is a special category used to identify shared pages that may be displayed prior to user login (and not after user login). For more information, see the section on the Shared Page Navigation portlet in Chapter 15, "About Portlets," on page 227.
Description	Type text that describes the page.

4 Click *Save Page* (at the bottom of the page properties section).

To create a shared page by copying an existing page:

1 On the Page Admin page, select *Maintain Shared Pages*.

The Maintain Shared Pages panel displays (as shown in the previous procedure).

2 In the list of shared pages, *select* the page you want to copy.

TIP: If the list is long, you can *refine* it (by category or starting text) to more easily find the desired page.

3 Select the *Copy* page action (in the bottom-left section of the panel).

A new shared page is created with the name *Copy of OriginalPageName*.

4 Specify the *page properties* of the shared page (as described in the previous procedure).

5 Click *Save Page* (at the bottom of the page properties section).

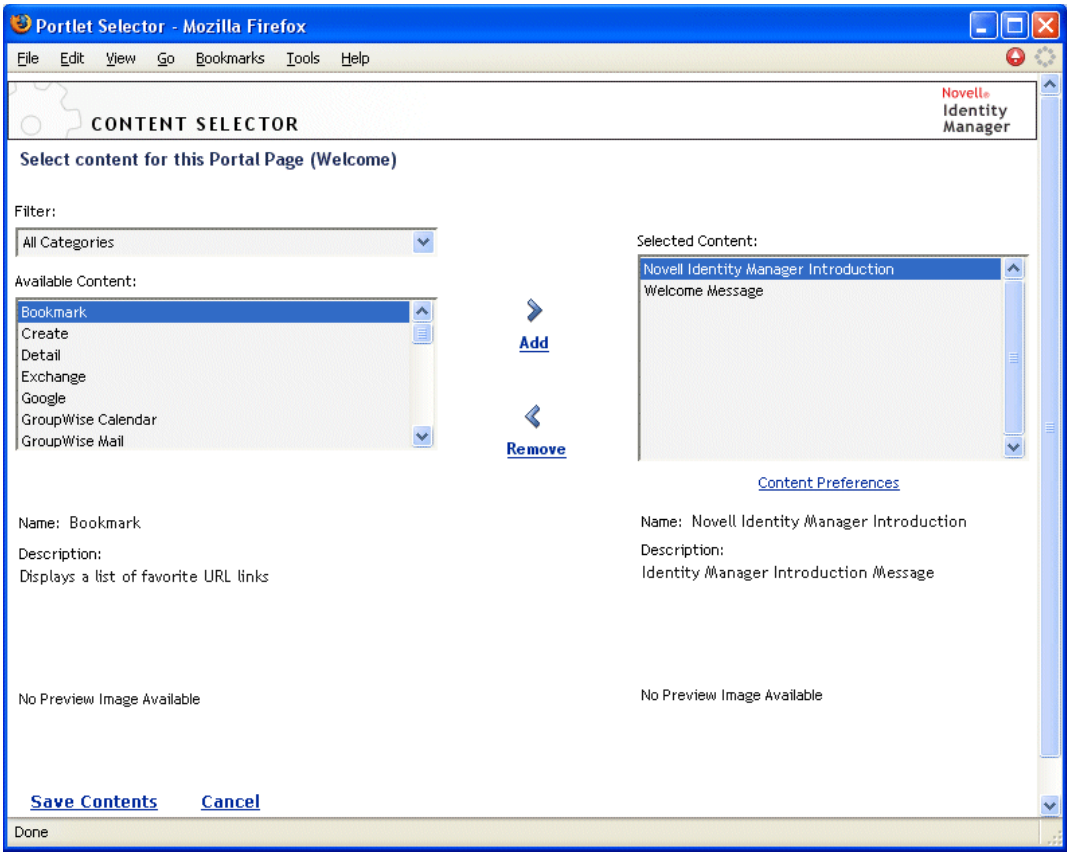
7.3.2 Adding content to a shared page

After you create a shared page, the next step is to add content by selecting portlets to place on the page. You can use prebuilt portlets supplied with the Identity Manager user application or other portlets you have registered.

To add content to a shared page:

1 Open a new or existing page on the Maintain Shared Pages panel, then click the *Select Content* page task (at the bottom of the panel).

The *Content Selector* displays in a new browser window:



2 If you want to display a specific category of available content, select a category from the *Filter* dropdown menu.

3 Select one or more portlets from the list of *Available Content*.

TIP: Hold down the *Control* key to select multiple non-contiguous portlets from the list; use the *Shift* key to make multiple contiguous selections.

---

4 Click *Add* to move your choices to the list of *Selected Content*.

5 You can click *Content Preferences* to edit the preferences of any portlet you have selected for your shared page. The preference values you specify take effect for the instance of the portlet that appears on your page.

6 Click *Save Contents*.

Now that you have chosen the content for your shared page, you can select a new layout as described in Section 7.3.4, “Modifying the layout of a shared page,” on page 152, or arrange the content on the current layout as described in Section 7.3.5, “Arranging content on the shared page,” on page 153.

7.3.3 Deleting content from a shared page

In the process of creating shared pages, you may want to delete content by removing portlets from a page. You can use the Content Selector or Layout Selector, as described in the following procedures.

To delete content from a shared page using the Content Selector:

1. Open a page on the Maintain Shared Pages panel, then click the *Select Content* page task (at the bottom of the panel).

   The *Content Selector* displays in a new browser window (as shown in the previous procedure).

2. Select a portlet you want to delete from the Selected Content list and click *Remove*.

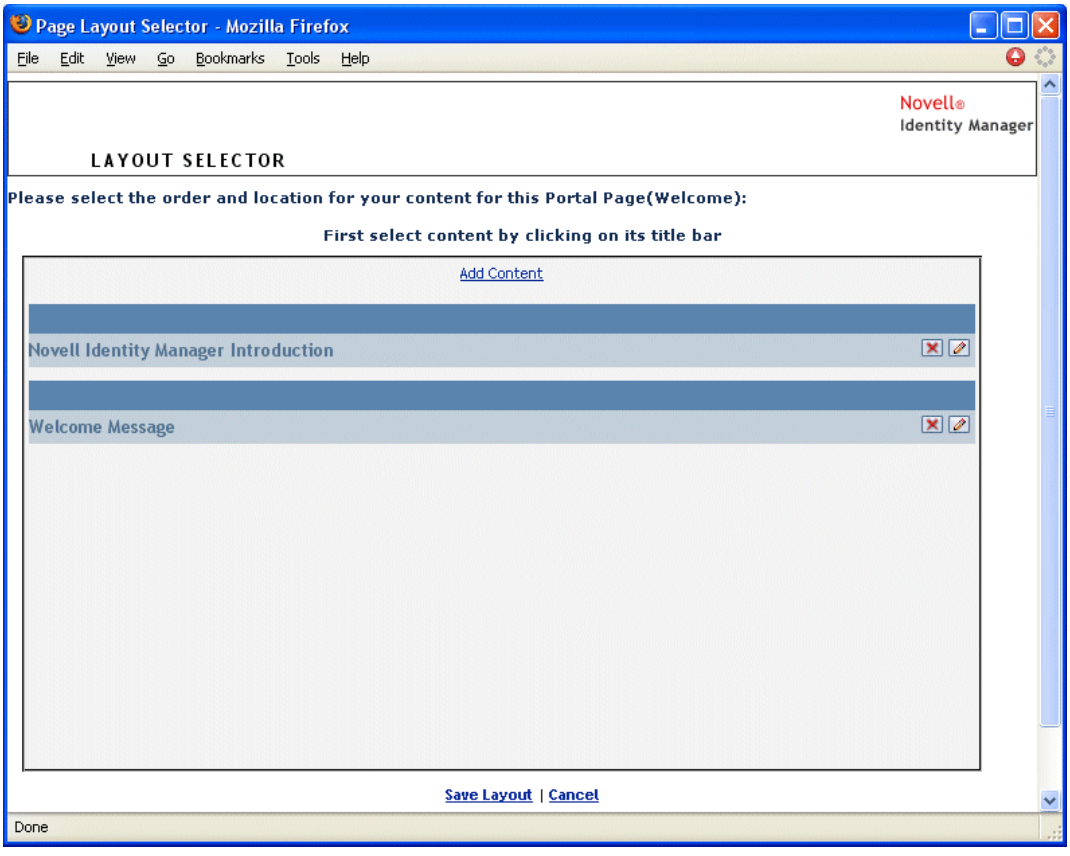
   The portlet is removed from the page.

3. Click *Save Contents*.

To delete content from a shared page using the Layout Selector:

1. Open a page on the Maintain Shared Pages panel, then click the *Arrange Content* page task (at the bottom of the panel).

   The *Layout Selector* displays in a new browser window, showing the portlets on that page:



2. Click the *X* button for a portlet you want to remove.
3. When you're prompted for confirmation, click *OK*.

   The portlet is removed from the page.

4. Click *Save Layout*.

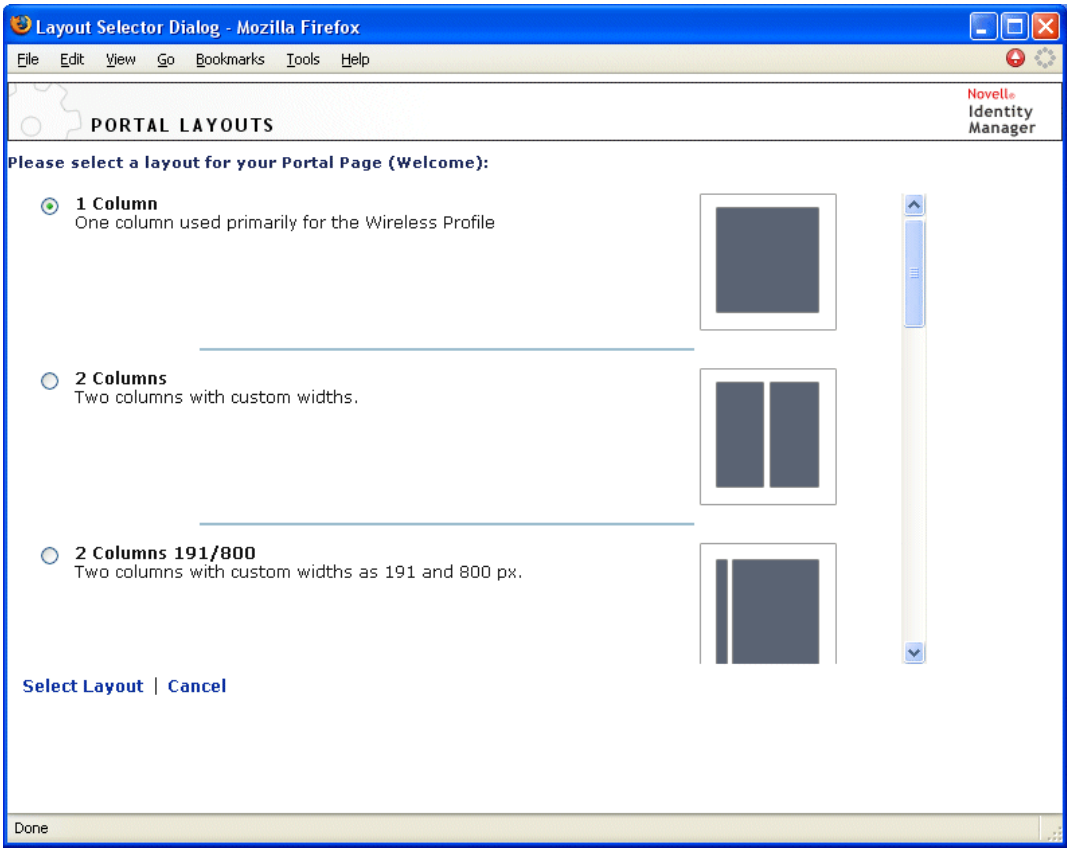
7.3.4 Modifying the layout of a shared page

When you modify the layout of a shared page, existing content is shifted to accommodate the new layout. In some cases, you may need to fine-tune the end result.

To modify the layout of a shared page:

1. Open a page on the Maintain Shared Pages panel, then click the *Select Layout* page task (at the bottom of the panel).

   The *Portal Layouts* list displays in a new browser window:



2. *Scroll* through the choices and *select* the layout you want.
3. Click *Select Layout*.

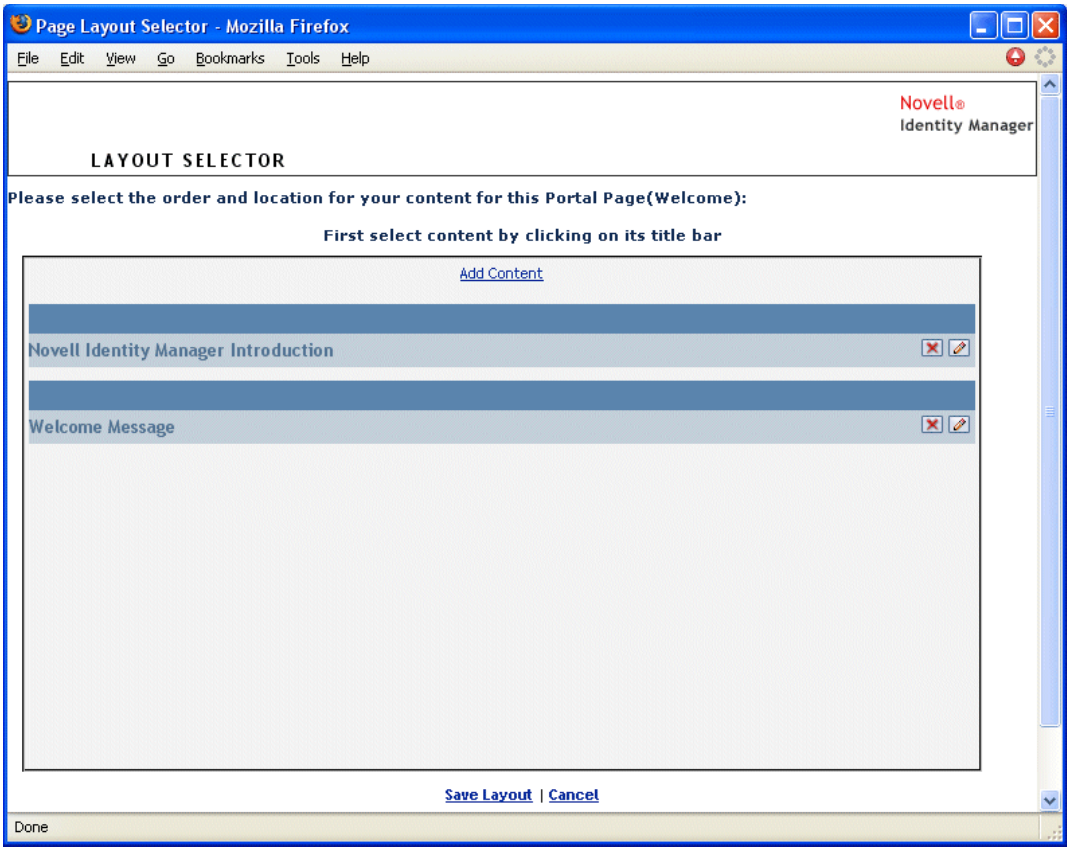
7.3.5 Arranging content on the shared page

After you have designated the content and layout for your shared page, you can position the content in the selected layout, add other portlets in specific locations, or delete portlets.

To arrange content on a shared page:

1. Open a page on the Maintain Shared Pages panel, then click the *Arrange Content* page task (at the bottom of the panel).

   The *Layout Selector* displays in a new browser window, showing the portlets on that page:



2 If you want to *add a portlet* to the page, follow these steps:

2a Click *Add Content* in the desired layout frame.

The *Portlet Selector* displays in a new browser window.

2b If you want to display a specific category of available content, select a category from the *Filter* dropdown menu.

2c Select a portlet you want from the list of *Available Content*.

2d Click *Select Content*.

The Portlet Selector closes and the portlet you selected appears in the target layout frame of the Layout Selector.

3 If you want to *move a portlet* to a different location in the layout, follow these browser-specific steps:

Browser	What to do
Internet Explorer	1. Move your cursor over the title bar of the portlet until the cursor changes to a hand shape. 2. Hold down the left mouse button and drag the portlet to the desired location in the layout.
Mozilla	1. Click the portlet you want to move. 2. Click inside the destination layout frame. The portlet moves to the destination.

4 If you want to *remove a portlet* from the layout, follow these steps:

4a Click the *X* button for the portlet you want to remove.

4b When you're prompted for confirmation, click *OK*.

The portlet is removed from the layout.

5 If you want to *edit the preferences* of a portlet, follow these steps:

5a Click the *pencil* button for the portlet you want to edit.

The portlet's *Content Preferences* display in your browser.

5b *Change* preference values, as appropriate.

The preference values you specify take effect for the instance of the portlet that appears on your page.

5c Click *Save Preferences*.

6 Click *Save Layout* to record your changes and close the Layout Selector.

7.3.6 Displaying a shared page

You can display your page by going to the shared page URL in your browser.

To display a shared page:

- In your *Web browser*, go to the following URL:

  `http://server:port/IDM-war-context/portal/pg/shared-page-name`

  For example, to display the shared page named *MySharedPage*:

  `http://myappserver:8080/IDM/portal/pg/MySharedPage`

7.4 Assigning permissions for pages

You can assign permission to other users, groups, and containers to work with specific container pages and shared pages. Two security levels of permission can be assigned:

Permission	Description	Can be assigned for
View	Allows a user, group, or container to access the page and see it in a list of available pages	**Container pages** and **shared pages**
Ownership	Allows a user, group, or container to modify the content and layout of the page, and to assign View and Ownership permission to other users, groups, and containers	**Shared pages**

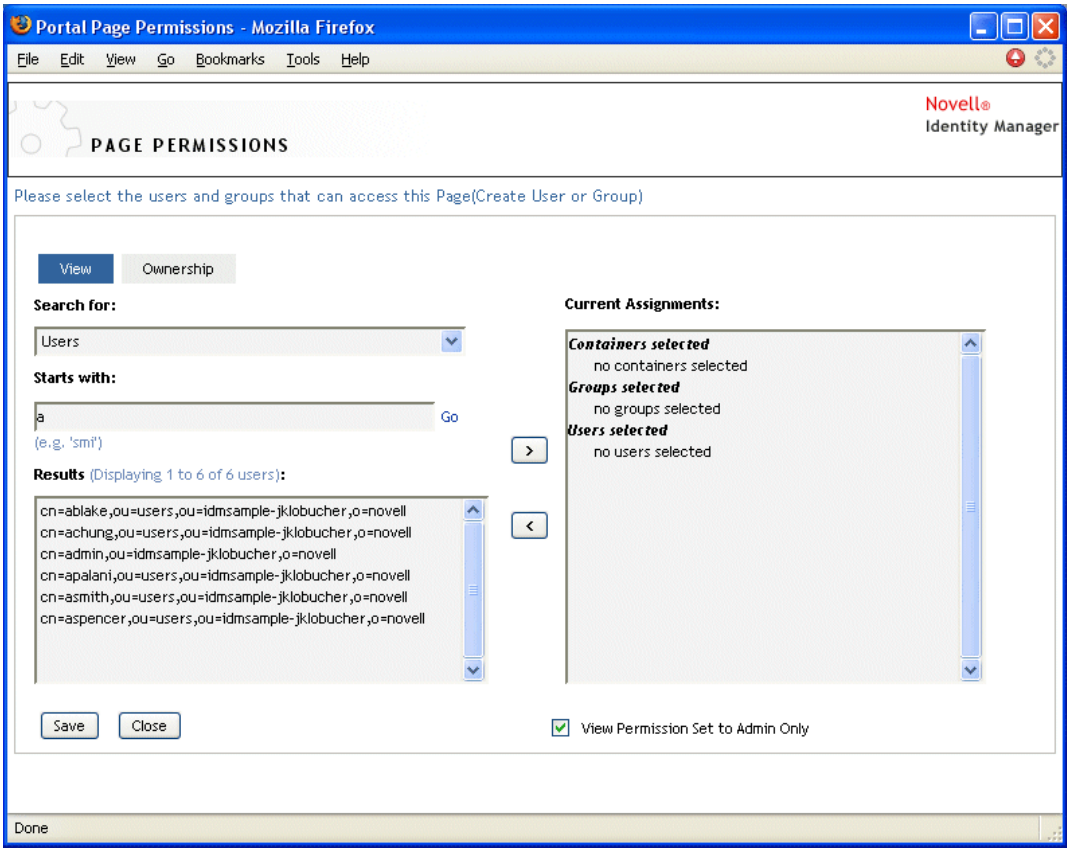
7.4.1 Assigning page View permission

When you assign users View permission for a container page or shared page, they can access the page and see it in a list of available pages.

To assign View permission for container pages or shared pages:

1 Open a page on the Maintain Container Pages panel or the Maintain Shared Pages panel, then click the *Assign Permissions* page task (at the bottom of the panel).

The *Page Permissions* dialog displays in a new browser window:



2 Go to the *View* tab.

3 Specify values for the following *search settings*:

Setting	What to do
Search for	Select one of the following from the dropdown menu: • Users • Groups • Containers

Setting	What to do

Starts with	If you want to:
	• **Find all** available objects of your specified type (user, group, or container), then make this setting blank.
	• **Find a subset** of those objects, then enter the starting character(s) of the CN values you want. (Case is not considered. Wildcards are not supported.)
	For example, searching for groups that start with S would narrow your search results to something like this:
	`cn=Sales,ou=groups,o=MyOrg`
	`cn=Service,ou=groups,o=MyOrg`
	`cn=Shipping,ou=groups,o=MyOrg`
	Searching for groups that start with Se would return:
	`cn=Service,ou=groups,o=MyOrg`

4 Click *Go*.

The results of your search appear in the *Results* list.

5 *Select* the users, groups, or containers you want to assign to the page, then click the *Add* (>) button.

TIP: Hold down the *Control* key to make multiple selections.

6 Enable or disable *page lock-down* as follows:

If you want to	Do this
Lock down the page so only User Application Administrators can view it	Check **View Permission Set to Admin Only**
Allow all assigned users, groups, and containers to view the page	Uncheck **View Permission Set to Admin Only** **NOTE:** If you uncheck this setting but there are no users, groups, or containers explicitly assigned to the page, then **everyone will have View permission** for this page.

7 Click *Save*, then *Close*.

7.4.2 Assigning shared page owners

Users who own shared pages can modify the content of the pages they own and change the preferences of portlets on those pages.

To assign Ownership permission for shared pages:

1. Open a page on the Maintain Shared Pages panel, then click the *Assign Permissions* page task (at the bottom of the panel).

   The *Page Permissions* dialog displays in a new browser window (as shown in the previous procedure).

2. Go to the *Ownership* tab.

3. Specify values for the following *search settings*:

Setting	What to do
Search for	Select one of the following from the dropdown menu: • Users • Groups • Containers

Starts with	If you want to:
	• **Find all** available objects of your specified type (user, group, or container), then make this setting blank.
	• **Find a subset** of those objects, then enter the starting character(s) of the CN values you want. (Case is not considered. Wildcards are not supported.)
	For example, searching for groups that start with S would narrow your search results to something like this:
	`cn=Sales,ou=groups,o=MyOrg`
	`cn=Service,ou=groups,o=MyOrg`
	`cn=Shipping,ou=groups,o=MyOrg`
	Searching for groups that start with Se would return:
	`cn=Service,ou=groups,o=MyOrg`

4. Click *Go*.

   The results of your search appear in the *Results* list.

5. *Select* the users, groups, or containers you want to assign to the page, then click the *Add* (>) button.

   TIP: Hold down the *Control* key to make multiple selections.

6. Enable or disable *page lock-down* as follows:

If you want to	Do this
Lock down the page so only User Application Administrators can work with it	Check **Ownership Permission Set to Admin Only**
Allow all assigned users, groups, and containers to work with the page	Uncheck **Ownership Permission Set to Admin Only**
	NOTE: If you uncheck this setting but there are no users, groups, or containers explicitly assigned to the page, then **everyone will have Ownership permission** for this page.

7 Click *Save*, then *Close*.

7.4.3 Enabling user access to the Create User or Group page

By default, only User Application Administrators can see and use the *Create User or Group* page, which is a shared page on the *Identity Self-Service tab* of the Identity Manager user interface. But, where appropriate, a User Application Administrator can *assign permission for one or more end users* to access that page too. For instance, selected people in administration or management positions might need the ability to create users, groups, or task groups themselves.

To give users access to the Create User or Group page:

1 On the *Maintain Shared Pages* panel, open the page named *Create User or Group*.

2 Use the *Assign Permissions* page task to give *View permission* to the appropriate users, groups, or containers for the Create User or Group shared page.

3 Switch from Page Admin to *Portlet Admin*, and open the portlet registration named *CreatePortlet* (which is used on the Create User or Group page).

4 Use the *Security* panel to give *List and Execute permissions* to the appropriate users, groups, or containers for the CreatePortlet portlet registration.

For more information about assigning permissions for portlets, see Chapter 9, “Portlet Administration,” on page 171.

5 Go to *iManager* and use an administrator account to *log in to the tree* for your identity vault.

6 Make sure that the people who will be using Create User or Group have *Create rights for the [Entry Rights] property* on the container(s) in which objects (users, groups, or task groups) will be created.

For example, you can *modify trustees* for a chosen container and add the appropriate users, groups, or containers as trustees. Then, for each trustee, you can assign the following rights:

Property name	Assigned rights	Inherit
[All Attributes Rights]	• Compare • Read • Write	Yes (select this check box)

Property name	Assigned rights	Inherit
[Entry Rights]	• Browse • Create	Yes (select this check box)

If you don’t assign the necessary rights in the identity vault (or if those rights can’t somehow be derived), an end user may get an *error message* such as this one from Create User or Group:

```
User 'cn=mmackenzie,ou=users,ou=idmsample,o=novell' does not have
permission
to create 'cn=MyNewGroup,ou=groups,ou=idmsample,o=novell' or
modify related
objects.
```

To learn how the Create User or Group page is used (by those with access to it), see the *Identity Manager User Application: User Guide*.

7.4.4 Enabling user access to individual Administration pages

By default, only User Application Administrators can access the *Administration tab* of the Identity Manager user interface and the *pages* contained on that tab (Page Admin, Themes, Portlet Admin, Portal, Security, Logging, Caching, Tools). But if necessary, a User Application Administrator can *assign permission for one or more end users* to see and use specific pages on the Administration tab. One example might be a small group of users who need to change themes periodically, even though they are not User Application Administrators.

To give users access to individual Administration pages:

1. On the *Maintain Container Pages* panel, open *Admin Container Page*.

   This is the container page that’s used when you go to the Administration tab of the Identity Manager user interface.
2. Use the *Assign Permissions* page task to give *View permission* to the appropriate users, groups, or containers for Admin Container Page.
3. On the *Maintain Shared Pages* panel, open the appropriate Administration page (one of the shared pages under the category *Administration*).
4. Use the *Assign Permissions* page task to give *View and Ownership permissions* to the appropriate users, groups, or containers for that shared page.
5. Make sure the specified users, groups, or containers have *Execute permission for each portlet* used on a specified page (if you have restricted those portlets).

   For more information about assigning permissions for portlets, see Chapter 9, “Portlet Administration,” on page 171.

7.5 Setting default pages for groups

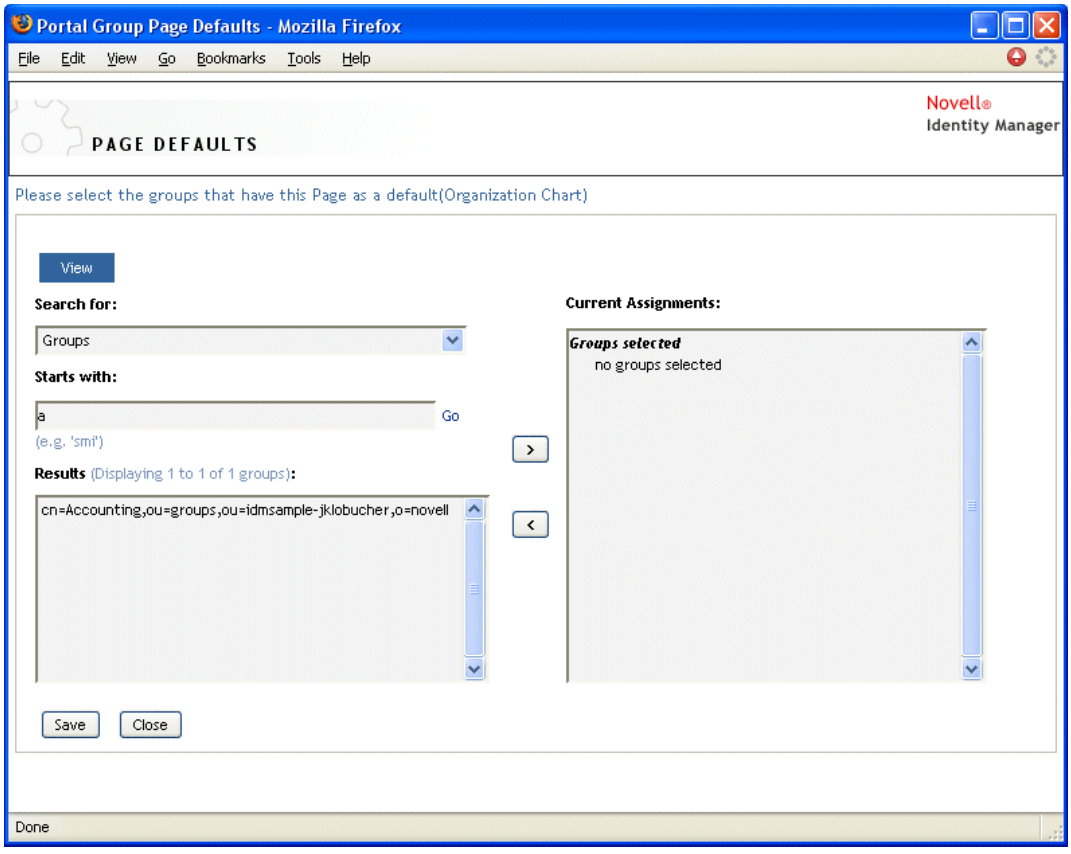
You can assign a *default container page* and a *default shared page* for any authorized group of users. These settings affect which container page those users see when they log in and which shared page they see on the container page.

When users belong to multiple groups with default page assignments, Navigation Priority is used in determining which container page and shared page to display.

To assign a default container page or a default shared page to a group:

1 Open a page on the Maintain Container Pages panel or the Maintain Shared Pages panel, then click the *Set As Default* page task (at the bottom of the panel).

The *Page Defaults* dialog displays in a new browser window:



2 Specify values for the following *search settings*:

Setting	What to do
Search for	(**Groups** is automatically selected.)

Setting	What to do

Starts with	If you want to:
	• **Find all** available groups, then make this setting blank.
	• **Find a subset** of those groups, then enter the starting character(s) of the CN values you want. (Case is not considered. Wildcards are not supported.)
	For example, searching for groups that start with S would narrow your search results to something like this:
	`cn=Sales,ou=groups,o=MyOrg`
	`cn=Service,ou=groups,o=MyOrg`
	`cn=Shipping,ou=groups,o=MyOrg`
	Searching for groups that start with Se would return:
	`cn=Service,ou=groups,o=MyOrg`

3 Click *Go*.

The results of your search appear in the *Results* list.

4 *Select* the groups for whom this page is to be a default, then click the *Add* (>) button.

TIP: Hold down the *Control* key to make multiple selections.

5 Click *Save*, then *Close*.

7.6 Selecting a default shared page for a container page

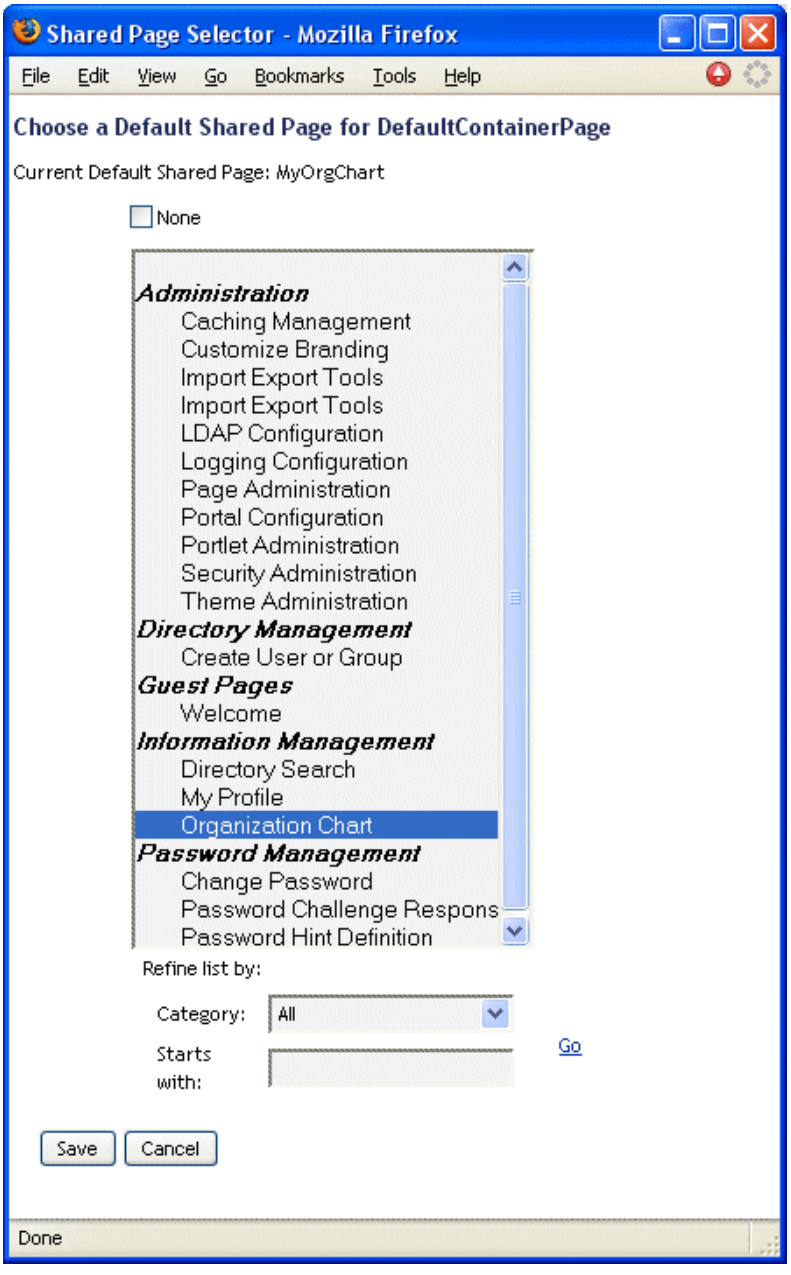
You can assign a default shared page to each container page you have. The user interface considers this page assignment when determining what to display.

To assign a default shared page for a container page:

1 Open a container page on the *Maintain Container Pages* panel.

2 In the page properties section, look for *Default Shared Page* and click *Select Default*.

The *Choose a Default Shared Page* dialog displays in a new browser window:



3 If the shared page list is long, you can *refine* it (by category or starting text) to more easily find the desired page.

4 *Select* a shared page to use as the default for the container page (or check *None* for no default).

5 Click *Save* to accept your selection and close the dialog.

6 Click *Save Page* (at the bottom of the page properties section).

Theme Configuration

8

This chapter tells you how to use the *Themes* page on the *Administration tab* of the Identity Manager user interface. Topics include:

For more general information about accessing and working with the Administration tab, see Chapter 6, “Using the Administration Tab,” on page 125.

8.1 About theme configuration

You can use the Themes page to control the look and feel of the Identity Manager user interface.

A *theme* is a set of visual characteristics that apply to the entire user interface (including the guest and login pages, the Identity Self-Service tab, the Requests & Approvals tab, and the Administration tab). There’s always exactly one theme in effect for the user interface. The Themes page offers a choice of several themes, in case you want to switch to a different one.

The Themes page also enables you to:

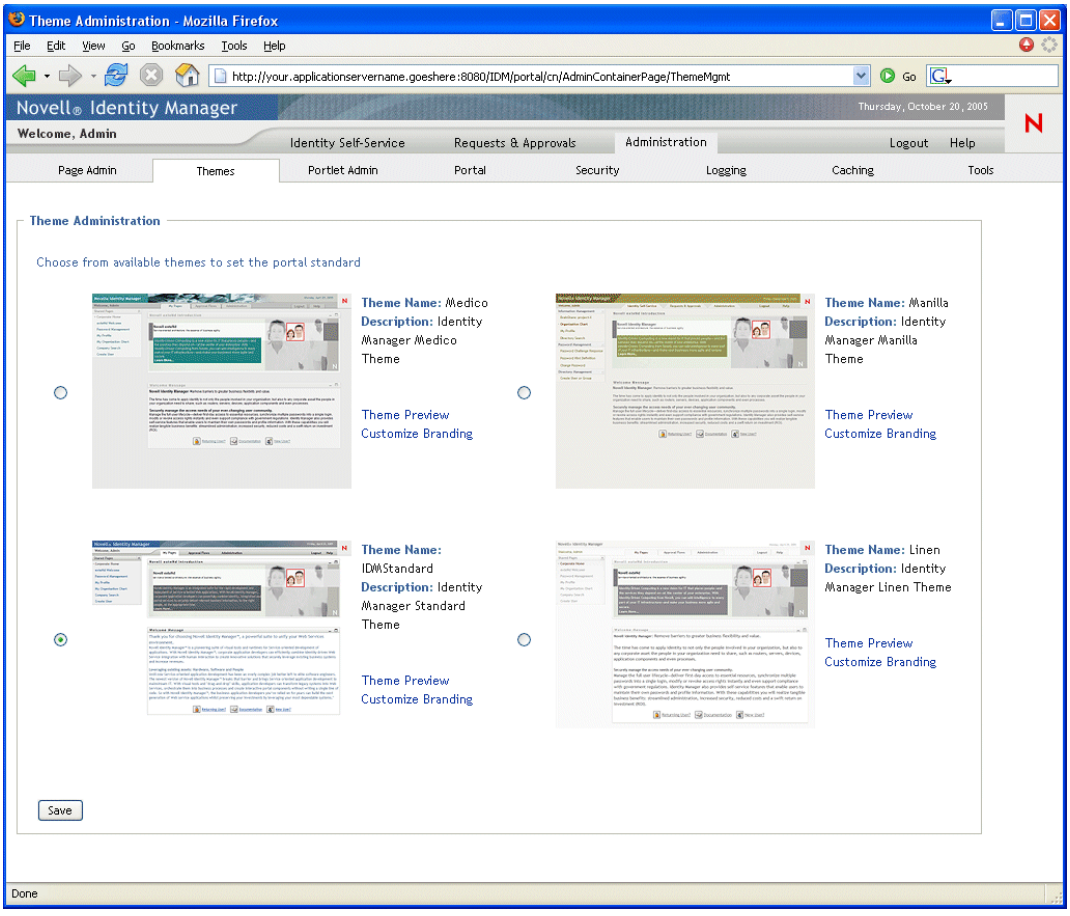
- *Preview* each theme choice to see how it looks
- *Customize* any theme choice to reflect your own branding (logo, etc.)

8.2 Previewing a theme

Before choosing a theme, you can preview how it will change the look of the Identity Manager user interface.

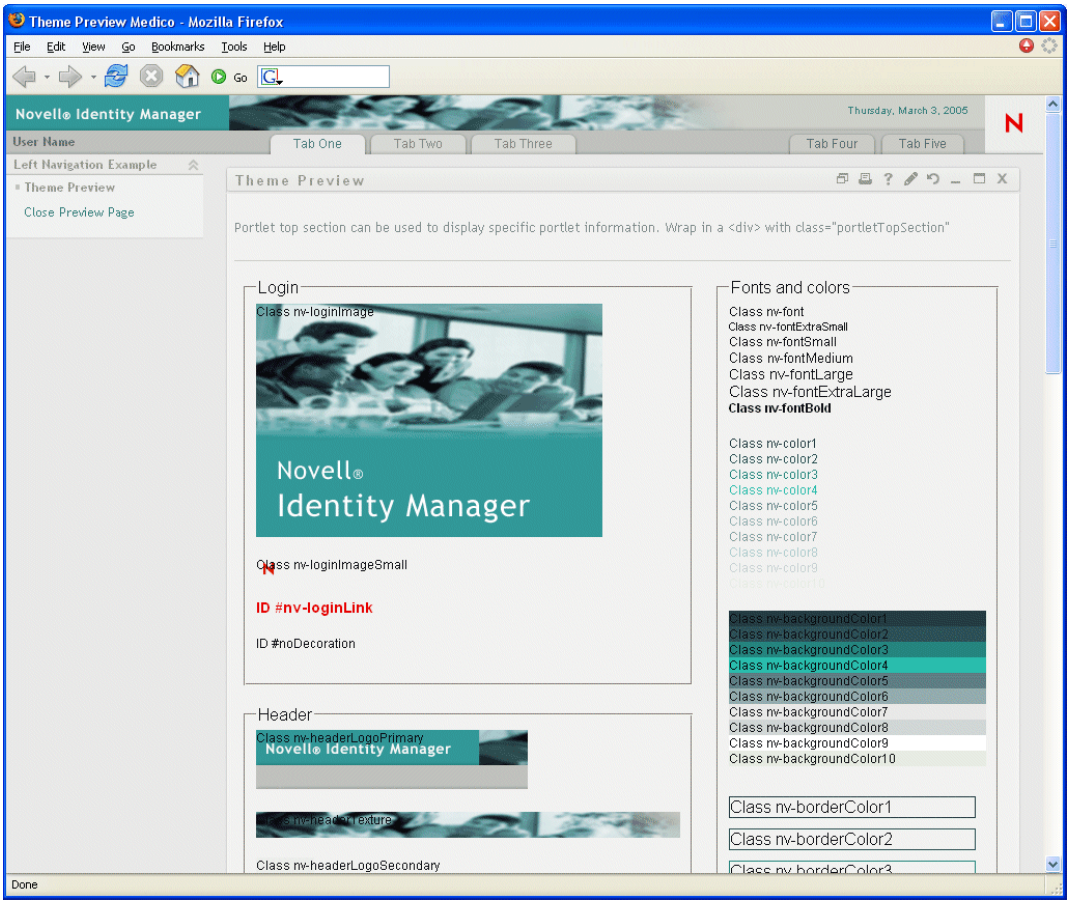
To preview a theme:

1. Go to the *Themes* page:



2 Find a theme that you are interested in, then click the corresponding *Theme Preview* link.

The preview for that theme displays in a new browser window:



3 *Scroll* through the preview to see the characteristics of this theme.

4 When you're done, click *Close Preview Page* (in the top-left corner) or close the preview window manually.

8.3 Choosing a theme

When you find a theme that you like, you can choose to make it the *current theme* for the Identity Manager user interface.

To choose a theme:

1 Go to the *Themes* page.

2 Click the *radio button* for the theme you want.

3 Click the *Save* button.

The look of the user interface changes to reflect your chosen theme.

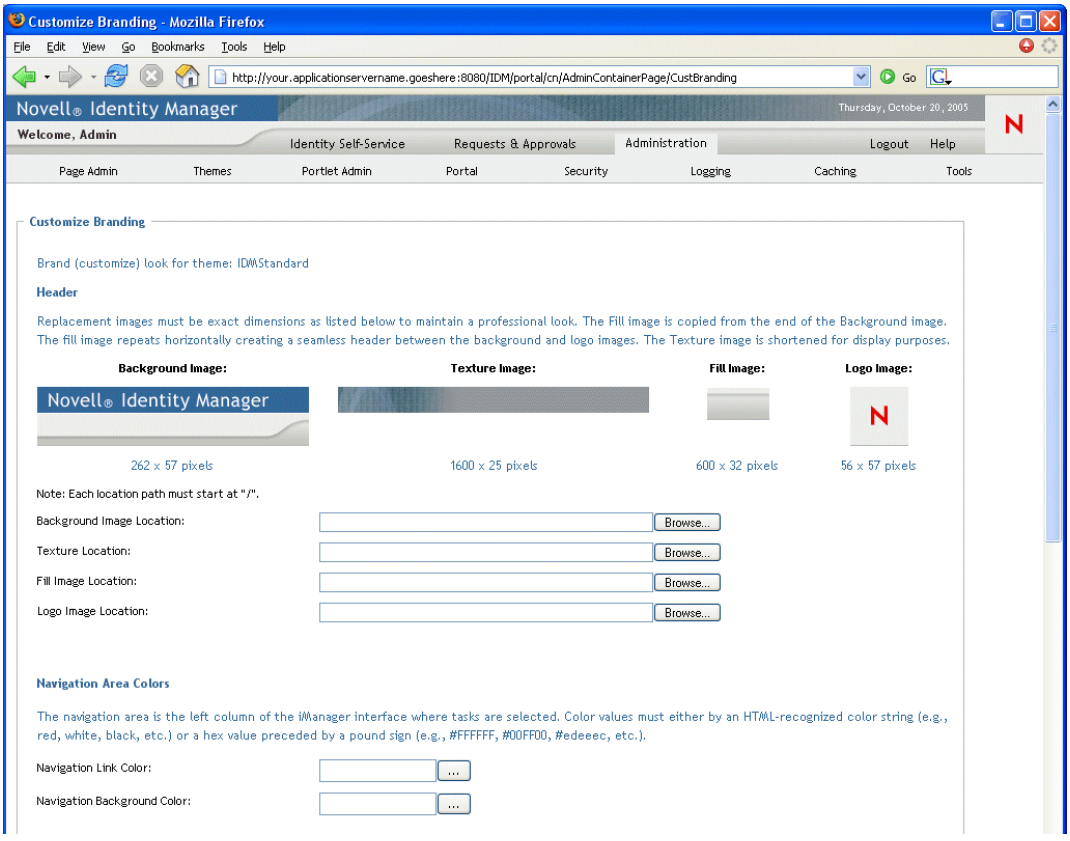
8.4 Customizing a theme's branding

You can tailor any theme by substituting your own *images* and changing some *color settings*. This enables you to give the Identity Manager user interface a custom look to meet the branding requirements of your company or organization.

To customize the branding for a theme:

1. Go to the *Themes* page.
2. Find a theme that you want to tailor, then click the corresponding *Customize Branding* link.

   The Themes page displays the Customize Branding settings for that theme:



3. Specify *your customizations* in these settings (as needed), including:

- Header images
- Navigation-area colors
- Login images

Follow the onscreen instructions for specifying each setting.

4 Click the *Save* button.

If you're editing the current theme, the look of the user interface changes to reflect your customizations. (If you want to undo all of your customizations to the theme, click the *Reset* button.)

NOTE: The *Theme Preview* button is available while you make customizations, but be aware that it always displays the *original characteristics* of the theme. It does not show your changes.

---

5 When you're done working on this theme, click the *Back to Theme Selector* button.

Portlet Administration

9

This chapter tells you how to use the *Portlet Admin* page on the *Administration tab* of the Identity Manager user interface. Topics include:

For more general information about accessing and working with the Administration tab, see Chapter 6, “Using the Administration Tab,” on page 125.

9.1 About portlet administration

You can use the Portlet Admin page to control the *portlets* available in the Identity Manager user interface and who has permission to access them. Portlets are pluggable user-interface elements (based on a *Java standard*) that provide the content for pages in the user interface (including container pages and shared pages).

Managing portlets involves working with the following:

What you work with	Description
Portlet applications	Java Portlet 1.0-compliant WARs that contain the portlet deployment descriptor portlet.xml and, optionally, other portlet runtime artifacts. See Section 9.2, “Administering portlet applications,” on page 171.
Portlet definitions	Descriptors (read from portlet.xml) that specify portlet configuration parameters. There is one definition for each portlet in an application. See Section 9.3, “Administering portlet definitions,” on page 174.
Portlet registrations	Registrations of portlets, based on their definitions. Multiple registrations of the same portlet can exist in a single portlet application. See Section 9.4, “Administering registered portlets,” on page 178.

For details on the portlets provided with the Identity Manager user interface, see Part IV, “Portlet Reference,” on page 225. To learn about using portlets on container pages and shared pages, see Chapter 7, “Page Administration,” on page 131.

9.2 Administering portlet applications

When the Identity Manager user application is installed, *IDM.war* is deployed to your application server and automatically registered as a portlet application. IDM.war (which may be renamed during the install) includes all of the portlets used in the default configuration of the Identity Manager user interface. It also includes some additional portlets that aren’t used by default. (The IDM.war portlets are described in Part IV, “Portlet Reference,” on page 225.)

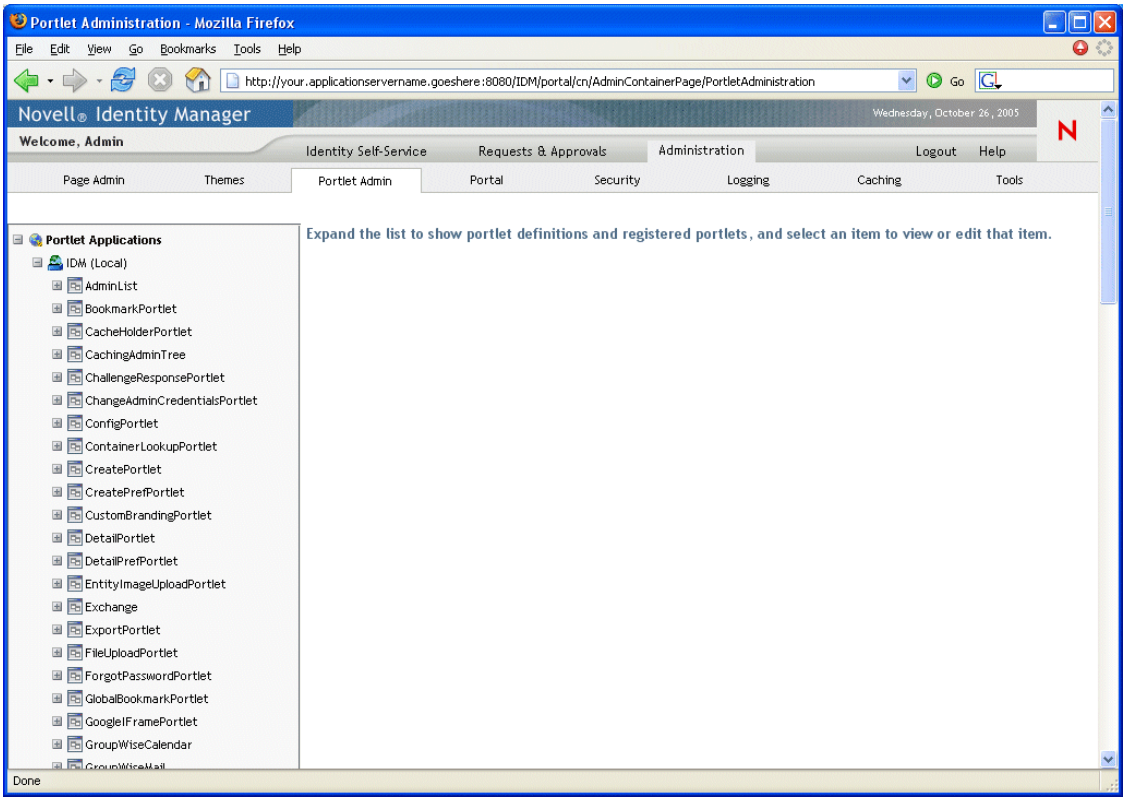
But you aren't limited to using portlets from IDM.war. If you deploy any other *standard portlet applications* (Java Portlet 1.0-compliant WARs) to your application server, you'll be able to work with those portlet applications and their portlets in the Identity Manager user interface. For example, you'll see those portlet applications listed along with IDM.war on the Portlet Admin page.

The Portlet Admin page enables you to *administer IDM.war and other portlet applications* in the following ways:

- Section 9.2.1, "Accessing portlet applications on the server," on page 172
- Section 9.2.2, "Viewing information about portlet applications," on page 172
- Section 9.2.3, "Unregistering portlet applications," on page 173

9.2.1 Accessing portlet applications on the server

When you go to the Portlet Admin page, it automatically *displays a list* of the portlet applications (IDM.war and any others) that are deployed to your application server. This list appears on the left as a tree that you can *expand and navigate* to administer a selected portlet application and its contents:



9.2.2 Viewing information about portlet applications

You can view the following read-only information about a listed portlet application:

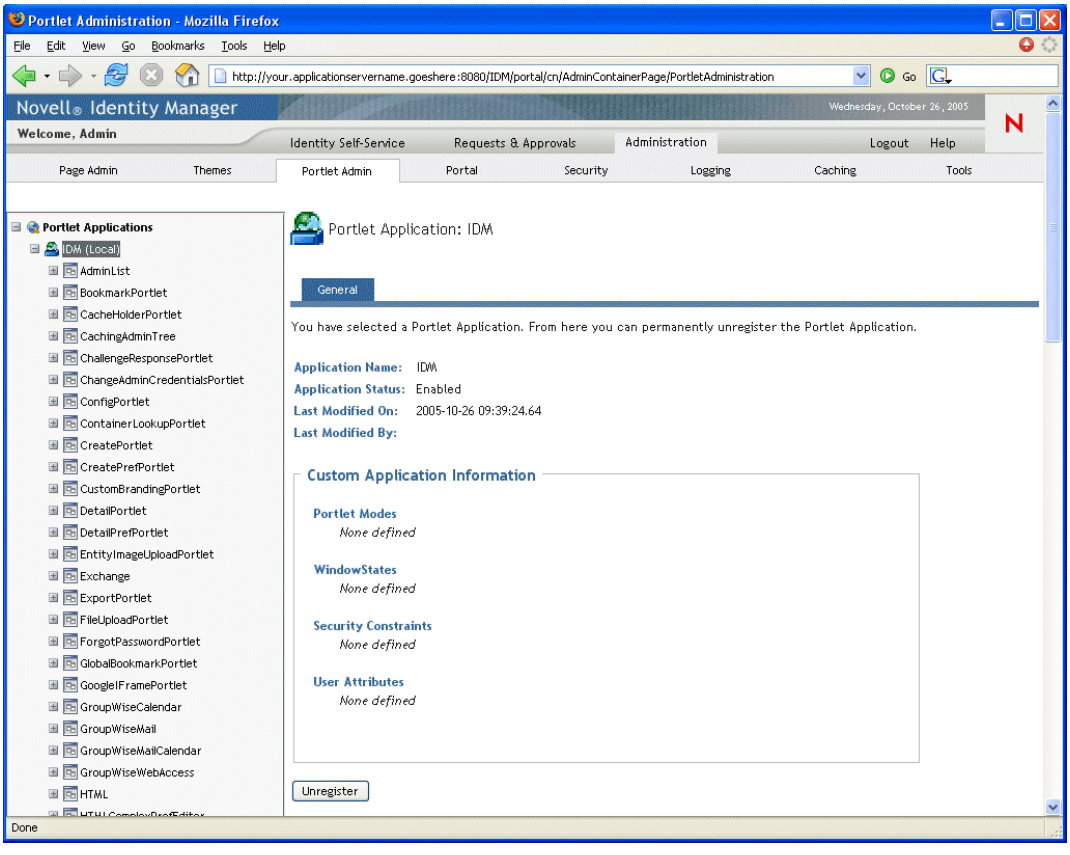
- Name
- Status (enabled or disabled)
- Date last modified

- User who last modified the application
- Custom application information (if any): portlet modes, window states, security constraints, and user attributes

To view information about a portlet application:

- In the Portlet Applications list, *select* the portlet application that you want to learn about.

  A *General* panel displays on the right, showing information about the selected portlet application:



9.2.3 Unregistering portlet applications

When you want to remove a portlet application from your application server, you must *unregister* the portlet application before undeploying it. Otherwise, the portlet application is automatically redeployed when the server restarts.

When you unregister a portlet application, all related preferences and settings are removed from the database that stores your application data.

NOTE: You cannot unregister the *local* portlet container, which is a portlet application that is local to the portal. The local portlet container manages portlets that are contained within the portal (Identity Manager user application).

To unregister a portlet application:

1. In the Portlet Applications list, *select* the portlet application that you want to unregister.

   A *General* panel displays on the right (as shown in the previous procedure).

2. Click *Unregister*.

   A confirmation window appears.

3. Click *OK* to confirm the action.

   When the process completes, the unregistered portlet application is removed from the Portlet Applications list.

4. To remove the portlet application from the application server, use your server's tools to *undeploy the archive* containing the portlet application.

NOTE: To reregister an unregistered portlet application, you must *redeploy* it.

9.3 Administering portlet definitions

The Portlet Admin page enables you to perform the following tasks related to *portlet definitions* in a portlet application:

- Section 9.3.1, "Accessing portlet definitions in the deployed portlet application," on page 174
- Section 9.3.2, "Registering portlet definitions," on page 175
- Section 9.3.3, "Viewing information about portlet definitions," on page 176

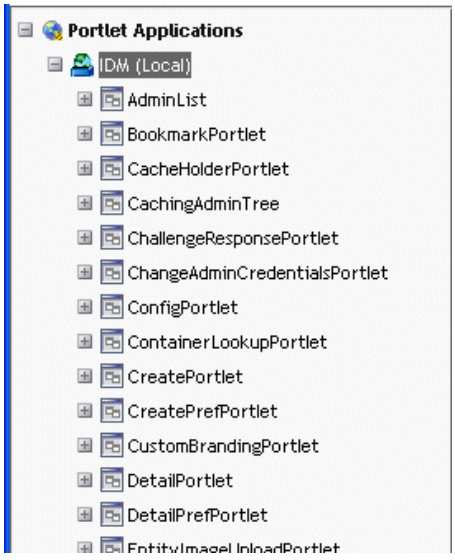
9.3.1 Accessing portlet definitions in the deployed portlet application

The Portlet Applications list shows the portlet definitions in a selected portlet application.

To access portlet definitions in the deployed portlet application:

- In the Portlet Applications list, *expand* the portlet application whose portlet definitions you want to access.

  The tree displays all of the portlet definitions under that portlet application:



9.3.2 Registering portlet definitions

Before you can use a portlet, you must register that portlet definition with the portal (Identity Manager user application). A registered portlet definition is called a *portlet registration*. You can create multiple registrations for a single portlet, which enables you to put multiple instances of that portlet on the same page.

The portlet registration inherits all the *preferences and settings* of the portlet class, but you can modify these values in the following ways:

- *When registering* the portlet definition — see Section 9.4, “Administering registered portlets,” on page 178
- *When adding an instance* of the portlet to a page — see Chapter 7, “Page Administration,” on page 131

All portlets that ship with the Identity Manager user application are *automatically registered*.

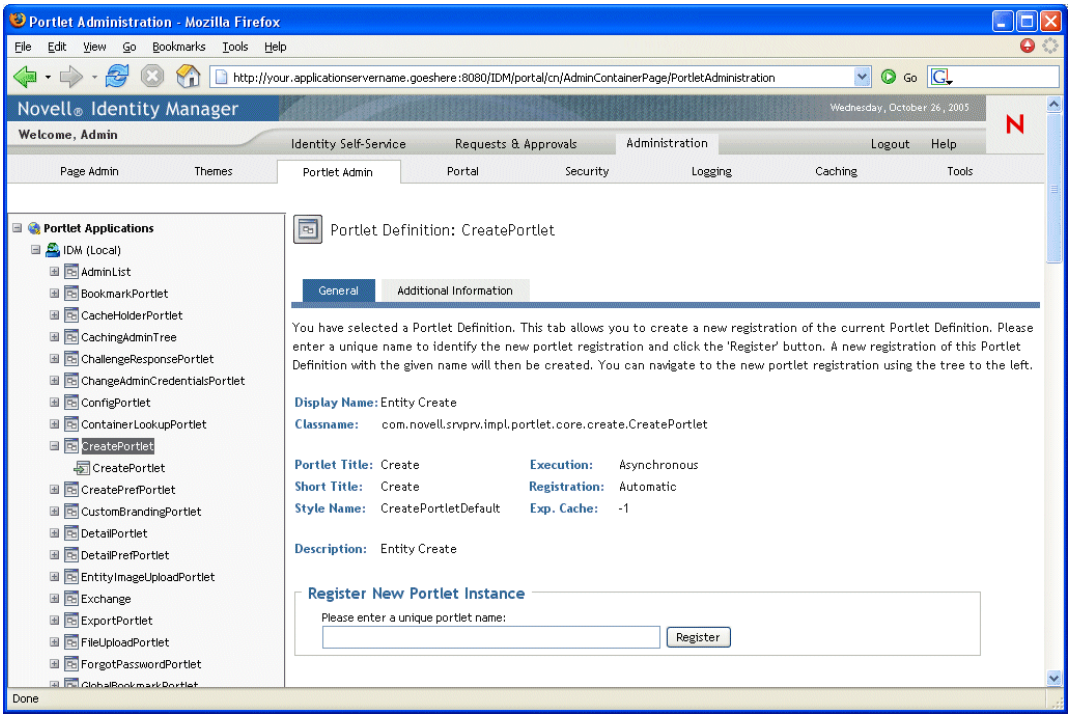
Edit mode If the portlet definition provides an Edit mode, the end user can modify specific preferences of the portlet registration at runtime, according to the logic of the portlet’s doEdit() method.

The Identity Manager user application also provides a default implementation for Edit mode. If the doEdit() method is not explicitly implemented, a default preference sheet is displayed.

To register a portlet definition:

1 In the Portlet Applications list, *select* the portlet definition for which you want to create a portlet registration.

A *General* panel displays on the right:



Note that all *existing registrations* of the selected portlet are listed in the Portlet Applications tree (on the left), under the corresponding portlet definition name.

2 In the *Register New Portlet Instance* text box, enter a unique name for the portlet registration, then click *Register*.

The new portlet registration is created and listed in the Portlet Applications tree.

3 If you want to modify the preferences and settings of the new portlet registration, see Section 9.4, “Administering registered portlets,” on page 178.

9.3.3 Viewing information about portlet definitions

You can view the following read-only information about a listed portlet definition:

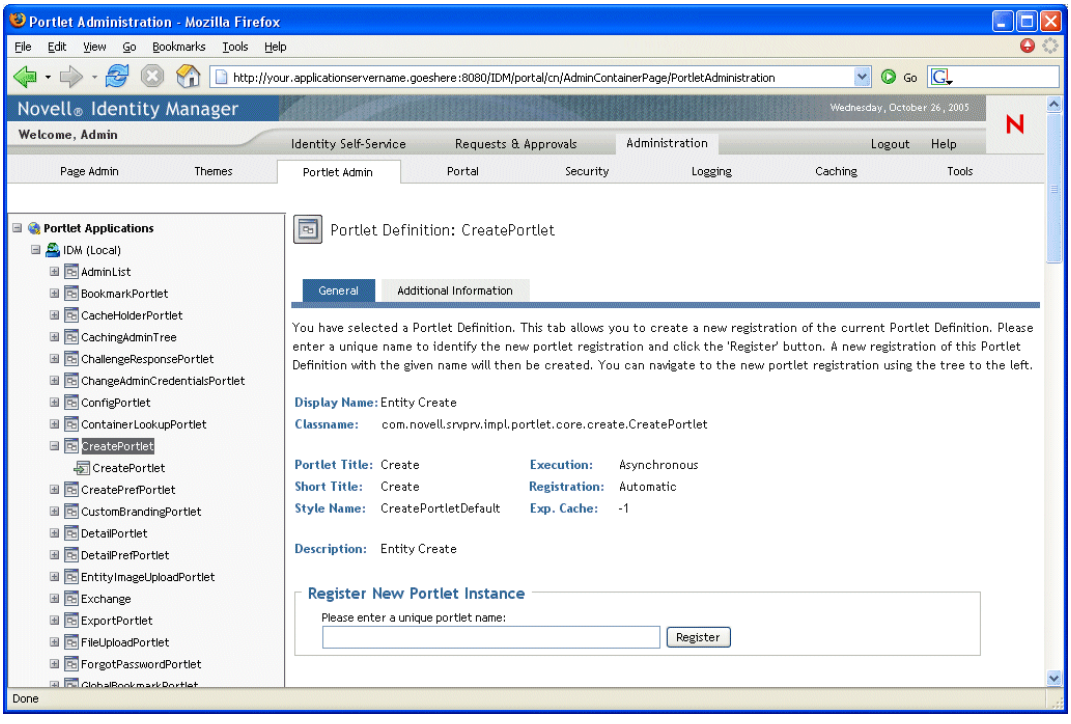
- Display name
- Class name
- Portlet title
- Type of execution (synchronous or asynchronous)
- Short title
- Type of registration
- Style name
- Cache expiration time
- Description
- Initialization parameters
- Keywords
- Supported mime types

- Modes supported by the portlet
- Supported locales
- Supported devices
- Security roles

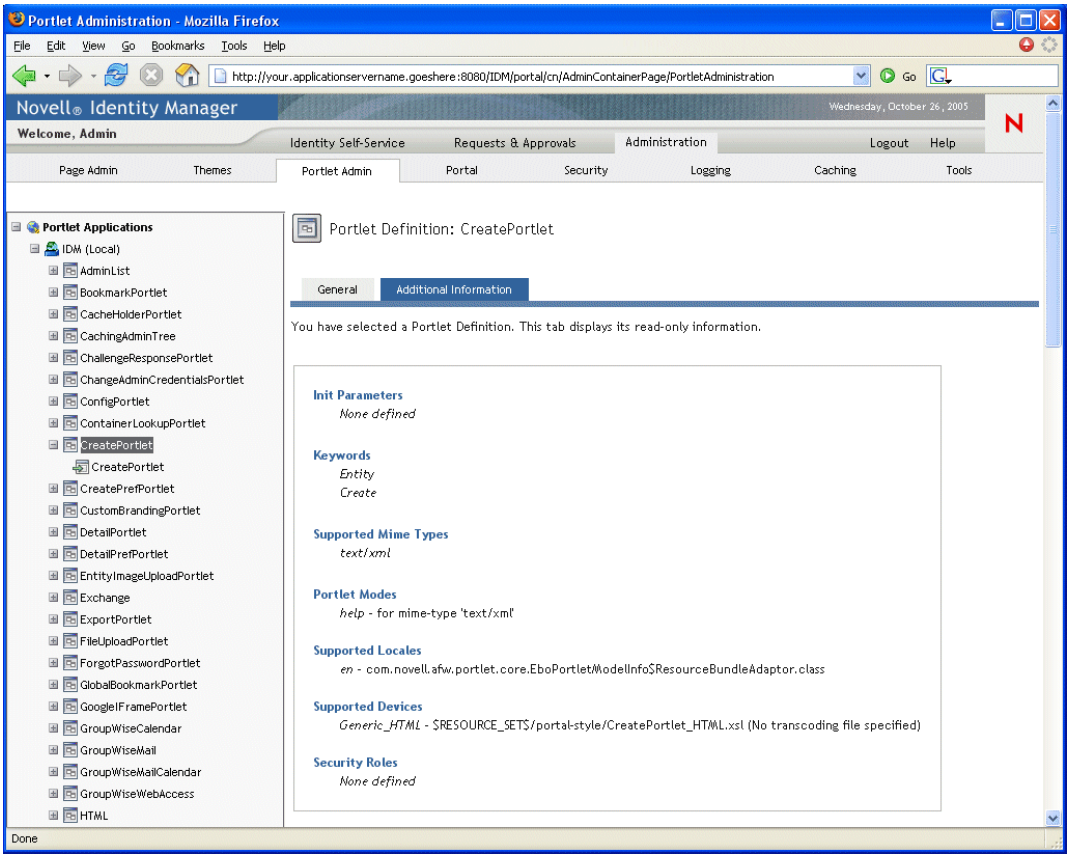
To view information about portlet definitions:

1 In the Portlet Applications list, *select* the portlet definition that you want to learn about.

A *General* panel displays on the right, showing information about the selected portlet definition:



2 Go to the *Additional Information* panel to view further details about the selected portlet definition:



9.4 Administering registered portlets

The Portlet Admin page enables you to perform the following tasks related to *portlet registrations* in a portlet application:

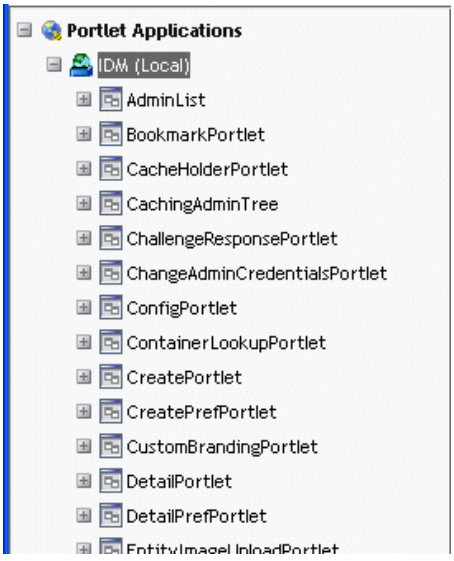
9.4.1 Accessing portlet registrations in the deployed portlet application

The Portlet Applications list shows the portlet registrations for each portlet definition in a selected portlet application.

To access portlet registrations in the deployed portlet application:

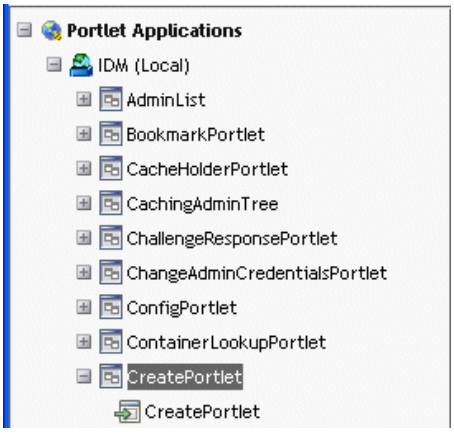
1. In the Portlet Applications list, *expand the portlet application* whose portlet definitions and registrations you want to access.

   The tree displays all of the portlet definitions under that portlet application:



2. *Expand the portlet definition* whose portlet registrations you want to access.

   The tree displays all of the portlet registrations under that portlet definition:



9.4.2 Viewing information about portlet registrations

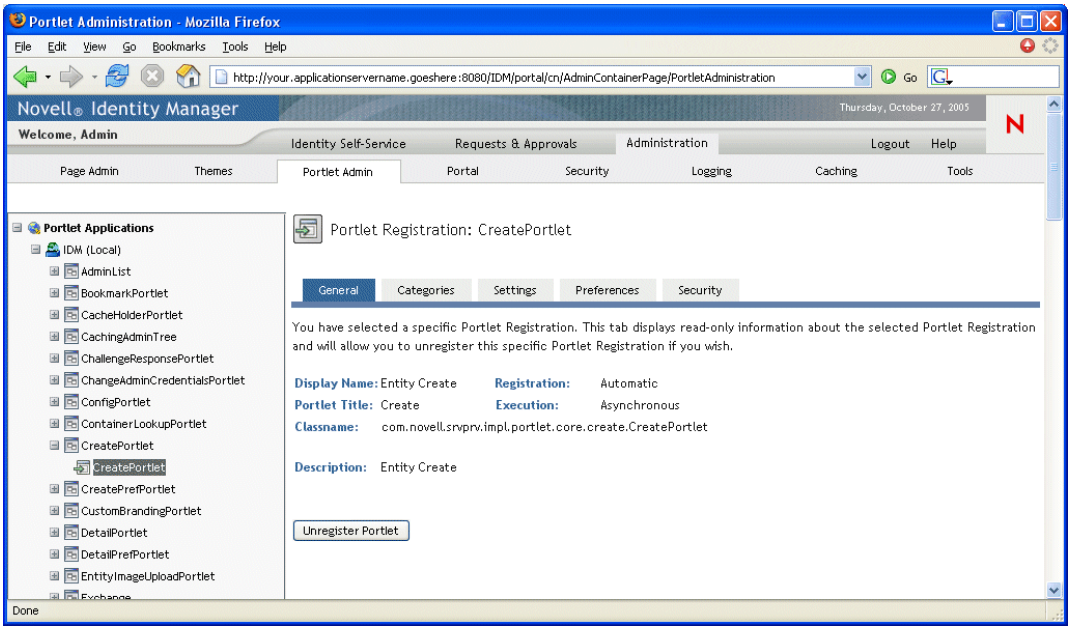
You can view the following read-only information about a listed portlet registration:

- Display name
- Type of registration
- Portlet title
- Type of execution (synchronous or asynchronous)
- Class name
- Description

To view information about portlet registrations:

- In the Portlet Applications list, *select* the portlet registration that you want to learn about.

  A *General* panel displays on the right, showing information about the selected portlet registration:



9.4.3 Assigning categories to portlet registrations

To facilitate searching for specific portlets in a portlet application, you can organize portlet registrations by category.

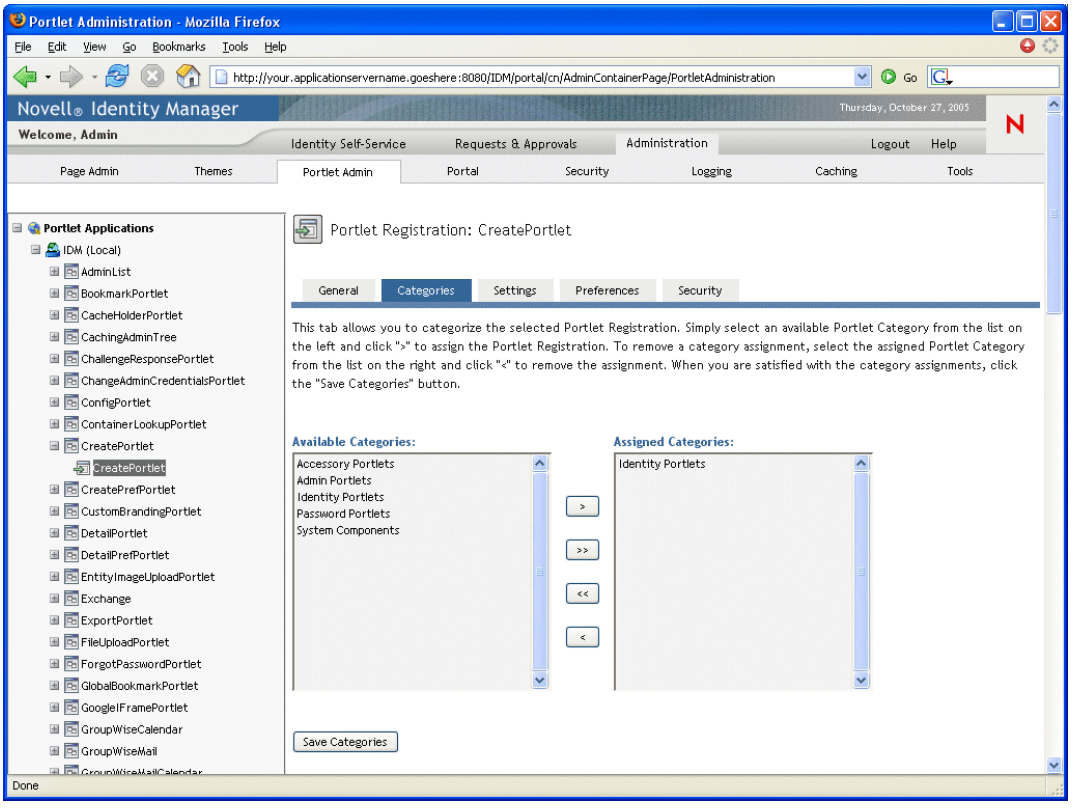
To assign categories to portlet registrations:

1. In the Portlet Applications list, *select* the portlet registration that you want to categorize.

   A *General* panel displays on the right.

2. Go to the *Categories* panel.

   This panel displays lists of available and assigned categories for the selected portlet registration:



3 Update the *Assigned Categories* list, as appropriate:

If you want to	Do this
Assign one or more categories to the portlet registration	Select each category you want to assign and click >
Assign all categories to the portlet registration	Click >>
Remove one or more category assignments	Select each category you want to remove and click <
Remove all category assignments	Click <<

4 Click *Save Categories*.

9.4.4 Modifying settings for portlet registrations

Portlet settings define how the portal (Identity Manager user application) interacts with individual portlets. Each portlet is configured with these settings:

- Title
- Maximum timeout
- Requires authentication
- Display title bar
- Hidden from user
- Options defined in the portlet application

Standard Java Portlet 1.0 settings are defined in the portlet deployment descriptor (portlet.xml) of the portlet application WAR. You can change the values of these settings on a registration-by-registration basis by using the Portlet Admin page. In this case, the new values take effect only for the selected portlet registration.

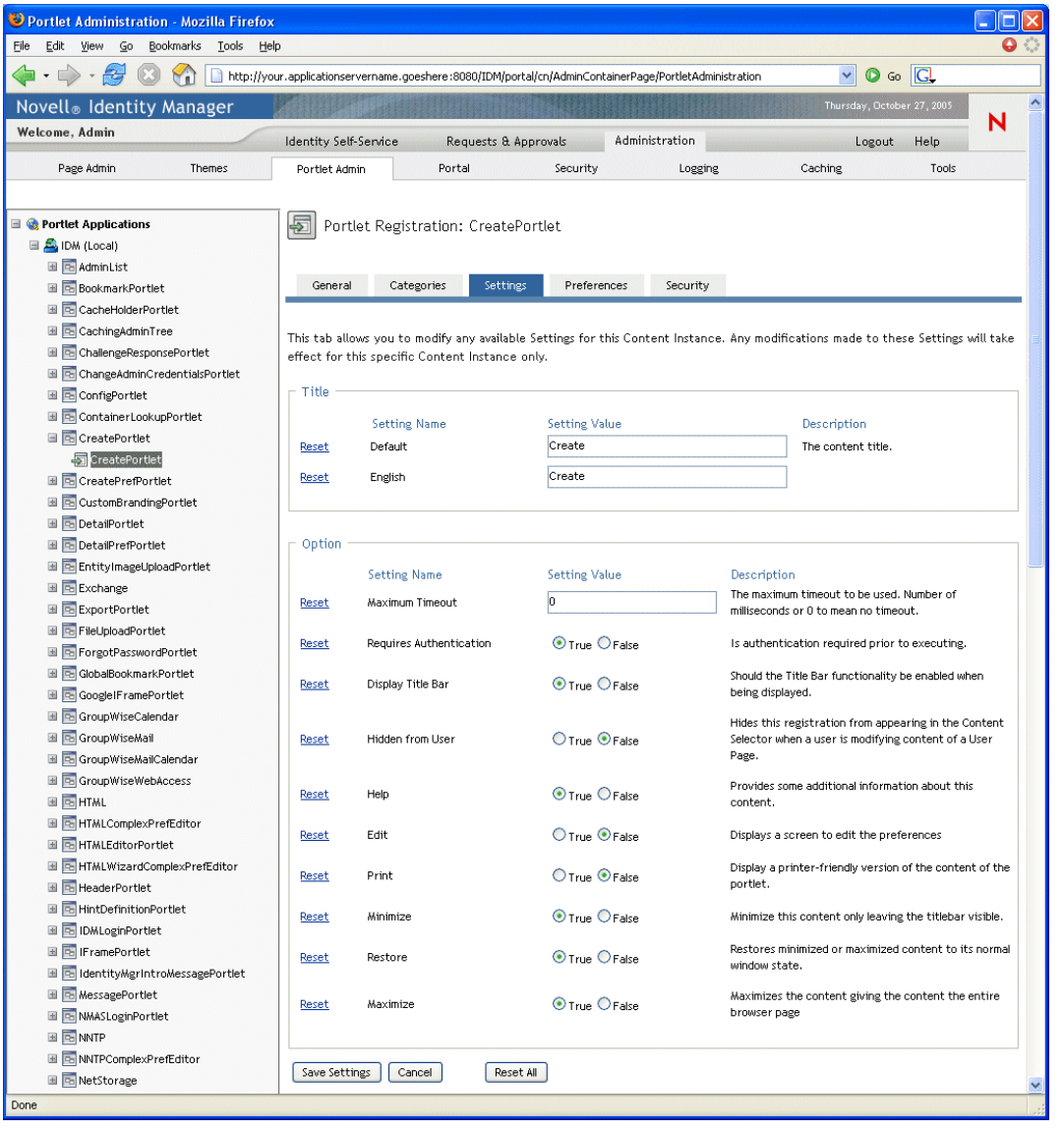
To modify portlet registration settings:

1. In the Portlet Applications list, *select* the portlet registration whose settings you want to modify.

   A *General* panel displays on the right.

2. Go to the *Settings* panel.

   This panel displays the current settings for the selected portlet registration:



3. *Modify* settings, as appropriate.

   While working on this panel, note that you can also perform the following actions:

If you want to	Do this
Discard your unsaved changes	Click **Cancel**
Return all settings for this portlet registration to their default values (as defined in the corresponding portlet definition)	Click **Reset All**
Return an individual setting to its default value	Click the **Reset** link beside that setting

4 Click *Save Settings*.

9.4.5 Modifying preferences for portlet registrations

Portlet preferences are defined by the portlet developer at design time in the portlet deployment descriptor. Preferences vary from portlet to portlet, based on the portlet developer's implementation.

You can change the values of these preferences on a registration-by-registration basis by using the Portlet Admin page. In this case, the new values take effect only for the selected portlet registration.

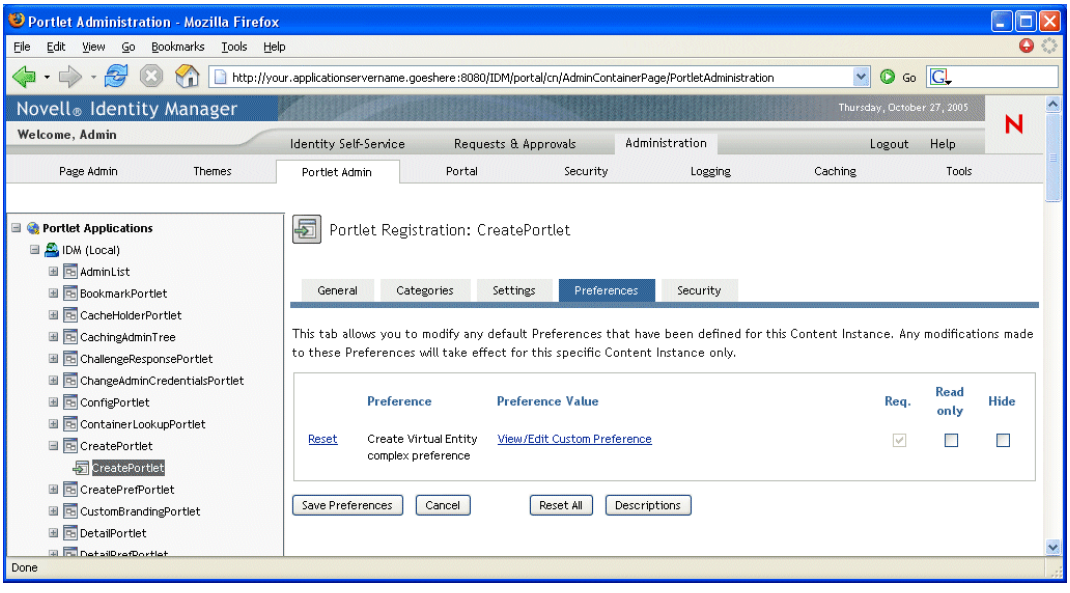
To modify portlet registration preferences:

1 In the Portlet Applications list, *select* the portlet registration whose preferences you want to modify.

A *General* panel displays on the right.

2 Go to the *Preferences* panel.

This panel displays the current preferences for the selected portlet registration:



3 *Modify* preferences, as appropriate.

While working on this panel, note that you can also perform the following actions:

If you want to	Do this
Display more information about the preferences	Click **Descriptions**
Discard your unsaved changes	Click **Cancel**
Return all preferences for this portlet registration to their default values (as defined in the corresponding portlet definition)	Click **Reset All**
Return an individual preference to its default value	Click the **Reset** link beside that preference

4 To modify the *localized version* of a preference for each locale specified in the portlet definition, follow these steps:

4a Click the *Detail* link beside that preference (if available).

The panel displays the preference values for each locale.

4b *Modify* values, as appropriate.

4c Click *OK* to apply your changes and return to the main preferences list.

5 Click *Save Preferences*.

9.4.6 Assigning security permissions for portlet registrations

You can assign the following security permissions to users, groups, and containers for portlet registrations:

Permission	Description
List	Users can **view** the portlet registration from a selection list
Execute	Users can **run** the portlet registration on a portal page

When you modify security permissions, the new values take effect only for the selected portlet registration.

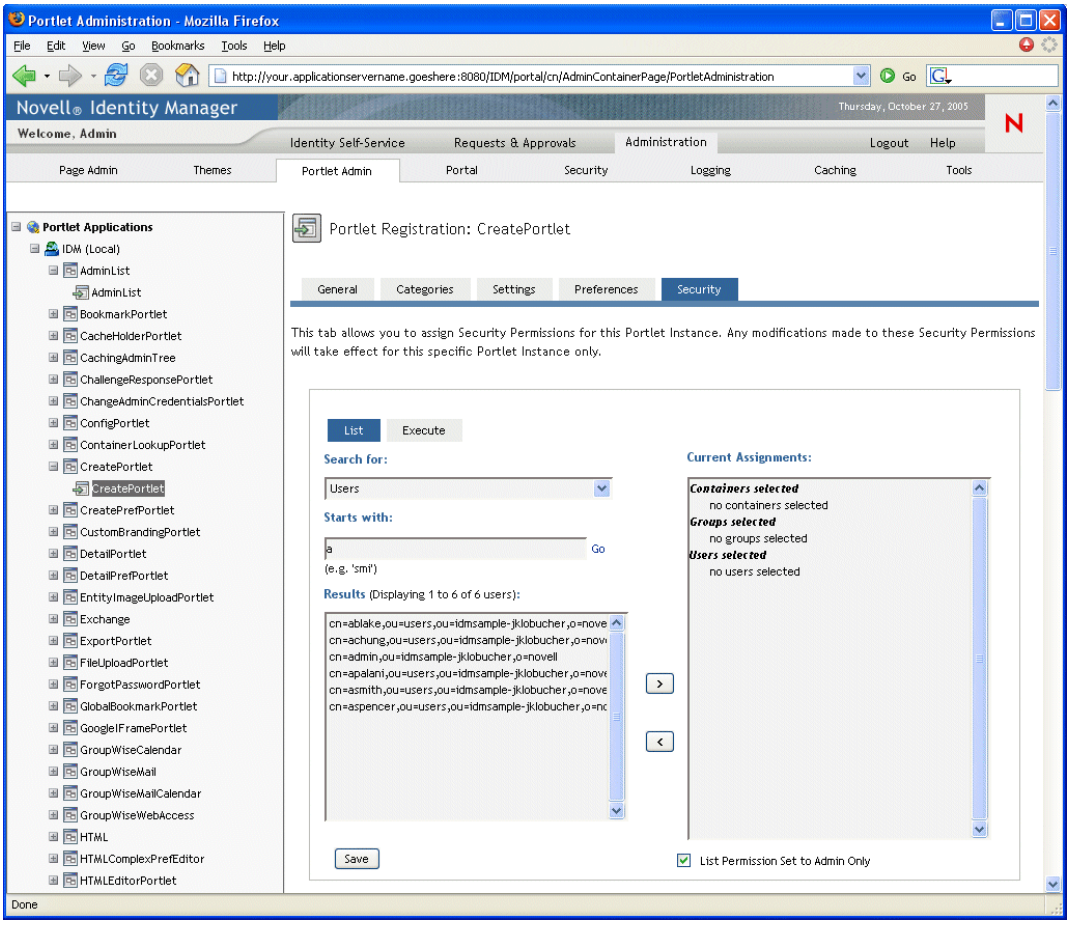
To assign security permissions for portlet registrations:

1 In the Portlet Applications list, *select* the portlet registration whose security permissions you want to modify.

A *General* panel displays on the right.

2 Go to the *Security* panel.

This panel displays the current security permissions for the selected portlet registration:



3 Go to the *List* or *Execute* tab, depending on which type of permission you want to assign.

4 Specify values for the following *search settings*:

Setting	What to do
Search for	Select one of the following from the dropdown menu: • Users • Groups • Containers

Setting	What to do

Starts with	If you want to:
	• **Find all** available objects of your specified type (user, group, or container), then make this setting blank.
	• **Find a subset** of those objects, then enter the starting character(s) of the CN values you want. (Case is not considered. Wildcards are not supported.)
	For example, searching for groups that start with S would narrow your search results to something like this:
	`cn=Sales,ou=groups,o=MyOrg`
	`cn=Service,ou=groups,o=MyOrg`
	`cn=Shipping,ou=groups,o=MyOrg`
	Searching for groups that start with Se would return:
	`cn=Service,ou=groups,o=MyOrg`

5 Click *Go*.

The results of your search appear in the *Results* list.

6 *Select* the users, groups, or containers you want to assign to the portlet registration, then click the *Add* (>) button.

TIP: Hold down the *Control* key to make multiple selections.

7 Enable or disable *lock-down* for the portlet registration as follows:

If you want to	Do this
Lock down the portlet registration so only User Application Administrators can list/execute it	Check **List/Execute Permission Set to Admin Only**
Allow all assigned users, groups, and containers to list/execute the portlet registration	Uncheck **List/Execute Permission Set to Admin Only** **NOTE:** If you uncheck this setting but there are no users, groups, or containers explicitly assigned to the portlet registration, then **everyone will have List/Execute permission** for this portlet registration.

8 Click *Save*.

9.4.7 Unregistering a portlet

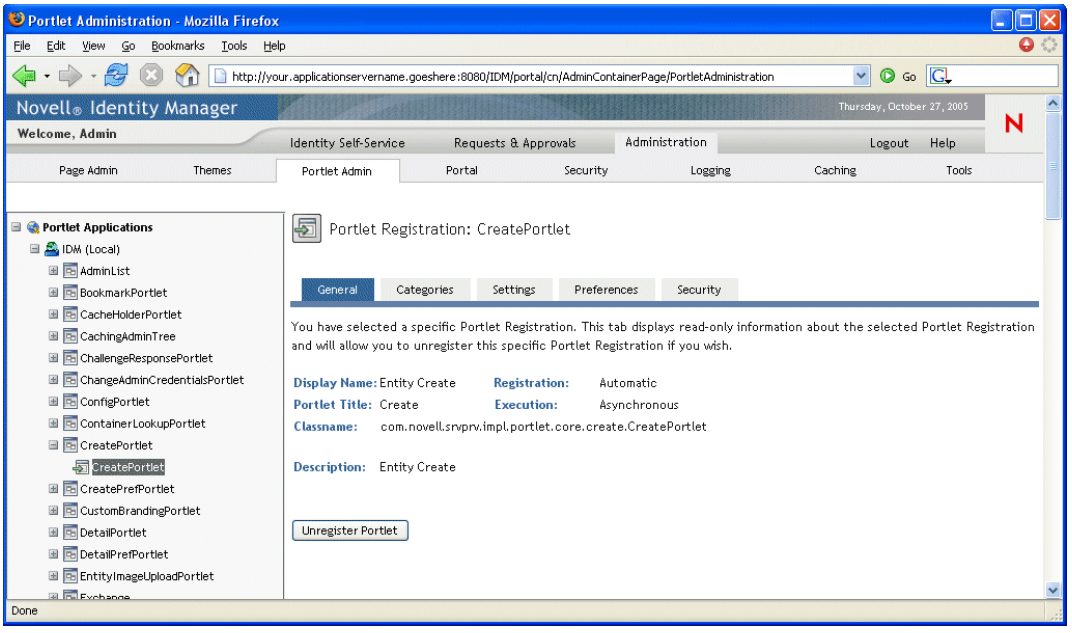
You can use the Portlet Admin page to unregister a portlet if necessary.

NOTE: If you unregister a portlet that is defined as *auto-registered*, that portlet will be registered again automatically when you restart your application server.

To unregister a portlet:

1 In the Portlet Applications list, *select* the portlet registration that you want to unregister.

A *General* panel displays on the right, showing information about the selected portlet registration:



2 Click *Unregister Portlet*.

3 When you are prompted to confirm the unregister operation, click *OK*.

Portal Configuration

10

This chapter tells you how to use the *Portal* page on the *Administration tab* of the Identity Manager user interface. Topics include:

For more general information about accessing and working with the Administration tab, see Chapter 6, “Using the Administration Tab,” on page 125.

10.1 About portal configuration

You can use the Portal page to control the *portal characteristics* of the Identity Manager user application and specify how the user application connects to the *identity vault* (LDAP provider).

10.2 General settings

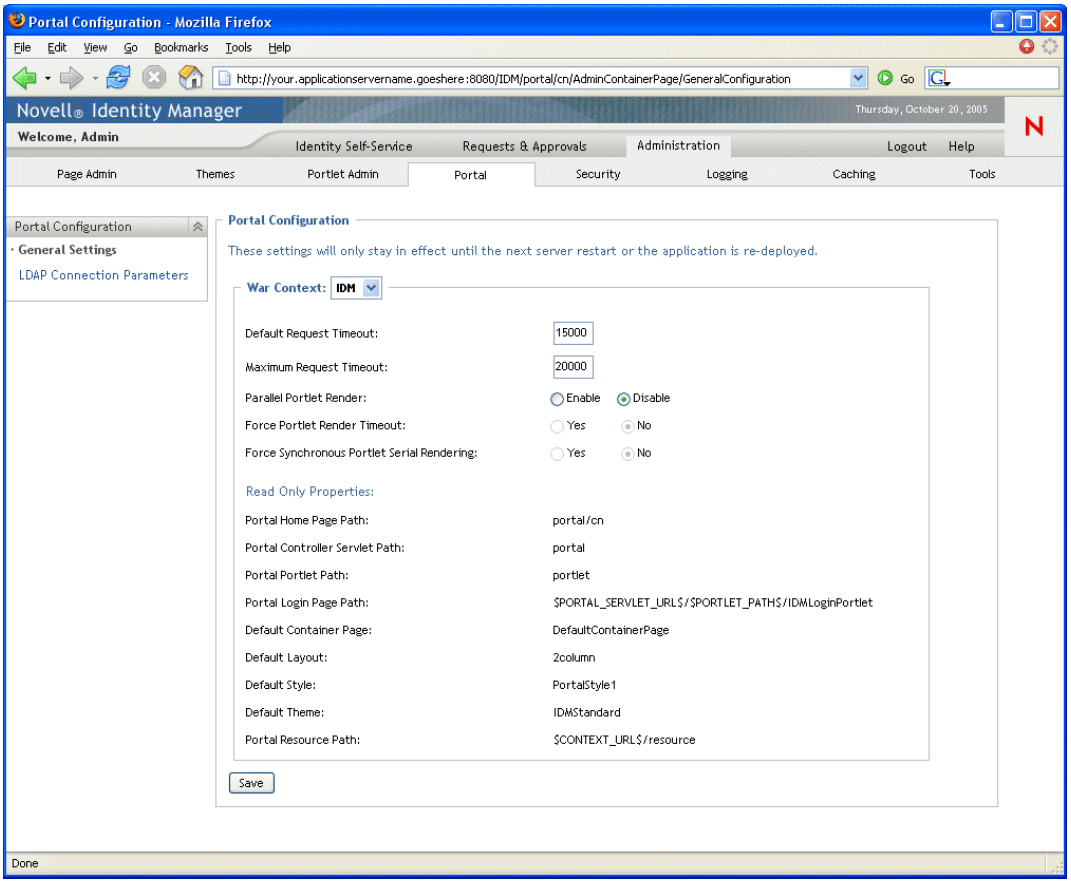
The Portal page provides a General Settings panel that you can use to:

- *Change some portal characteristics* of the Identity Manager user application temporarily (until the next application-server restart or user-application redeployment)
- *View other portal characteristics* of the Identity Manager user application

To administer general settings:

1. On the Portal page, select *General Settings* from the navigation menu on the left.

   The General Settings panel displays:



2 If you have more than one *War Context*, select the one whose settings you want to access.

   The panel refreshes to show the current settings for your chosen context.

3 *Examine* and *modify* the settings, as appropriate. For details, see:

   - Section 10.2.1, “Settings you can change,” on page 190
   - Section 10.2.2, “Read-only settings,” on page 192

4 If you make changes that you want to apply, click *Save*.

10.2.1 Settings you can change

You can modify several portal settings on the General Settings panel. Your values stay in effect until the next application-server restart or user-application redeployment. When a restart or redeployment occurs, these settings revert to the default values for the user application WAR.

Setting	What to do
Default Request Timeout	Specify the default time (in milliseconds) that a request will wait before it times out. If none of the asynchronous portlets defines a timeout, or none of the portlets defines a timeout that is bigger than this value, this default value will be used. If one or more of the portlets to render defines a timeout that is bigger than this default value, the bigger one will be used instead of the default. This setting can be used to protect the application from getting too many messages indicating that portlets have timed out (which might happen if the portlets define values that are too small). **NOTE:** In the event that all portlets can be rendered before this default timeout occurs, the request will immediately return to the client.
Maximum Request Timeout	Specify the maximum time (in milliseconds) that a request will be held back from finishing. This means that after this amount of time, every request will return to the client, regardless of whether any portlet defines a bigger timeout value. This setting can be used to make sure that the portal responds in a timely fashion even if one or more of the portlets define a large timeout value.
Parallel Portlet Render	Enable or disable asynchronous portlet rendering in the portal. This is an advanced feature that is disabled by default. If you enable this feature, the portal assigns asynchronous render requests to individual threads (which allows portlets to render content in parallel). When this feature is disabled, all portlets render content synchronously in the main request thread.
Force Portlet Render Timeout	Determine whether asynchronous portlets are delegated to the main request thread to render content if there are not enough individual threads available in the thread pool. If you choose **No**, asynchronous portlets can execute in the main request thread if no individual threads are available. Choosing **Yes** forces asynchronous portlets to wait until individual threads are available before they can render content. If portlets time out before they execute the render request, a portlet-specific error message is generated in the portlet window.
Force Synchronous Portlet Serial Rendering	Determine how synchronous portlets are executed. If you choose **Yes**, all synchronous portlets execute in the main request thread. Choosing **No** enables the portal to allocate a separate thread for processing synchronous render requests (thereby preventing bottlenecks in the main request thread).

10.2.2 Read-only settings

The following settings are displayed for informational purposes only and cannot be changed on the General Settings panel:

Portal Home Page Path	Default Layout
Portal Controller Servlet Path	Default Style
Portal Portlet Path	Default Theme
Portal Login Page Path	Portal Resource Path
Default Container Page	

The values of these settings are set in the user application WAR. (Note that Default Theme reflects your current theme choice from the Themes page.)

10.3 LDAP connection parameters

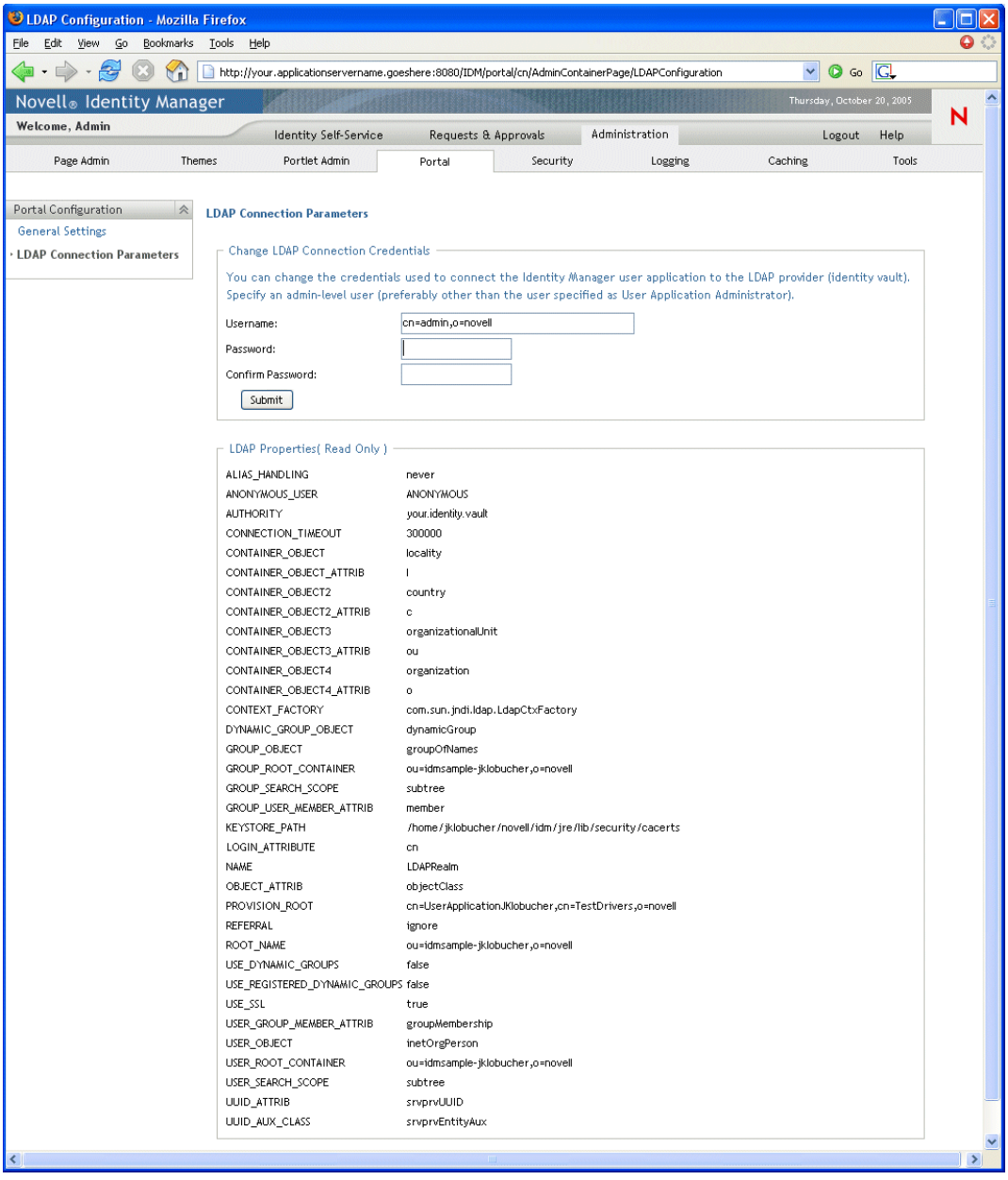
The Portal page provides an LDAP Connection Parameters panel that you can use to:

- *Change the credentials* used by the Identity Manager user application when connecting to the identity vault (LDAP provider)
- *View other LDAP properties* of the Identity Manager user application

To administer LDAP connection parameters:

1. On the Portal page, select *LDAP Connection Parameters* from the navigation menu on the left.

   The LDAP Connection Parameters panel displays:



2 *Examine* and *modify* the settings, as appropriate. For details, see:

- Section 10.2.1, “Settings you can change,” on page 190
- Section 10.3.2, “Read-only settings,” on page 194

3 If you make changes that you want to apply, click *Submit*.

10.3.1 Settings you can change

On the LDAP Connection Parameters panel, you can modify settings for the credentials to be used by the Identity Manager user application whenever it connects to the identity vault (LDAP provider). Your changes on this panel are saved to the user application’s database for use at runtime and checked against the identity vault. (Note that this panel does not update the original credential values recorded in the user application WAR during installation.)

Setting	What to do
Username	Type the name of a user who has full **administrator** rights in the identity vault. The Identity Manager user application needs to access the identity vault as an administrator in order to function. It is typical to specify the identity vault's **root** administrator as the LDAP connection username. The root administrator has full control over the tree, so you need not assign any special trustee rights. For example: `cn=admin,o=myorg` If you specify some other user, you'll need to assign inheritable trustee rights to the properties [All Attributes Rights] and [Entry Rights] on your user application driver. **NOTE:** To avoid confusion, it is recommended that you do **not** specify the user application's User Application Administrator as the LDAP connection username. It is best to use separate accounts for these two different purposes.
Password and Confirm Password	Type the password that is currently set for that username in the identity vault.

10.3.2 Read-only settings

The following settings are displayed for informational purposes only and cannot be changed on the LDAP Connection Parameters panel:

ALIAS_HANDLING	GROUP_USER_MEMBER_ATTRIB
ANONYMOUS_USER	KEYSTORE_PATH
AUTHORITY	LOGIN_ATTRIBUTE
CONNECTION_TIMEOUT	NAME
CONTAINER_OBJECT	OBJECT_ATTRIB
CONTAINER_OBJECT_ATTRIB	PROVISION_ROOT
CONTAINER_OBJECT2	REFERRAL
CONTAINER_OBJECT2_ATTRIB	ROOT_NAME
CONTAINER_OBJECT3	USE_DYNAMIC_GROUPS
CONTAINER_OBJECT3_ATTRIB	USE_REGISTERED_DYNAMIC_GROUPS
CONTAINER_OBJECT4	USE_SSL
CONTAINER_OBJECT4_ATTRIB	USER_GROUP_MEMBER_ATTRIB

CONTEXT_FACTORY	USER_OBJECT
DYNAMIC_GROUP_OBJECT	USER_ROOT_CONTAINER
GROUP_OBJECT	USER_SEARCH_SCOPE
GROUP_ROOT_CONTAINER	UUID_ATTRIB
GROUP_SEARCH_SCOPE	UUID_AUX_CLASS

The values of these settings are determined when you install the user application.

Security Configuration

11

This chapter tells you how to use the *Security* page on the *Administration tab* of the Identity Manager user interface. Topics include:

For more general information about accessing and working with the Administration tab, see Chapter 6, “Using the Administration Tab,” on page 125.

11.1 About security configuration

You can use the Security page to specify who is a *User Application Administrator* for the Identity Manager user application.

A User Application Administrator is authorized to perform all management functions related to the Identity Manager user application. This includes accessing the Administration tab of the Identity Manager user interface to perform any administration actions that it supports.

During installation, a user is specified as User Application Administrator. After installation, that user can use the Security page to specify other User Application Administrators, as needed.

A user who is to be User Application Administrator should typically be *located under the user root container* specified in the user application’s LDAP configuration; that enables the user to log in simply by user name (instead of requiring the fully-distinguished name each time). It is also common that this user has *rights to maintain and create objects* in the tree; however, this is not required.

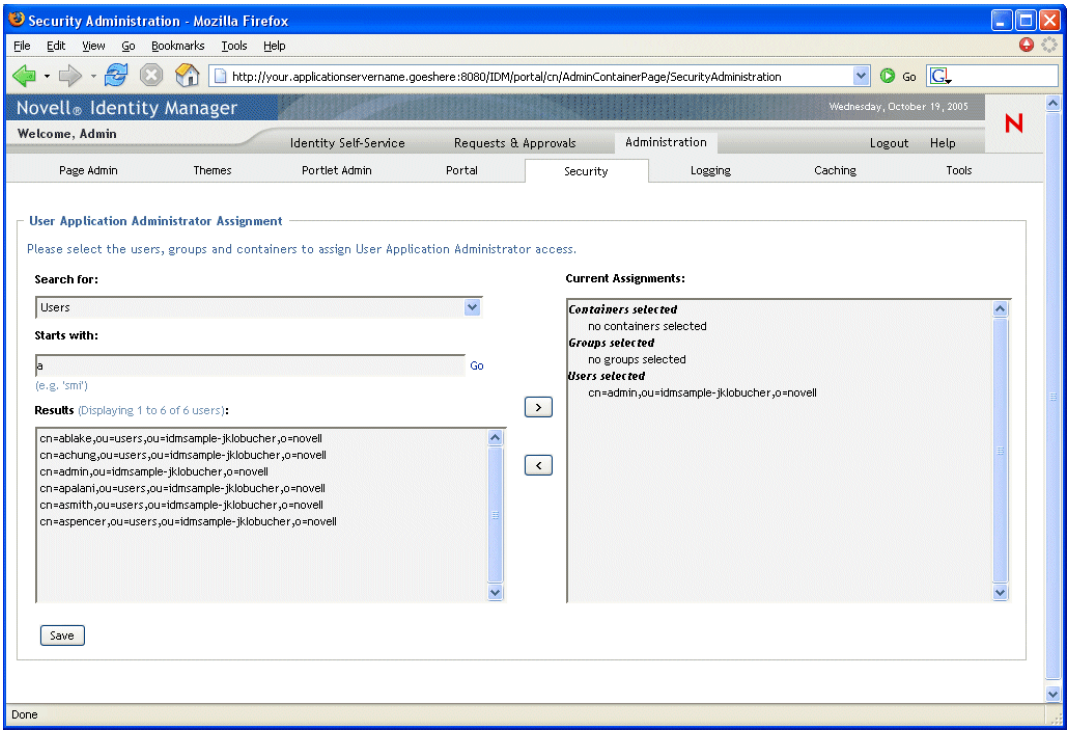
NOTE: If necessary, a User Application Administrator can assign permission for one or more end users to see and access specific pages on the Administration tab. These permissions are assigned by using the *Page Admin* page on the Administration tab. (For details, see Chapter 7, “Page Administration,” on page 131.)

11.2 Assigning the User Application Administrator

When assigning User Application Administrators, you can specify users, groups, or containers.

To assign User Application Administrators:

1 Go to the *Security* page:



2 Specify values for the following *search settings*:

Setting	What to do
Search for	Select one of the following from the dropdown menu: • Users • Groups • Containers

Setting	What to do

Starts with	If you want to:
	• **Find all** available objects of your specified type (user, group, or container), then make this setting blank.
	• **Find a subset** of those objects, then enter the starting character(s) of the CN values you want. (Case is not considered. Wildcards are not supported.)
	For example, searching for groups that start with S would narrow your search results to something like this:
	`cn=Sales,ou=groups,o=MyOrg`
	`cn=Service,ou=groups,o=MyOrg`
	`cn=Shipping,ou=groups,o=MyOrg`
	Searching for groups that start with Se would return:
	`cn=Service,ou=groups,o=MyOrg`

3 Click *Go*.

The results of your search appear in the *Results* list.

4 *Select* the users, groups, or containers you want to assign as User Application Administrators, then click the *Add* (>) button.

TIP: Hold down the *Control* key to make multiple selections.

5 Click *Save*.

To unassign User Application Administrators:

1 In the *Current Assignments* list, select the users, groups, or containers you want to unassign as User Application Administrators, then click the *Remove* (<) button.

TIP: Hold down the *Control* key to make multiple selections.

2 Click *Save*.

Logging Configuration

12

This chapter tells you how to use the *Logging* page on the *Administration tab* of the Identity Manager user interface. Topics include:

For more general information about accessing and working with the Administration tab, see Chapter 6, “Using the Administration Tab,” on page 125.

12.1 About logging configuration

You can use the Logging page to control the *levels of logging messages* you want the Identity Manager user application to generate and specify whether those messages are sent to *Novell Audit*.

The Identity Manager user application implements logging by using *log4j*, an open-source logging package distributed by The Apache Software Foundation. By default, event messages are logged to both of the following:

- *The system console* of the application server where the Identity Manager user application is deployed
- *A log file* on that application server, for example:

  `jboss/server/IDM/log/server.log`

This is a rolling log file; once it reaches a certain size, it rolls over to another file (and so on).

If you’ve configured your environment to include Novell Audit, you have the option of logging event messages there as well.

For details on configuring your logging environment and Novell Audit, see Chapter 5, “Setting up Logging,” on page 115.

12.2 About the logs

The Logging page lists a variety of logs, each outputting event messages from a different part of the Identity Manager user application. Each log has its own independent output level.

The log names are based on log4j conventions. You’ll see these log names in the event messages that are generated, indicating the context of the message output.

Log name	Description
com.novell	Parent of other Identity Manager user application logs
com.novell.afw.portal.aggregation	Messages related to portal page processing

Log name	Description
com.novell.afw.portal.persist	Messages related to the persistence of portal data (including portal pages and portlet registrations)
com.novell.afw.portal.portlet	Messages from the portal core portlets and accessory portlets
com.novell.afw.portal.util	Messages from the portal import/export and navigation portlets
com.novell.afw.portlet.consumer	Messages related to portlet rendering
com.novell.afw.portlet.core	Messages related to the core portlet API
com.novell.afw.portlet.persist	Messages related to the persistence of portlet data (including portlet preferences and setting values)
com.novell.afw.portlet.producer	Messages related to the registration and configuration of portlets within the portal
com.novell.afw.portlet.util	Messages related to utility code used by portlets
com.novell.afw.theme	Messages from the theme subsystem
com.novell.afw.util	Messages related to portal utility classes
com.novell.soa.af.impl	Messages from the approval flow (provisioning workflow) subsystem
com.novell.srvprv.apwa	Messages from the Requests & Approvals web application (actions and tags)
com.novell.srvprv.impl.portlet.core	Messages from the core identity portlets and password portlets
com.novell.srvprv.impl.portlet.util	Messages from the identity-related utility portlets
com.novell.srvprv.impl.servlet	Messages from the UI control framework's ajax servlet and ajax services
com.novell.srvprv.impl.uictrl	Messages from the UI control registry API and approval form rendering
com.novell.srvprv.impl.vdata	Messages from the directory abstraction layer
com.novell.srvprv.spi	Messages from the UI control registry API
com.sssw.fw.cachemgr	Messages related to the framework cache subsystem
com.sssw.fw.core	Messages related to the framework core subsystem
com.sssw.fw.directory	Messages related to the framework directory subsystem
com.sssw.fw.event	Messages related to the framework event subsystem
com.sssw.fw.factory	Messages related to the framework factory subsystem
com.sssw.fw.persist	Messages related to the framework persistence subsystem
com.sssw.fw.resource	Messages related to the framework resource subsystem
com.sssw.fw.security	Messages related to the framework security subsystem
com.sssw.fw.server	Messages related to the framework server subsystem

Log name	Description
com.sssw.fw.servlet	Messages related to the framework servlet subsystem
com.sssw.fw.session	Messages related to the framework session subsystem
com.sssw.fw.usermgr	Messages related to the framework user subsystem
com.sssw.fw.util	Messages related to the framework utility subsystem
com.sssw.portal.manager	Messages related to the Portal Manager
com.sssw.portal.persist	Messages related to portal persistence

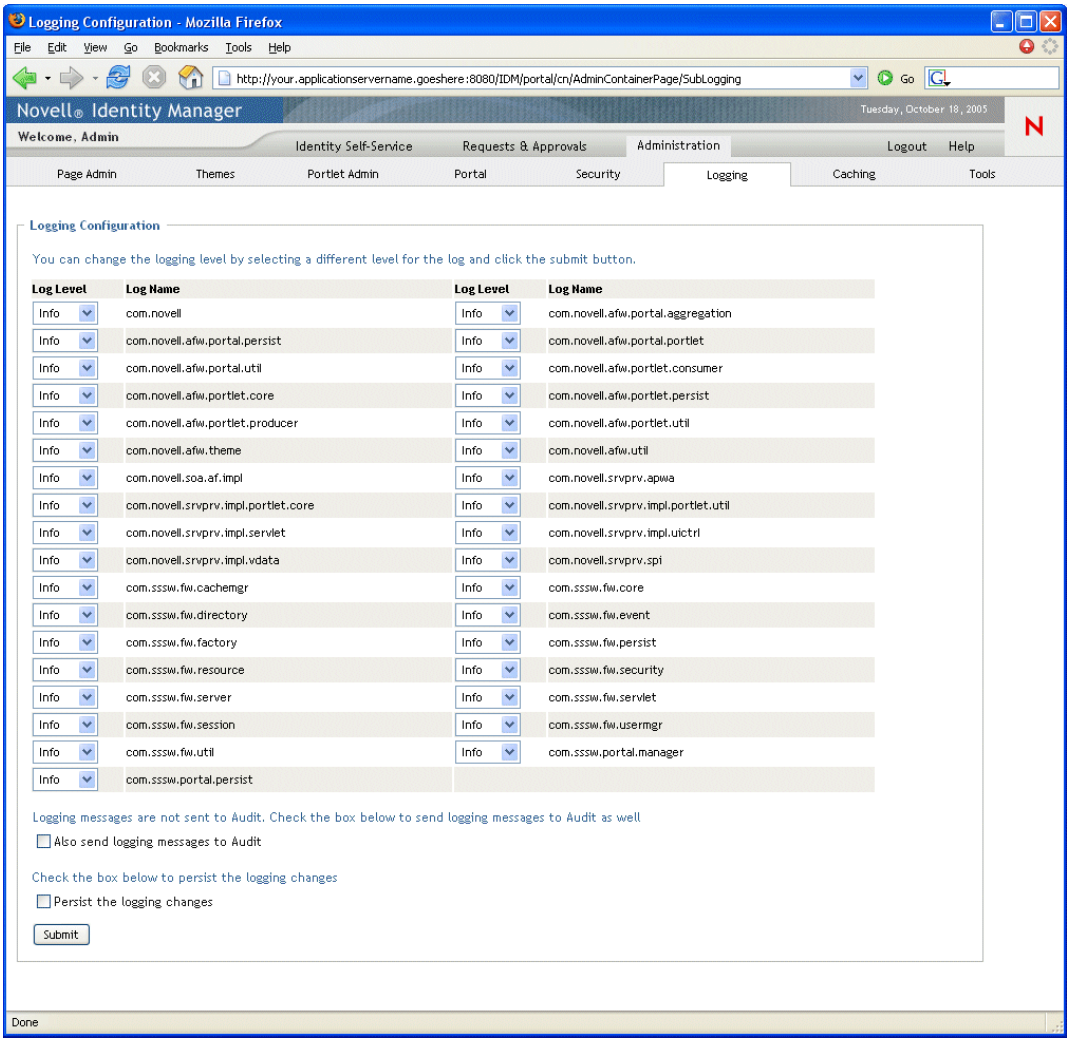
Note that the user application's logs are hierarchical. For example, com.novell is the parent of other logs underneath it. Any additional logs will inherit its properties.

12.3 Changing log levels

You can control the amount of information that is written to a particular log by changing the level that is set for it. By default, all logs are set to *Info*, which is an intermediate level.

To change log levels:

1 Go to the *Logging* page:



2 At the top of the page, *find a log* whose level you want to change.

3 Use the dropdown to *select* one of the following levels:

Level	Description
Fatal	**The least detail:**
	Writes fatal errors to the log
Error	Writes errors (plus all of the above) to the log
Warn	Writes warnings (plus all of the above) to the log
Info	Writes informational messages (plus all of the above) to the log
Debug	Writes debugging information (plus all of the above) to the log
Trace	**The most detail:**
	Writes tracing information (plus all of the above) to the log

4 *Repeat* Step 2 and Step 3 for other logs, as needed.

5 Click *Submit*.

12.4 Sending log messages to Novell Audit

You can use the Logging page to control whether the Identity Manager user application sends event message output to Novell Audit. Novell Audit logging is off by default, unless you turn it on when installing the user application.

To toggle Novell Audit logging on/off:

1 Go to the *Logging* page.

2 *Check or uncheck* the following setting, as appropriate:

`Also send logging messages to Audit`

3 Click *Submit*.

12.5 Persisting your log settings

By default, changes you make on the Logging page stay in effect until the next application-server restart or user-application redeployment. After that, the log settings revert to their default values.

But the Logging page does offer you the option of persisting your changes to its settings. If you turn on this feature, values for the log settings are stored in a *logging configuration file* on the application server where the Identity Manager user application is deployed. For example:

`jboss/server/IDM/conf/extendlogging.xml`

To toggle persistence of settings on/off:

1 Go to the *Logging* page.

2 *Check or uncheck* the following setting, as appropriate:

`Persist the logging changes`

3 Click *Submit*.

Caching Configuration

13

This chapter tells you how to use the *Caching* page on the *Administration tab* of the Identity Manager user interface. Topics include:

For more general information about accessing and working with the Administration tab, see Chapter 6, “Using the Administration Tab,” on page 125.

13.1 About caching configuration

You can use the Caching page to manage various *caches* maintained by the Identity Manager user application. The user application employs these caches to store reusable, temporary data on the application server so it can optimize performance.

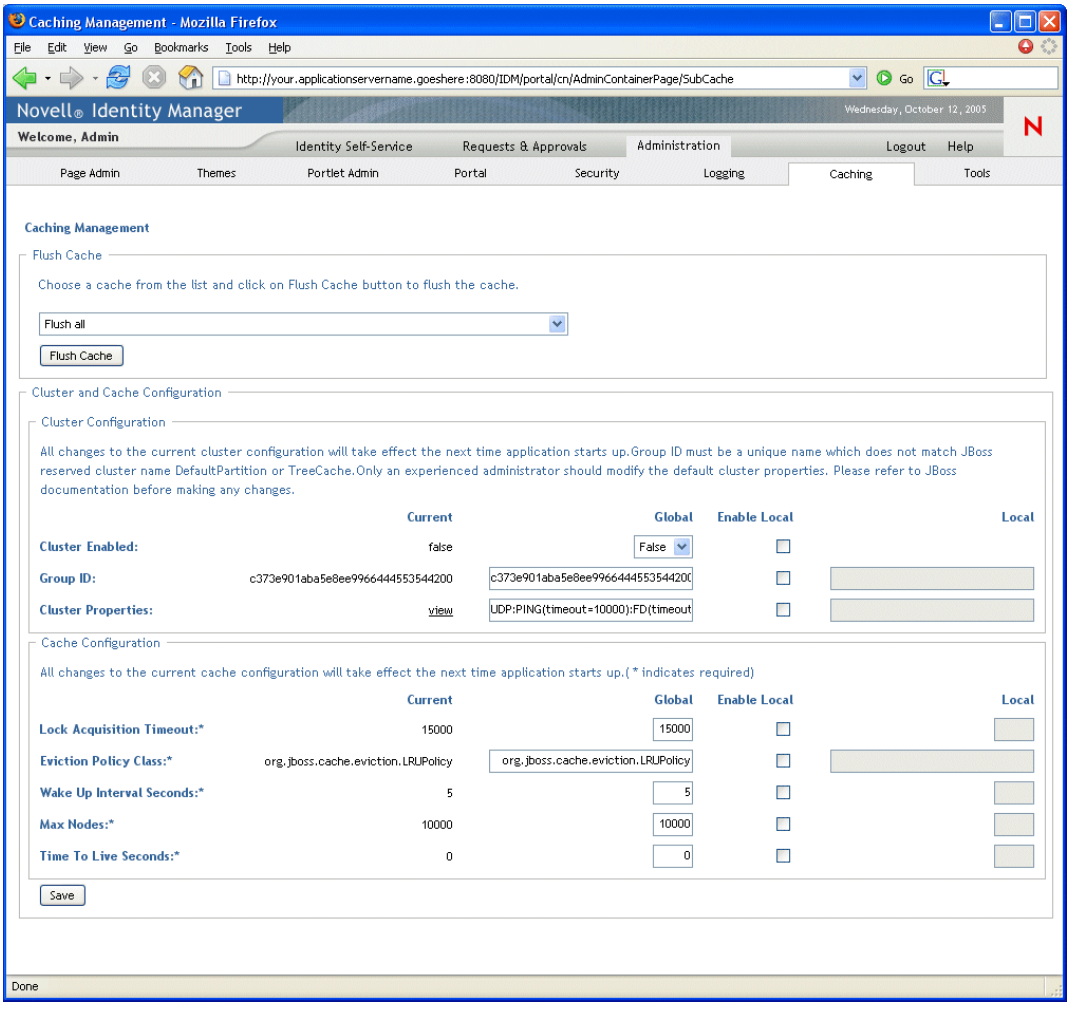
You have the ability to control these caches when necessary by *flushing their contents* and *changing their configuration settings*.

13.2 Flushing caches

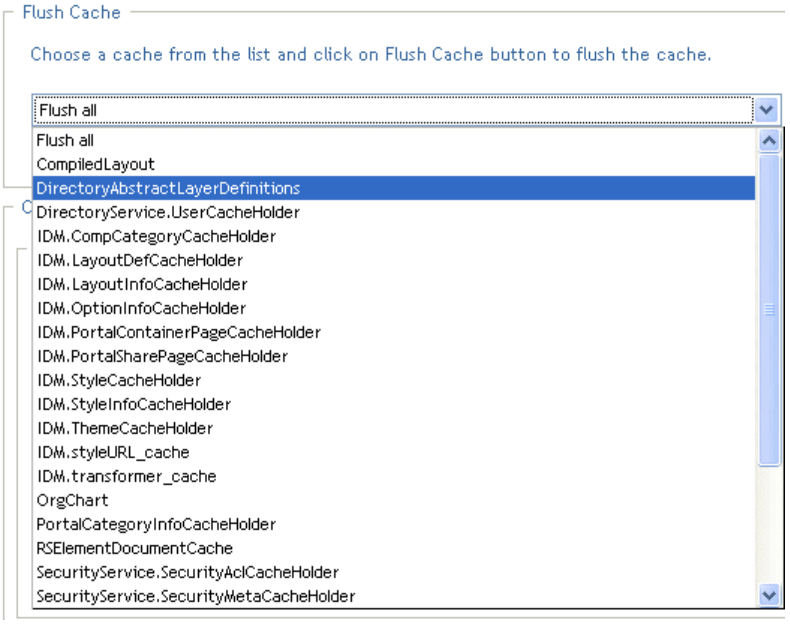
The caches are named according to the *subsystems* that use them in the Identity Manager user application. Normally, you don’t need to flush them yourself, because the user application does that automatically based on how frequently their data is used or when the source data changes. But if you have a specific need, you can *manually flush* selected caches or all caches.

To flush caches:

1 Go to the *Caching* page:



2 In the *Flush Cache* section of the page, use the dropdown to *select* a particular cache to flush (or select *Flush all*):



Note that the list of available caches is *dynamic*; it changes depending on what data is cached at the moment.

3 Click the *Flush Cache* button.

13.2.1 Flushing the directory abstraction layer cache

The user application's *directory abstraction layer* also has a cache. The *DirectoryAbstractLayerDefinitions* cache stores abstraction layer definitions on the application server to optimize performance for all data model operations.

In a typical situation, the user application automatically keeps the DirectoryAbstractLayerDefinitions cache synchronized with the abstraction layer definitions stored in the identity vault. But, if necessary, you can manually flush the DirectoryAbstractLayerDefinitions cache (as described above) to force the latest definitions to be loaded from the identity vault.

For more information on the user application's directory abstraction layer, see Chapter 4, "Configuring the Directory Abstraction Layer," on page 75.

13.2.2 Flushing caches in a cluster

Cache flushing is supported in both clustered and non-clustered application server environments. If your application server is part of a cluster and you manually flush a cache, that cache will automatically be *flushed on every server* in the cluster.

13.3 Configuring cache settings

You can use the Caching page to display and change cache configuration settings for a *clustered or non-clustered* application server environment. Your changes are saved immediately, but they don't take effect until the next *user-application restart*.

TIP: To restart the user application, you can do one of the following: reboot the application server; redeploy the application (if the WAR has been changed in some way); or force the application to restart (as described in your application server's documentation).

To configure cache settings, you need to know about:

13.3.1 How caching is implemented

In the Identity Manager user application, caching is implemented via *JBoss Cache*. JBoss Cache is an open-source caching architecture that's included with the JBoss Application Server but also runs on other application servers.

To learn more about JBoss Cache, go to www.jboss.org/products/jbosscache (http://www.jboss.org/products/jbosscache).

13.3.2 How cache settings are stored

There are *two levels of settings* available for you to control cache configuration. You can use them in concert to tailor the caching behavior of the Identity Manager user application.

Level	Description

Global settings	Global settings are **stored in a central location** (the identity vault) so that multiple application servers can use the same setting values. For example, someone with a cluster of application servers would typically use global settings for the cluster configuration values.
	
	To **find the global settings** in your identity vault, look for the following object under your Identity Manager user application driver:
	
	`configuration.AppDefs.AppConfig`
	
	For example:
	
	`configuration.AppDefs.AppConfig.MyUserApplicationDriver.MyDriverSet.MyOrg`
	
	The **XmlData attribute** of the configuration object contains the global settings data.

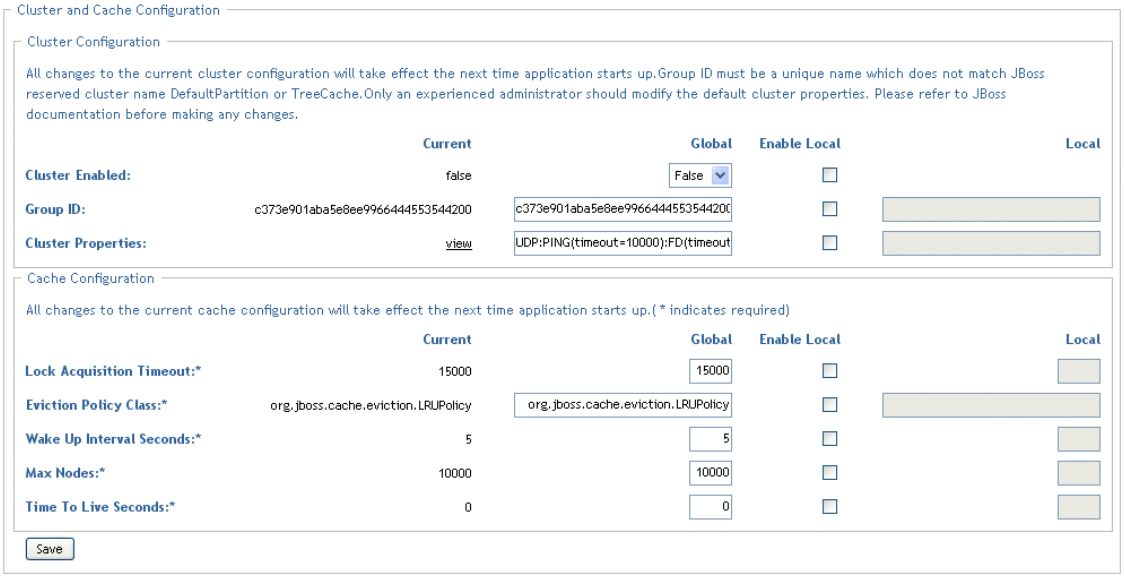
Level	Description

Local settings	Local settings are **stored separately on each application server** so that an individual server can **override** the value of one or more global settings. For example, you might want to specify a local setting to remove an application server from the cluster specified in the global settings, or maybe to reassign a server to a different cluster.
	
	To **find the local settings** on your application server, look for the following file under your JBoss server configuration's conf directory:
	
	`sys-configuration-xmldata.xml`
	
	For example:
	
	`jboss/server/IDM/conf/sys-configuration-xmldata.xml`
	
	If your server has local settings, that data is contained in this file. (If no local settings have been specified, the file won't exist.)

You should think of global settings as the *default values* for every application server that uses a particular instance of the user application driver. When you change a global setting, you are *affecting each of those servers* (at the next user-application restart), except for those cases where an individual server specifies a local override.

13.3.3 How cache settings are displayed

The Caching page displays the *current cache settings* (from the latest user-application restart). It also displays the corresponding *global and local values* of those settings, and lets you *change* them (for use at the next user-application restart).



Note that the global settings always have values. The local settings are optional.

13.3.4 Basic cache settings

These cache settings apply to both clustered and non-clustered application servers.

To configure basic cache settings:

1. Go to the *Caching* page.
2. In the *Cache Configuration* section of the page, specify *global or local values* for the following settings, as appropriate:

Setting	What to do
Lock Acquisition Timeout	Specify the **time interval (in milliseconds)** that the cache waits for a lock to be acquired on an object. You may want to increase this setting if the user application gets a lot of lock timeout exceptions in the application log. The default is 15000 ms.
Eviction Policy Class	Specify the **classname** for the cache eviction policy that you want to use. The default is the LRU eviction policy that JBoss Cache provides: `org.jboss.cache.eviction.LRUPolicy` If appropriate, you can change this to another eviction policy that JBoss Cache supports. To learn about supported eviction policies, go to www.jboss.org/products/jbosscache (http://www.jboss.org/products/jbosscache).
Wake Up Interval Seconds	Specify the **time interval (in seconds)** that the cache eviction policy waits before waking up to do the following: • Process the evicted node events • Clean up the size limit and age-out nodes
Max Nodes	Specify the **maximum number of nodes** allowed in the cache. For no limit, specify: `0`
Time To Live Seconds	Specify the **time to idle (in seconds)** before the node is swept away. For no limit, specify: `0`

These settings are *required*, which means that there must be a global value for each, and optionally a local value too.

If you want to *override* the global value of a setting with a local value, select the *Enable Local* check box for that setting. Then specify the local value. (Make sure that all of your local values are *valid*. Otherwise, you won't be able to save your changes.)

NOTE: For those settings where Enable Local is unselected, any existing local values are deleted when you save.

3 Click *Save*.

4 When you're ready for your saved settings to take effect, *restart the user application* on the applicable application server(s).

13.3.5 Cache settings for clusters

This section discusses how to configure caching when you run the Identity Manager user application across a cluster of application servers. You need to know about:

How clustering is implemented

In the Identity Manager user application, cluster support for caching is implemented via *JGroups*. JGroups is an open-source clustering architecture that's included with the JBoss Application Server but also runs on other application servers.

The *user application's cluster* consists of nodes on a network that run JGroups and use a common *Group ID*. By default, the Group ID provided for the user application's cluster is a UUID that looks like this:

```
c373e901aba5e8ee9966444553544200
```

The UUID helps ensure uniqueness, so that the Group ID of the user application's cluster doesn't conflict with the Group IDs of other clusters in your environment. For instance, the JBoss Application Server itself uses two JGroups clusters and *reserves the Group IDs DefaultPartition and TreeCache* for them.

To learn more about JGroups, go to www.jboss.org/products/jgroups (http://www.jboss.org/products/jgroups).

How caching works with a cluster

When you start the user application, the application's cache configuration settings determine whether to participate in a cluster and replicate cache changes to the other nodes in that cluster. If clustering is enabled, the user application accomplishes this replication by sending *cache entry invalidation messages* to each node as changes occur.

Preparing to use a cluster

There are two major steps required to use caching across a cluster:

1 *Setting up your JGroups cluster*

This involves installing the JBoss Application Server to use the *all* configuration, and then distributing the Identity Manager user application (IDM.war) to every server in the cluster, typically by putting it in the *farm* directory.

2 *Enabling the use of that cluster* in the user application's cache configuration settings

See "Configuring cache settings for clusters" on page 214 (below).

Configuring cache settings for clusters

Once you have a cluster ready to use, you can specify settings for the support of caching across that cluster.

To configure cache settings for clusters:

1 Go to the *Caching* page.

2 In the *Cluster Configuration* section of the page, specify *global or local values* for the following settings, as appropriate:

Setting	What to do
Cluster Enabled	Select **True** to replicate cache changes to the other nodes in the cluster specified by Group ID. If you don't want to participate in a cluster, select **False**.
Group ID	Specify the Group ID of the JGroups cluster in which you want to participate. There's **no need to change the default** Group ID that's provided for the user application's cluster, **unless you want to use a different cluster**. Remember that the following Group IDs are reserved for use by the JBoss Application Server: DefaultPartition and TreeCache. **TIP:** To see the Group ID in logging messages, make sure that the level of the caching log (com.sssw.fw.cachemgr) is set to Info or higher.
Cluster Properties	Specify the JGroups **protocol stack** for the cluster specified by Group ID. Note that this setting is **for experienced administrators** who may need to adjust the cluster properties. Otherwise, you should not change the default protocol stack. To see the current cluster properties, click **view**. For details on the JGroups protocol stack, go to www.joss.org/wiki/Wiki.jsp?page=JGroups (http://www.jboss.org/wiki/Wiki.jsp?page=JGroups).

If you want to *override* the global value of a setting with a local value, select the *Enable Local* check box for that setting. Then specify the local value.

NOTE: For those settings where Enable Local is unselected, any existing local values are deleted when you save.

Make sure that *all nodes* in your cluster *specify the same* Group ID and Cluster Properties. (To see these settings for a particular node, you must access the Identity Manager user interface running on that node — by browsing to the URL of the user interface on that server — and then display the Caching page there.)

3 Click *Save*.

4 When you're ready for your saved settings to take effect, *restart the user application* on the applicable application server(s).

Tools for Exporting and Importing Portal Data

14

This chapter tells you how to use the *Tools* page on the *Administration tab* of the Identity Manager user interface. Topics include:

For more general information about accessing and working with the Administration tab, see Chapter 6, “Using the Administration Tab,” on page 125.

14.1 About exporting and importing portal data

You can use the Tools page to *export or import* portal content (pages and portlets) used in the Identity Manager user application. This content is also known as the *portal configuration state* and it includes:

- Container and shared pages (including each page’s assigned portlets, and each portlet’s preferences and settings)
- Portlet registrations

The export and import tools enable you to move the portal configuration state from one portal (user application) to another, as needed. Here’s how these tools work:

Tool	How it works
Portal Data Export	Generates XML descriptions of a set of selected container and shared pages, and portlets. The XML files are stored in a **Portal Data Export ZIP file** that can be used as input to the Portal Data Import tool.
Portal Data Import	Accepts a Portal Data Export ZIP file as input. Uses the Portal Data Export ZIP file to generate container and shared pages, and portlets in a portal (user application).

14.1.1 Uses

You can use the Portal Data Export/Import tools to:

- *Move* your portal configuration state from a test (source) environment to a production (target) environment
- *Update* the configuration state of a portal incrementally
- *Clone* a portal
- Optionally, *overwrite* the configuration state on the target portal

14.1.2 Requirements

To use the Portal Data Export/Import tools, make sure that the Identity Manager user application (portal) is *deployed and running* on your source and target application servers.

It is *not required* that your source and target servers access the same *identity vault*; they can access different ones, if appropriate. The *users, groups, and containers* in those identity vaults are *not required* to be the same.

14.1.3 Restrictions

You *cannot* use the Portal Data Export/Import tools to:

- Export or import portal configuration state when a server is currently servicing user requests
- Export or import portal classes and resources
- Export or import portlet classes and resources
- Export or import the identity and provisioning data used in a portal
- Export or import administration settings other than for pages and portlets
- Migrate configuration state from an earlier portal version to a later version (the portals must be the same version)

14.1.4 Steps

To export and import portal data:

1. If you are performing an incremental update, *back up* the target portal.
2. From the source portal, *export* the portal data by using the Portal Data Export tool.

   See Section 14.2, “Exporting portal data,” on page 218.
3. From the target portal, *import* the portal data by using the Portal Data Import tool.

   See Section 14.3, “Importing portal data,” on page 220.
4. *Test* the target portal to ensure that you imported the data that you expected.

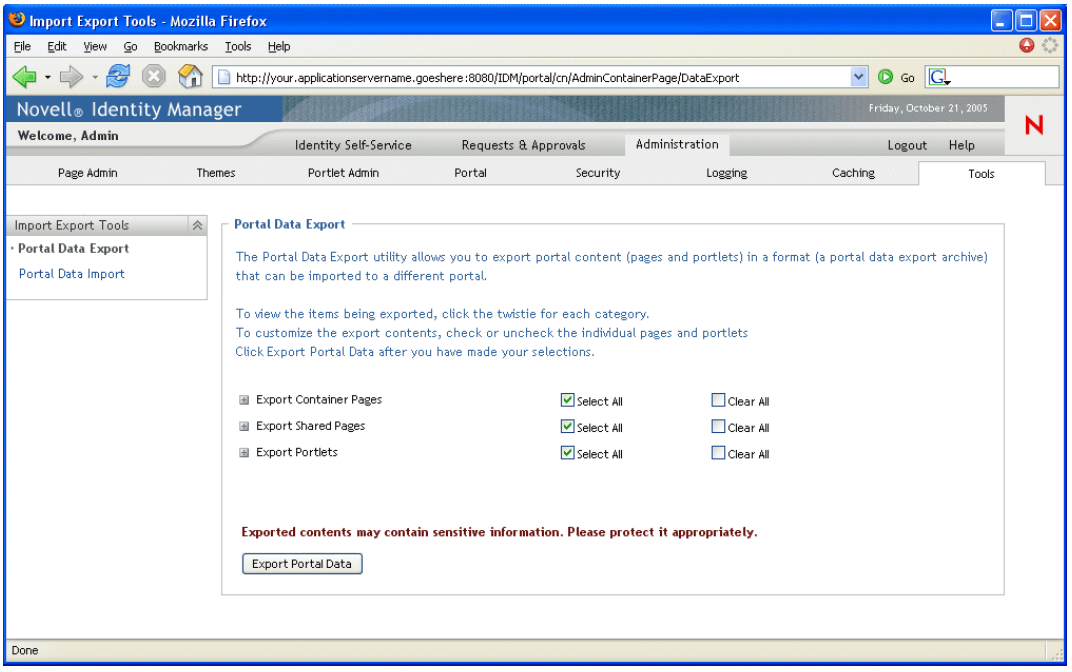
14.2 Exporting portal data

This section describes how to export a portal’s configuration state to a Portal Data Export ZIP file.

To export portal data:

1. On the Tools page, select *Portal Data Export* from the navigation menu on the left.

   The Portal Data Export panel displays:



2 Follow the onscreen instructions to *select the portal pages and portlets* that you want to export.

NOTE: Some portlets that you have not selected for export might still be exported. If you export a page that contains a portlet, but do not select that portlet for export, the portlet is still exported (to ensure that a runtime error does not occur for the exported page).

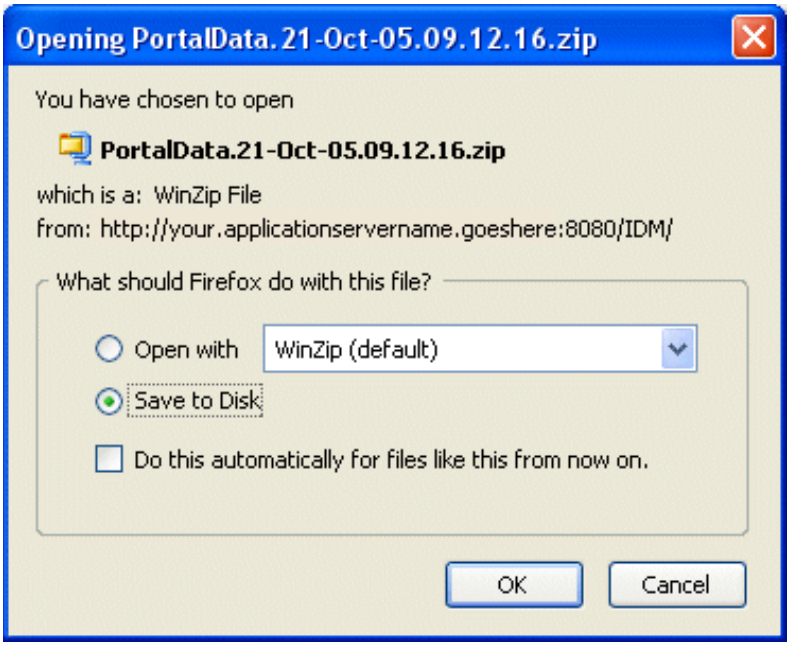
---

3 When you are done making selections, click the *Export Portal Data* button.

Your new *Portal Data Export ZIP file* is generated, with a default name that includes the current date and time. For example:

```
PortalData.21-Oct-05.09.12.16.zip
```

You are then prompted to save this ZIP file locally (or to open it in an appropriate archive utility). For example:



4 *Save* the Portal Data Export ZIP file to an appropriate location.

14.3 Importing portal data

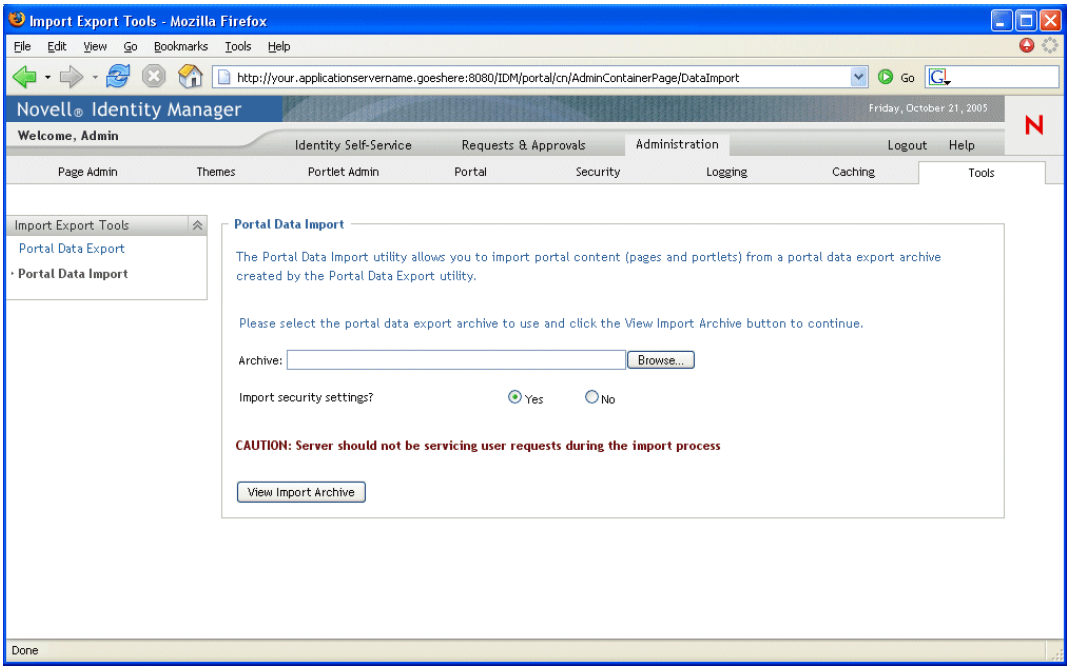
This section describes how to import a Portal Data Export ZIP file to a portal.

NOTE: Remember that, during the import, your target application server must be running but *not currently servicing user requests*.

To import portal data:

1 On the Tools page, select *Portal Data Import* from the navigation menu on the left.

The Portal Data Import panel displays:

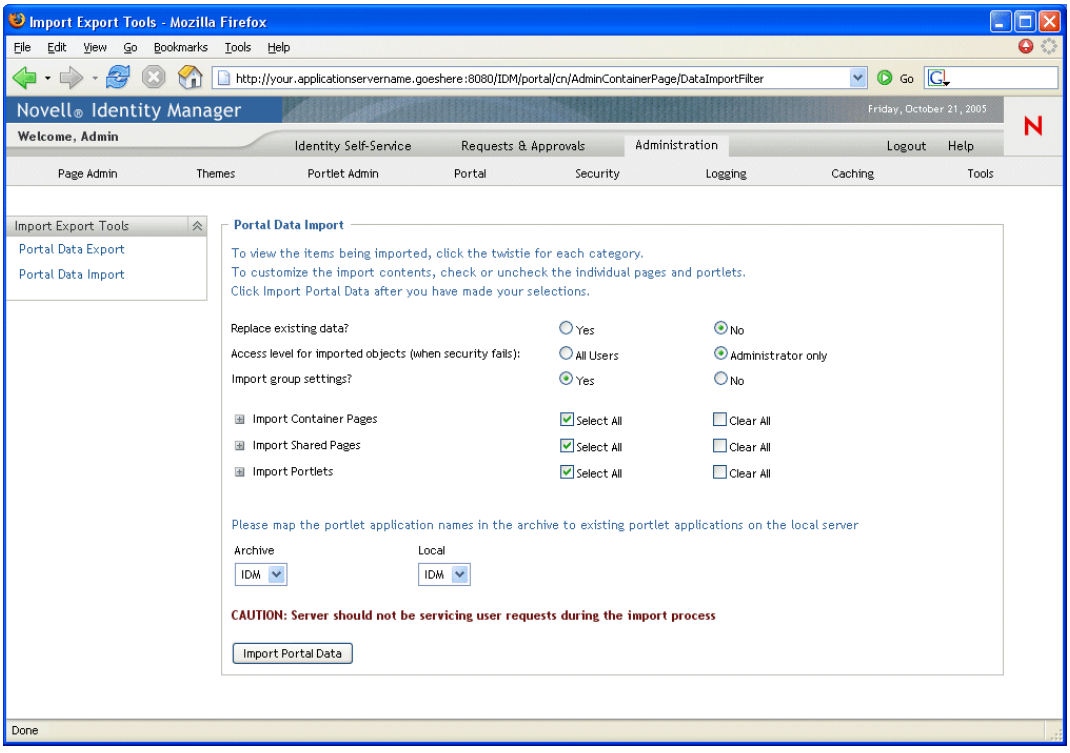


2 Specify the following *general import settings*:

Setting	What to do
Archive	Click the **Browse** button to select the **Portal Data Export ZIP file** to import. For example: `PortalData.21-Oct-05.09.12.16.zip`
Import security settings?	Select one of the following: • **Yes** — If you want to import the permissions that the Portal Data Export ZIP file specifies for access to pages and portlets by users, groups, and containers. Make sure that the users, groups, and containers involved exist in the target portal's identity vault; permissions for missing entities will fail to be imported. • **No** — If you want to ignore the permissions that the Portal Data Export ZIP file specifies.

3 Click the *View Import Archive* button.

The panel displays more specifics about your selected Portal Data Export ZIP file and how you want to import it:



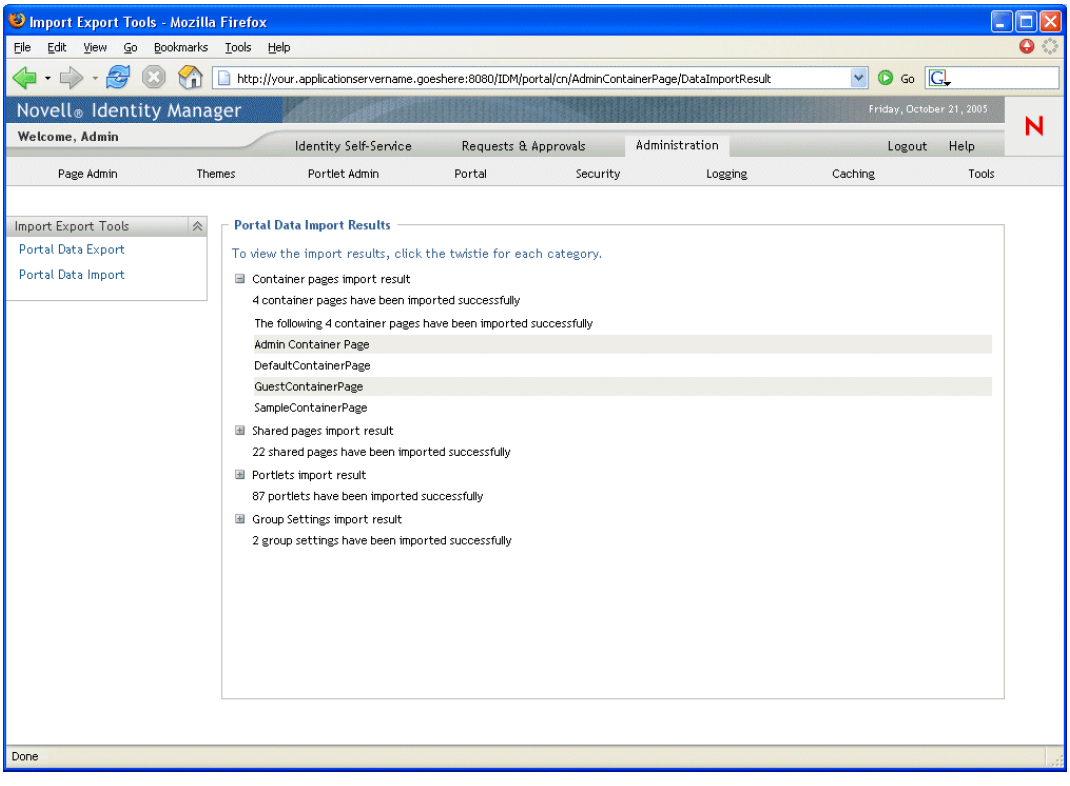
4 Specify the following *detailed import settings*:

Setting	What to do
Replace existing data?	Select one of the following: • **Yes** — If you want the contents of the Portal Data Export ZIP file to overwrite corresponding pages and portlets that already exist in the target portal. For example, if the Portal Data Export ZIP file contains a shared page named MyPage and the target portal contains a shared page named MyPage, that existing page will be overwritten in the target portal. • **No** — If you want to skip the import for all existing pages and portlets.
Access level for imported objects	Select one of the following: • **All Users** — For unrestricted access to imported pages and portlets. • **Administrator only** — For restricted access to imported pages and portlets. **If you chose to import security settings**, then this access level is applied only to those imported pages and portlets where a security setting failed to be imported (typically because specified users, groups, or containers do not exist in the target portal's identity vault). **If you chose not to import security settings**, then this access level is applied to all pages and portlets that are imported.

Setting	What to do
Import group settings?	(If you chose to import security settings) Select one of the following: • **Yes** — If you want to import the default container page and default shared page assignments that the Portal Data Export ZIP file specifies for groups. Make sure that the groups involved exist in the target portal's identity vault; assignments for missing groups will fail to be imported. • **No** — If you want to ignore the default page assignments that the Portal Data Export ZIP file specifies for groups.
Import Container Pages Import Shared Pages Import Portlets	Follow the onscreen instructions to **select the pages and portlets** that you want to import from the Portal Data Export ZIP file to the target portal. **NOTE:** Some portlets that you have not selected for import might still be imported. If you import a page that contains a portlet, but do not select that portlet for import, the portlet is still imported (to ensure that a runtime error does not occur for the imported page).
Please map the portlet application names... Archive/ Local	Use the **Archive** and **Local** dropdown menus to map the portlet application names in the archive (Portal Data Export ZIP file) to existing portlet applications on the local (target) application server.

5 When you're ready to begin the import, click the *Import Portal Data* button.

When the import completes, the *Portal Data Import Results* panel displays:



Unsuccessful imports display in red. To *troubleshoot* import (or export) problems, look at your application server's system console or log file (such as jboss/server/IDM/log/server.log) for messages from the following user application *log*:

`com.novell.afw.portal.util`

Portlet Reference

IV

These chapters tell you how to configure the identity and system portlets used in the Identity Manager user interface.

About Portlets

15

This chapter provides information about the portlets used in the Identity Manager user application. Topics include:

For more information about managing portlets, see Chapter 9, “Portlet Administration,” on page 171.

15.1 Accessory portlets

Accessory portlets provide a diverse set of functions that you can add to your Identity Manager user application. Accessory portlets provide e-mail, file system and other functions. For more information:

Portlet Category	For more information
E-mail File System Miscellaneous	See the Identity Manager Accessory Portlet Administration Guide

15.2 Admin portlets

The portlets in the Admin category are used to control the layout and contents of the user interface.

NOTE: It is recommended that you do not use or modify these portlets. They provide framework services to the user application.

The Admin portlets include:

Portlet Name	Description
Header Portlet	Displays the header information and top-level tab controls for the user interface. There are no preferences for this portlet.
Shared Page Navigation	Displays a menu containing the Identity Manager user application shared pages. Preferences define what is displayed, and how it is displayed. See Section 15.2.1, “Shared Page Navigation portlet,” on page 228.

15.2.1 Shared Page Navigation portlet

The Shared Page Navigation portlet generates links to the Identity Manager user application's shared pages. Preference settings define the shared page links that are displayed. Preferences include:

Preference	What to specify
sharedpages-sorting	The order in which the shared pages are displayed within a category: Ascending/Descending.
sharedpages-sortmode	How to sort the shared pages: Alphabetical or Priority.
sharedpages-category	Specify one or more of the shared pages categories. The category name displays as a header with all of the shared pages in that category displayed as links. If a category does not contain any shared pages, then it does not display. If the shared page is not in a category then it displays as uncategorized.
guest-category	Specify a category whose portlets you want to display in the portal landing page. It must be a pre-existing category and the pages contained in this category must not have any ACL read constraints.

15.3 Identity portlets

The Identity portlets are used by the Identity Self-Service tab of the Identity Manager user application. They include:

Portlet Name	Description
Create	Provides a wizard-based interface that enables users to create objects in the identity vault. See Chapter 16, "Create Portlet Reference," on page 231.
Detail	Lets users display and manipulate an entity's attribute data. See Chapter 17, "Detail Portlet Reference," on page 237.
Org Chart	Lets users view and browse the hierarchical relationships between objects in the identity vault. See Chapter 18, "Org Chart Portlet Reference," on page 251.
Search List	Allows users to search for objects in the identity vault. See Chapter 20, "Search List Portlet Reference," on page 281.

15.4 Password portlets

The password portlets provide the password self-service functionality to the Identity Manager user application. They include:

Portlet name	For more information
IDM Challenge Response IDM Change Password IDM Forgot Password IDM Hint Definition IDM Login	See Chapter 19, “Password Management Portlets Reference,” on page 267

15.5 System portlets

The system portlets provide services to the Identity Manager user application.

NOTE: It is recommended that you do not use or modify portlets in this category.

The system portlets include:

Portlet Name	Description
Portal Page Controller	Displays the shared page that the user has currently selected via the Shared Page Navigation portlet. There are no preferences for this portlet.

Create Portlet Reference

16

This chapter tells you how to use the *Create portlet* in your Identity Manager user application. Topics include:

16.1 About the Create portlet

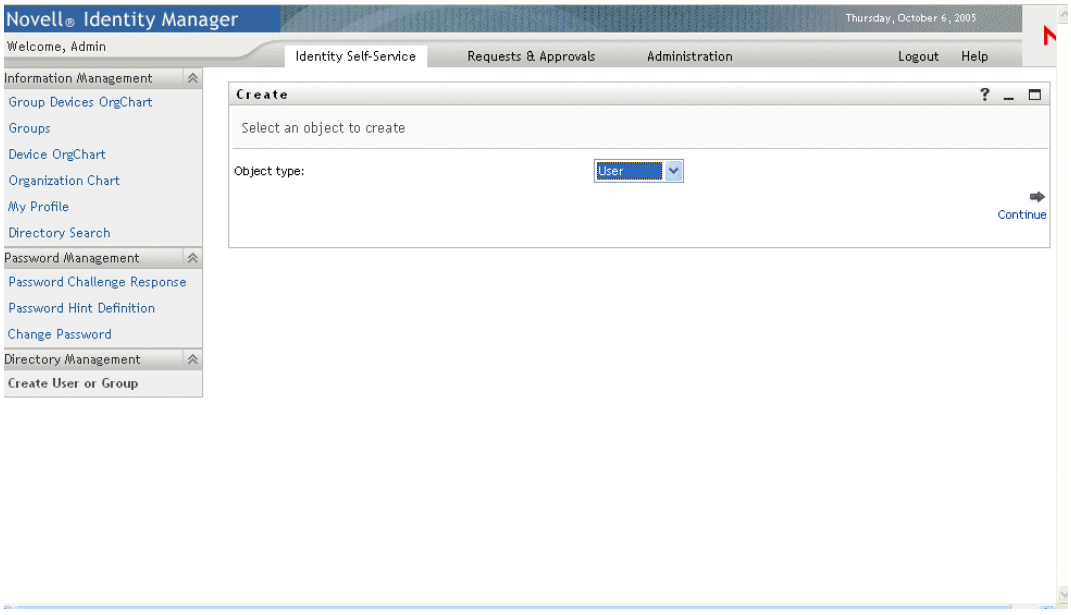
The Create portlet provides an easy-to-use wizard that allows users to create identity vault objects of different types. Portlet preferences control:

- The types of objects that the user can create.
- The attributes that the user can supply.

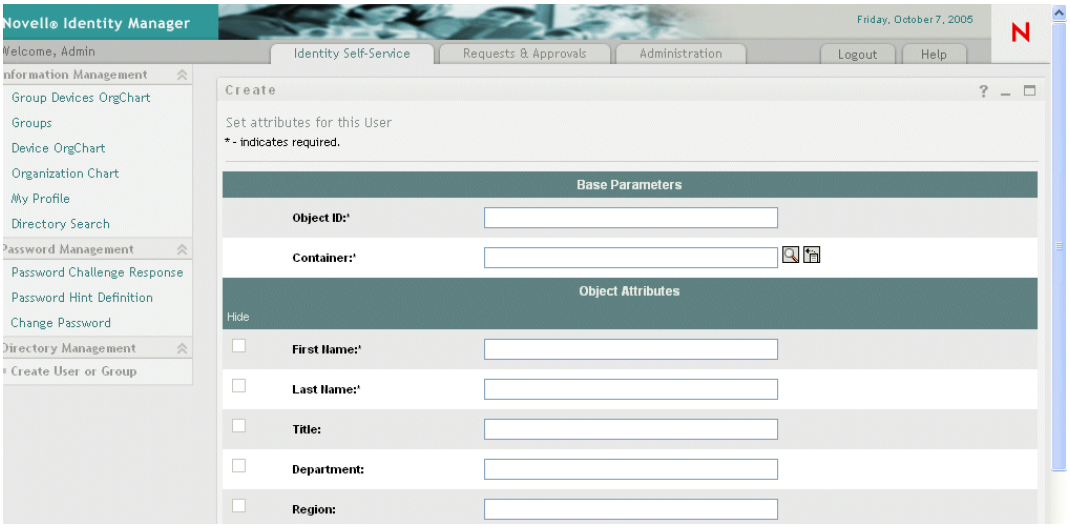
For more information, see Section 16.3, “Setting Create Preferences,” on page 234.

The default configuration of the Create portlet (which is accessed via the *Create User or Group* action of the Identity Manager user application) allows users to create a User, a Group, or a Task Group. This portlet is restricted, by default, to the User Application Administrator. The following example shows how the default Create portlet wizard prompts the user to:

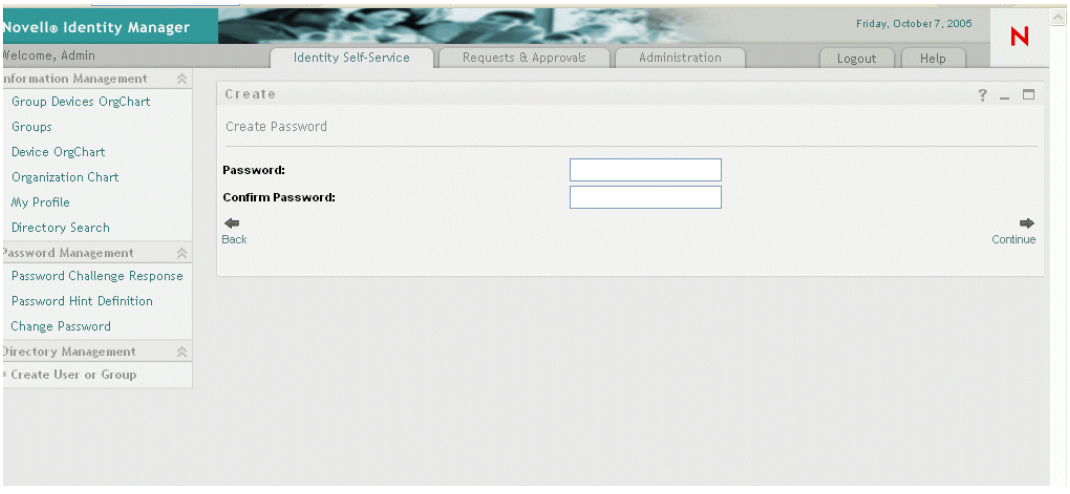
- *Select the type of object* to create:



- *Populate the object’s attributes*:



- *Prompt for a password*, when required by the object type:



If a password policy is assigned, then any custom policy messages are displayed by this portlet.

- *Provide an informational message* when the object is successfully created, that links to the Detail portlet for that object (assuming the Detail portlet is likewise configured) for further editing.

16.2 Configuring the Create Portlet

To configure the Create portlet, you'll:

Step	Task	Description
1	Decide if the default Create User or Group feature meets your needs	If it does then you do not need to take any further action. If it does not then you need to complete the remaining steps.

Step	Task	Description
2	Define the types of objects that you want to allow users to create	Add the objects and attributes to the directory abstraction layer. For more information, see Chapter 4, “Configuring the Directory Abstraction Layer,” on page 75
3	Determine how you want users to access this new portlet	Do you want users to launch this portlet from an existing or a new page? Which users can access the portlet and the page? For more information about pages, see Chapter 7, “Page Administration,” on page 131.
4	Specify the users that have access to the page and the portlet instance	Edit the page security and add the users to the list. For more information on restricting user access to pages, see Chapter 7, “Page Administration,” on page 131. Edit the portlet instance to change security. For more information on restricting user access to portlets, see Chapter 9, “Portlet Administration,” on page 171.
5	Set preferences for the portlet	Preferences let you define: • What objects users can create. • What attributes to supply during the create. For more information, see Section 16.3, “Setting Create Preferences,” on page 234.
6	Test	Verify that the objects are created and that the attributes are populated properly.
7	Establish the proper effective rights in eDirectory for your end users	To create an object, the user will need to be **Trustee** of the organizational unit and the organization in which the object is created.

16.2.1 Directory abstraction layer setup

Objects that can be created and attributes that can be populated by users of the Create portlet must be defined in the directory abstraction layer as follows:

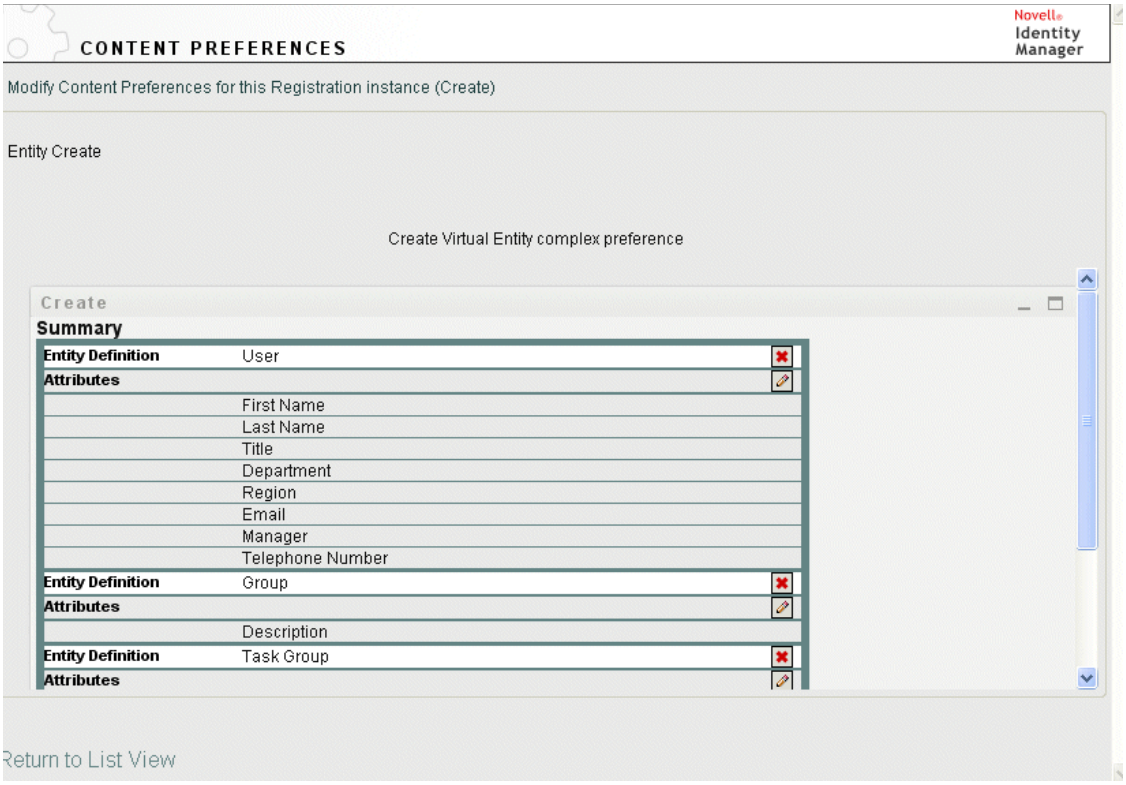
Definition Type	Property	Value
entity	create	Selected
	view	Selected If not selected, the entity will not display in the list of entities that can be created.
	Container for Create	Specify a valid identity vault container. If a valid container is not supplied the root container specified during the user application installation is used.
	Password	Selected, if the entity type requires a password on create. Anyone who has access to Create and has Trustee rights to the OU can create users and assign the **initial password**. When the new user first logs in, they are redirected to the IDM Change Password portlet where they'll modify the initial password. For more information on the IDM Change Password portlet, see Chapter 19, “Password Management Portlets Reference,” on page 267.
attribute	enabled	Selected
	viewable	If enabled or viewable are not selected (false), the attribute cannot be used by the portlet.

For more information on setting up the abstraction layer, see Chapter 4, “Configuring the Directory Abstraction Layer,” on page 75.

16.3 Setting Create Preferences

You can configure the types of objects that a user is allowed to create and the attributes that they are allowed or required to provide by setting preferences.

The Create portlet’s preferences are contained in a single custom preferences page. When you open it, the individual Create preferences display:



The preferences are described below (or you can click the Descriptions button to display online help for this portlet).

Preference	Description
Entity Definition	The name of the object type to create. This represents the beginning of an entity definition block in which you define how the portlet will handle the create operation. **To restrict objects:** Objects listed in the complex preferences are displayed to the user in a dropdown. To restrict the objects that users can create, remove them from this preference sheet via the delete button. **To add other entities:** Click **Add Entity Definition** and complete the wizard.

Preference	Description
Attributes	Controls the attributes that the user is prompted to populate. You must include all of the object's required attributes otherwise the actual create of the object will fail. In addition, the preferences will not save properly if you are missing a required attribute. **To add or remove an attribute:** • Click the Modify Attributes button.  • To add an attribute, select it (from the list of Available attributes). You can multi-select attributes by using the CTRL or Shift keys. • Click the arrow to move it to the Selected list. Do the reverse to remove an attribute. • To reorder the attributes list, click the up and down arrows to the right of the Selected list. Click **Submit**. **Attributes and data types:** The attribute's data type affects the way it is displayed. For example, if an attribute is defined as a Local or Global list subtype, then it displays in a listbox. For more information, see Section 4.3, "Working with entities and attributes," on page 86.

Completing the preferences panel To verify that you've submitted valid entries, click *Submit*. If an entry is invalid, you'll see an error message displayed at the top of the preferences page. Click *Return to List View* once you are able to click *Submit* and no errors occur. You must click *Save Preferences* when you've returned to List View.

Detail Portlet Reference

17

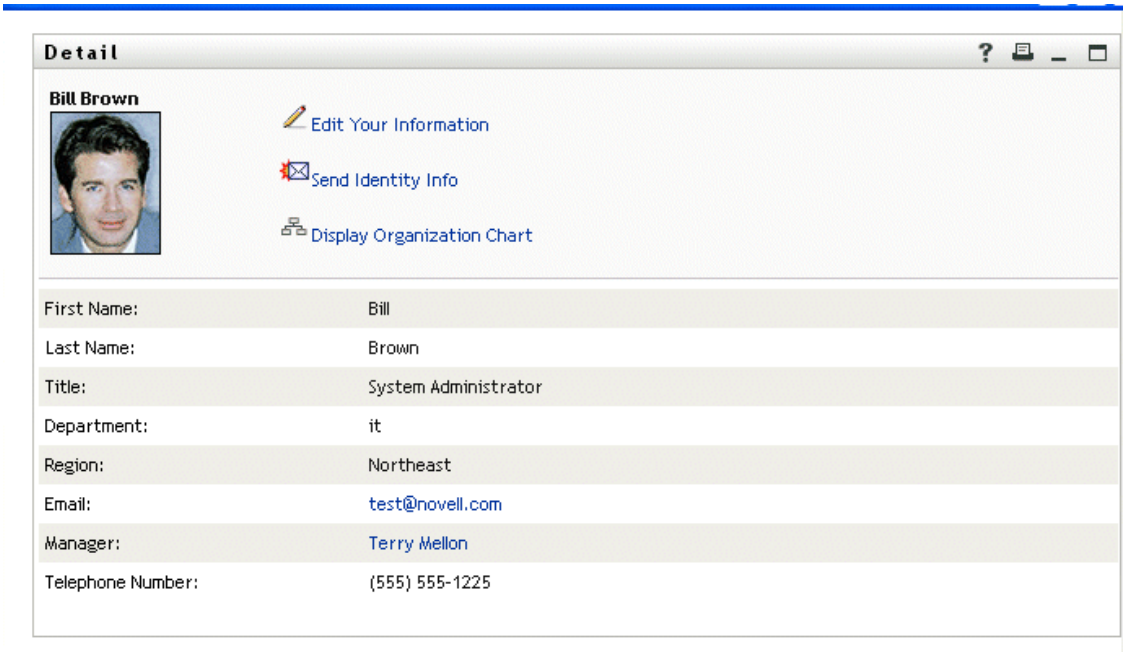
This chapter tells you about the *Detail portlet* which lets users display and manipulate an entity's attribute data. It is the basis for the My Profile action in the Identity Manager user application's Identity Self-Service tab. Topics include:

17.1 About the Detail portlet

The Detail portlet provides users with a detailed view of an entity's attributes and their values. The portlet has two modes: display and edit. When accessing the Detail portlet, users can take advantage of its built-in capabilities to work with this information, including:

17.1.1 Displaying entity data

When accessed, the Detail portlet displays *attribute data about a selected entity*, such as a user or group. For example, here's what the Detail portlet might display when user Bill Brown views his own information:



User images By default, the Detail portlet is configured to include the User Photo attribute. However, if your identity vault does not include this attribute or it is not populated, a default image is displayed at runtime. If you store your user images in a different location, you can configure the portlet to display them instead.

For more information, see "Dynamically loading images" on page 241.

Determining which attributes display

The Detail portlet displays only those attributes that:

- Your *directory abstraction layer* data definitions make available for viewing

  For more information on VDD configuration, see Chapter 4, "Configuring the Directory Abstraction Layer," on page 75.

- Are specified in the Detail *preferences*

  To learn about specifying which attributes display in the Detail portlet, see Section 17.5, "Setting Preferences," on page 248.

- The current user has *rights* to view

  For instance, managers with rights to the salary attribute will see that data, but other users won't.

  For more information, see Section 17.2.2, "Assigning rights to entities," on page 246.

- Are currently populated with a *value*

Determining how attributes display

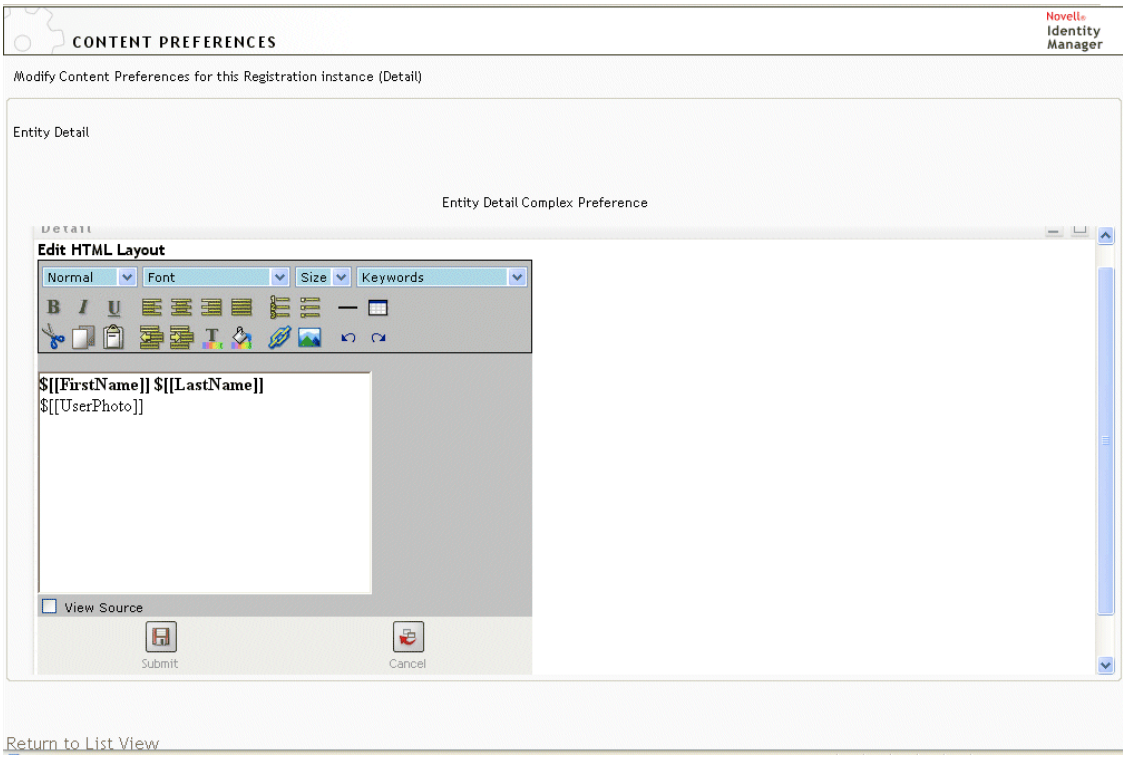
When displaying attributes, Detail *formats the data as text*, except in the following cases:

Format specification in abstraction layer definition	How it displays
Format: email	As a mail-to link
Format: • groupwise-im • aol-im • yahoo-im	As an icon that initiates a chat and adds that user
Data type: Binary **Format:** image	As a button and link to view the image
Data type: Boolean	As disabled radio buttons indicating true or false The buttons display without indicating a default value because the attribute is not actually created for the user until a value is specified.
Multivalue: Selected	As a repeating set of controls for editing, adding, and removing individual attribute values (in the form of a comma-separated list)
Control type:DNLookup	As a link In the example above, a link (`Terry Mellon`) displays to access the Detail data of Bill Brown's manager.
Control type: • Local List • Global List	As the display-label rather than the actual (key) value For example, the EmployeeType attribute displays `Full Time` instead of the actual value `ft`.

Determining what the heading area displays

You can lay out the heading area of the Detail portlet using standard HTML features:

The Detail preferences provide an *HTML Layout Editor* that you can use to create the look and content you want:



Using the HTML Layout Editor

The HTML Layout Editor provides the typical features of an HTML editor for defining text formatting and lists, and specifying anchors and images, and so on.

Keywords When designing your layout, you can use the Keywords dropdown to insert variables within the heading area of the Detail portlet to be replaced at runtime with specific attribute values. You can also type them using this syntax:

`$[[keyword]]`

Where *keyword* is the value of an attribute such as LastName.

You can concatenate attributes using this syntax:

`$[[keyword+keyword]]`

For example:

`$[[FirstName+LastName]]`

You can concatenate as many attributes as you want and can also include quoted strings like this:

`$[[keyword+"sample text"+keyword]]`

This will render the values of the keywords and the quoted text.

NOTE: When a keyword is mistyped in a layout, it will be rendered as-is at runtime (including the $[[]]).

Dynamically loading images To display images, such as user photos, that are stored in your identity vault, you can add the attribute name using the HTML Layout Editor. For example, adding the User Photo attribute to the displays the user's photo. If you store images outside the identity vault, you'll need to use the IMG: tag (from the *View Source mode* of the HTML Editor) as follows:

1. Go to the portlet's preferences and access the HTML Editor.
2. Click *View Source*.
3. Use the IMG: tag to combine a location, an attribute key, and a file extension using a syntax like this:

   ```
   $[[IMG:”URL” + attribute-key-name + ”fileextension”]]
   ```

The following example shows the syntax you'd use if you stored employee photos as JPG images by Last Name in the /images subdirectory of your application server:

```
$[[IMG:"http://myhost:8080/images/"+LastName+".jpg"]]
```

At runtime, the portlet concatenates the URL with the LastName attribute and the file extension .jpg.

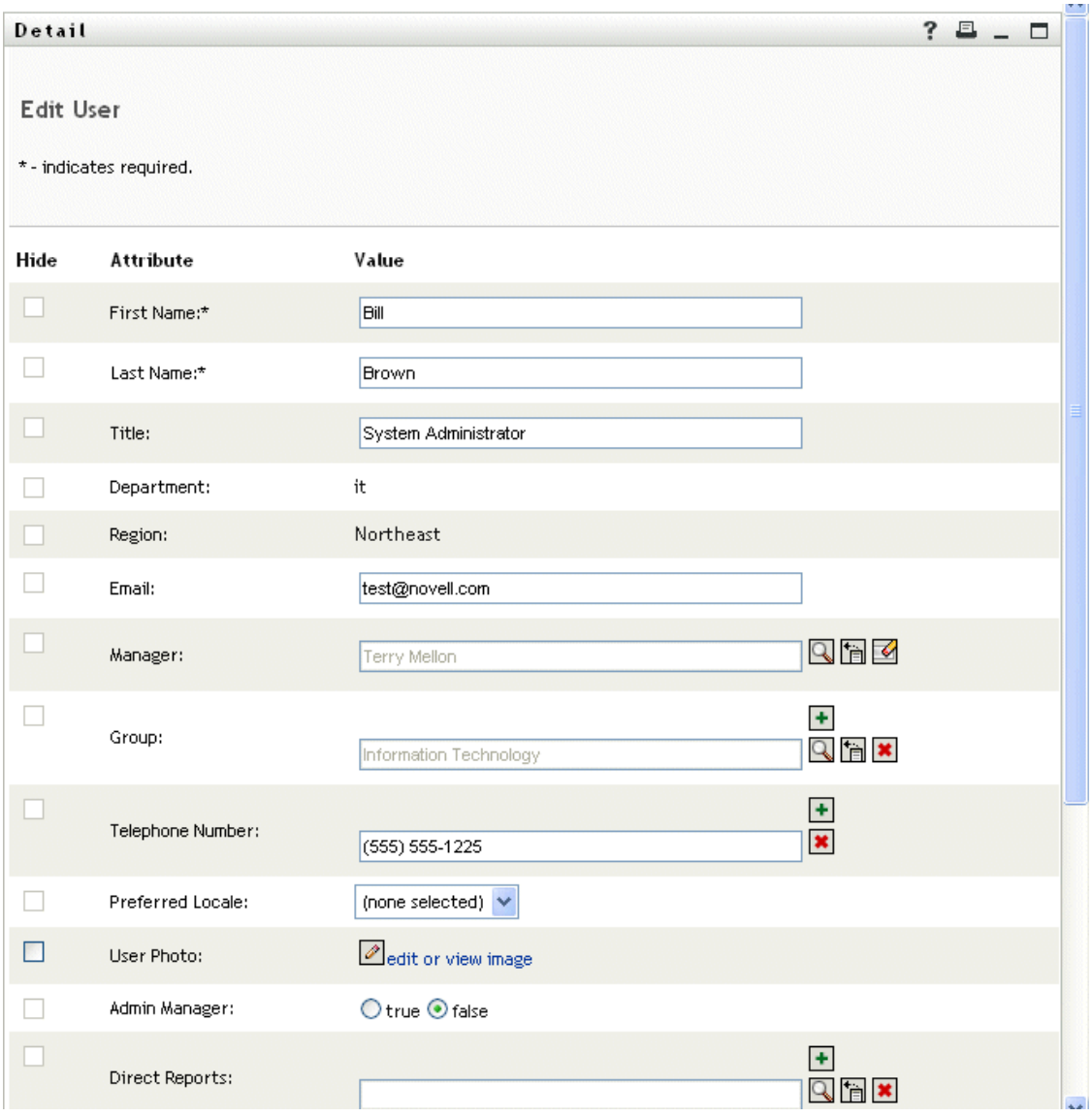
Note that the HTML Editor supports a flexible syntax. It supports any combination of text and attributes so that the syntax is:

```
$[[IMG:”some text” + attribute-key-name + ...]]
```

17.1.2 Editing entity data

The Detail portlet automatically provides an *Edit* link (such as *Edit Your Information* or *Edit User* or *Edit Device*) to switch from display mode to edit mode. This enables users with appropriate rights for the current entity to change its attribute values and save those changes.

For example, here's what Detail might display when user Bill Brown (who has the necessary rights) edits his own information:



NOTE: For Boolean attributes, when both radio buttons are unselected it means that the attribute does not exist for the user. Checking the *true* or *false* radio button will both create the attribute for the user and set its value.

Determining which attributes display

In edit mode, the Detail portlet displays only those attributes that:

- Your *directory abstraction layer* data definitions make available for viewing

  For more information on data definitions, see Chapter 4, “Configuring the Directory Abstraction Layer,” on page 75.

- The current user has *rights* to view

  For instance, managers with rights to the salary attribute will see that data, but other users won’t.

For more information, see Section 17.2.2, “Assigning rights to entities,” on page 246.

An attribute must meet all of the above criteria in order to display in edit mode.

Determining how attributes display

In edit mode, Detail formats each editable attribute as a *text box*, except in the following cases:

Attribute type specification (in VDD files)	How it displays
Data type: Binary Format: image	As a button and link to the Entity Image Upload portlet for viewing, updating, or adding the image
Data type: Boolean	As radio buttons indicating true or false
hide:Selected	As a check box labeled Hide
multivalue=Selected	As a set of controls for editing, adding, and removing attribute values
Control type: DNLookup	As a button to launch the Param List portlet for searching and selecting a DN
Control type: • Local list • Global list	As a dropdown list (allowing multiple selections if applicable)

Attributes that can’t be edited (either by definition or due to inadequate user rights) display as *disabled* or *read only*.

Validating changes

During editing, data validation is automatically performed for the following attribute type specifications:

- Format: email
- Data type: Integer
- Control type: Range

When using a control type of local or global list, it is possible for the displayed list to include values that are outside of an attribute’s specified bounds. But such values will be flagged as out-of-range, and validation will prevent them from being submitted.

Defining a Default My Profile Entity

When defining an entity in the directory abstraction layer, you can specify a value for *Default MyProfile Entity* (in the Configuration element of the directory abstraction layer editor) to specify that another entity definition is to be used for editing. When switching from display mode to edit mode, the Detail portlet always checks whether this element is specified, then uses the appropriate entity definition to present the attributes.

For example, suppose the entity definition for Student includes *user* as the value for *Default My Profile Entity*. In this case, display mode will use the Student entity definition, but edit mode will use the user entity definition.

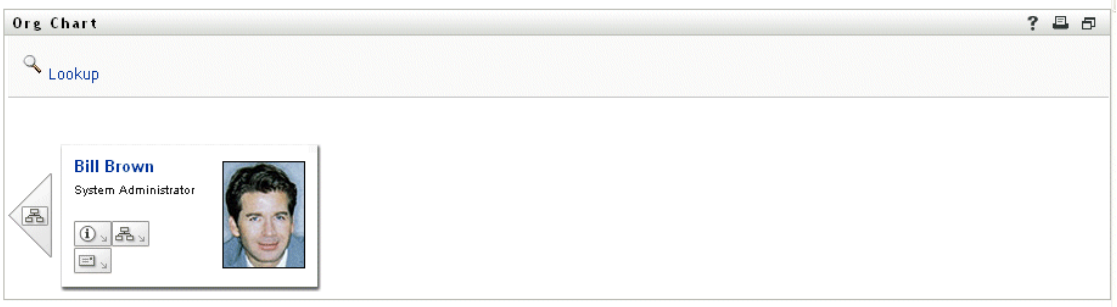
17.1.3 Emailing entity data

The Detail portlet automatically provides a link named *Send Identity Info*. Users can click it to email the URL of the current entity's Detail to one or more other users. By emailing the Detail URL rather than the actual information, security is maintained (because anyone receiving the URL will need appropriate authority to use it).

17.1.4 Linking to an organization chart

The Detail portlet automatically provides a link named *Display Organization Chart*. Users can click it to display the Org Chart portlet for the current entity.

For example, if you're viewing Detail for user Bill Brown, clicking this link displays:

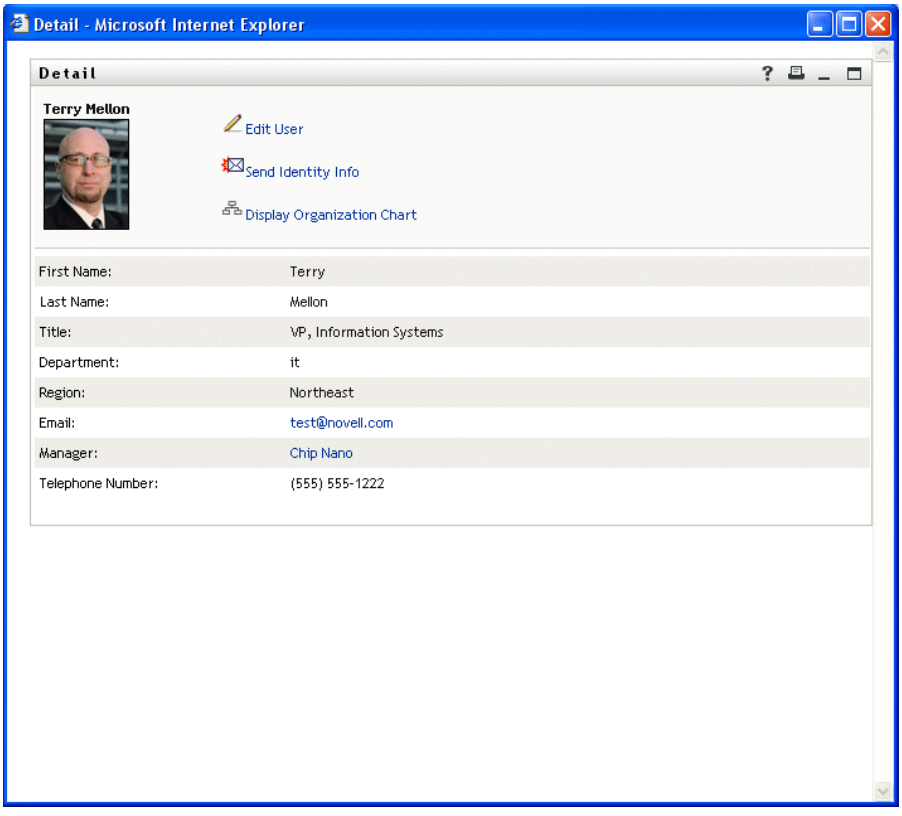


For more information on the Org Chart portlet, see Chapter 18, "Org Chart Portlet Reference," on page 251.

17.1.5 Linking to details of other entities

When configuring the Detail portlet, you may want to enable users to link to related entities from the current one. You can do that by including attributes that are defined (in your directory abstraction layer) with the *control type DNLookup*.

When the Manager attribute is displayed in a user's Detail, it appears as a *link*. Clicking that link displays Detail for the Manager.



For more information on the directory abstraction layer, see Chapter 4, “Configuring the Directory Abstraction Layer,” on page 75.

To learn about specifying which attributes display in the Detail portlet, see Section 17.5, “Setting Preferences,” on page 248.

17.1.6 Printing entity data

By default, the display settings for the Detail portlet enable the *Print* option on the portlet’s title bar. If you keep Print enabled, users can click it to display a printer-friendly version of the Detail content:

To change this or other settings for the Detail portlet, use the Administration tab to update the Portlet Registration for *DetailPortlet* (on the Portlet Administration page).

For more information, see Chapter 9, “Portlet Administration,” on page 171.

17.2 Prerequisites

Before you start using the Detail portlet, you need to know about:

- Section 17.2.1, “Configuring the directory abstraction layer,” on page 246
- Section 17.2.2, “Assigning rights to entities,” on page 246

17.2.1 Configuring the directory abstraction layer

The Detail portlet depends on *directory abstraction layer* definitions in a variety of ways. Instructions on how to configure your abstraction layer data definitions to support specific Detail portlet features are provided in the following sections of this chapter:

- Section 17.1.1, “Displaying entity data,” on page 237
- Section 17.1.2, “Editing entity data,” on page 241
- Section 17.4, “Using Detail on a page,” on page 248

For more information on configuration, see Chapter 4, “Configuring the Directory Abstraction Layer,” on page 75.

17.2.2 Assigning rights to entities

In order to access an entity and its attributes in the Detail portlet, users must have the appropriate *rights assigned in eDirectory*:

To do this	A user needs this right
Display an attribute	Read
Edit an attribute	Write

You can assign rights by specifying that a user is a *trustee* of an object (entity). Then you can specify which rights to assign for which attributes.

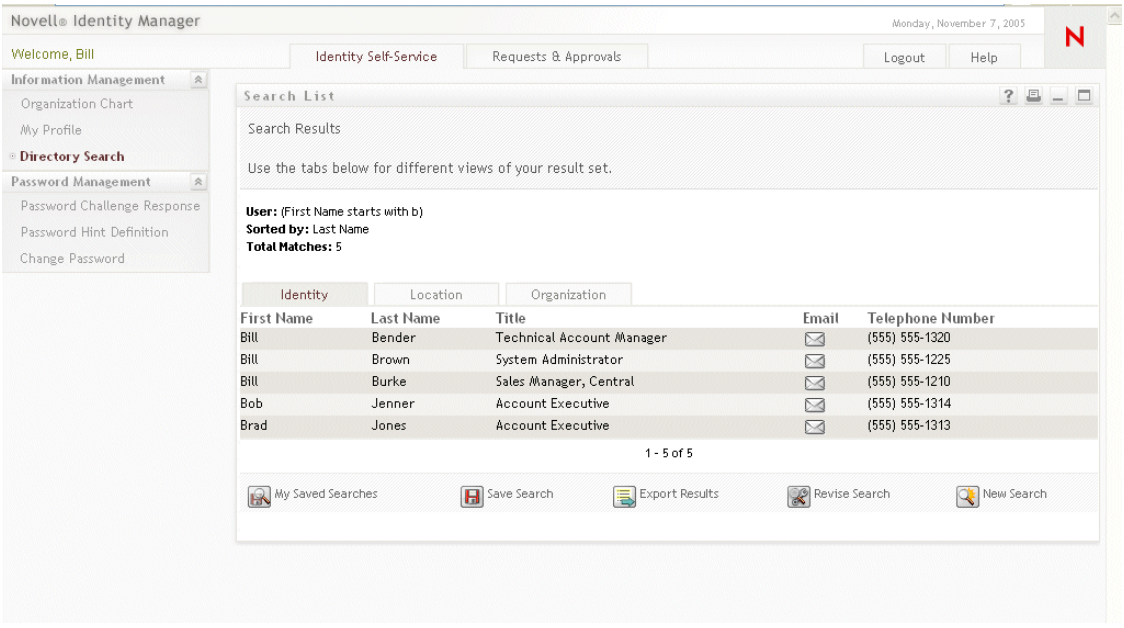
17.3 Launching Detail from other portlets

A common use of the Detail portlet is to launch it after selecting an entity from one of the other identity portlets. You can launch Detail:

- Section 17.3.1, “From the Search List portlet,” on page 246
- Section 17.3.2, “From the Org Chart portlet,” on page 247

17.3.1 From the Search List portlet

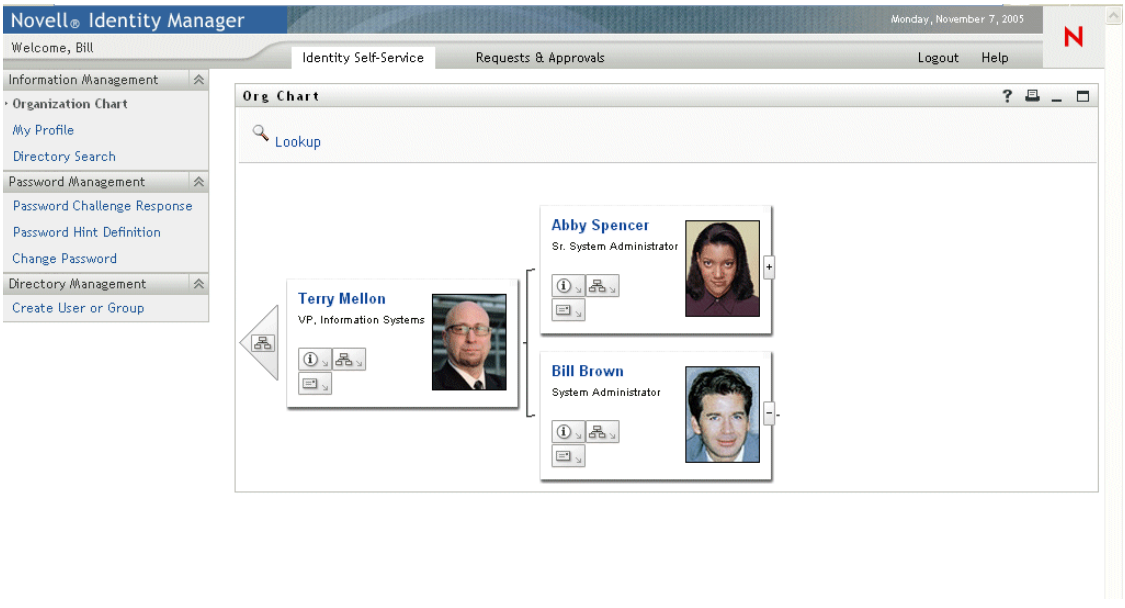
In the Search List portlet, users can *click an entity row* in the search results to display Detail for that entity. For example, clicking the Bill Brown row in the following list will display the Detail portlet with his attribute data:



For more information on the Search List portlet, see Chapter 20, “Search List Portlet Reference,” on page 281.

17.3.2 From the Org Chart portlet

In the Org Chart portlet, users can click the *Identity Actions icon* for an entity and then select *Show Info* to display details for that entity. For example, clicking Show Info for Bill Brown in the following organization chart will display the Detail portlet with his attribute data:



For more information on the Org Chart portlet, see Chapter 18, “Org Chart Portlet Reference,” on page 251.

17.4 Using Detail on a page

If you want to provide users with self service for displaying and possibly editing their own attribute data, you can add the Detail portlet to a *shared page*. When used on a shared, the Detail portlet automatically accesses the data of the current user (or other default entity).

For example, user Bill Brown can log in and go to the following personal page to maintain his own information via the Detail portlet:

To determine which entity definition the Detail portlet is to use in this scenario (where it's accessed on a page, not launched by another portlet), you specify the *Default 'My Profile' Entity* setting in the Configuration element of the directory abstraction layer.

17.5 Setting Preferences

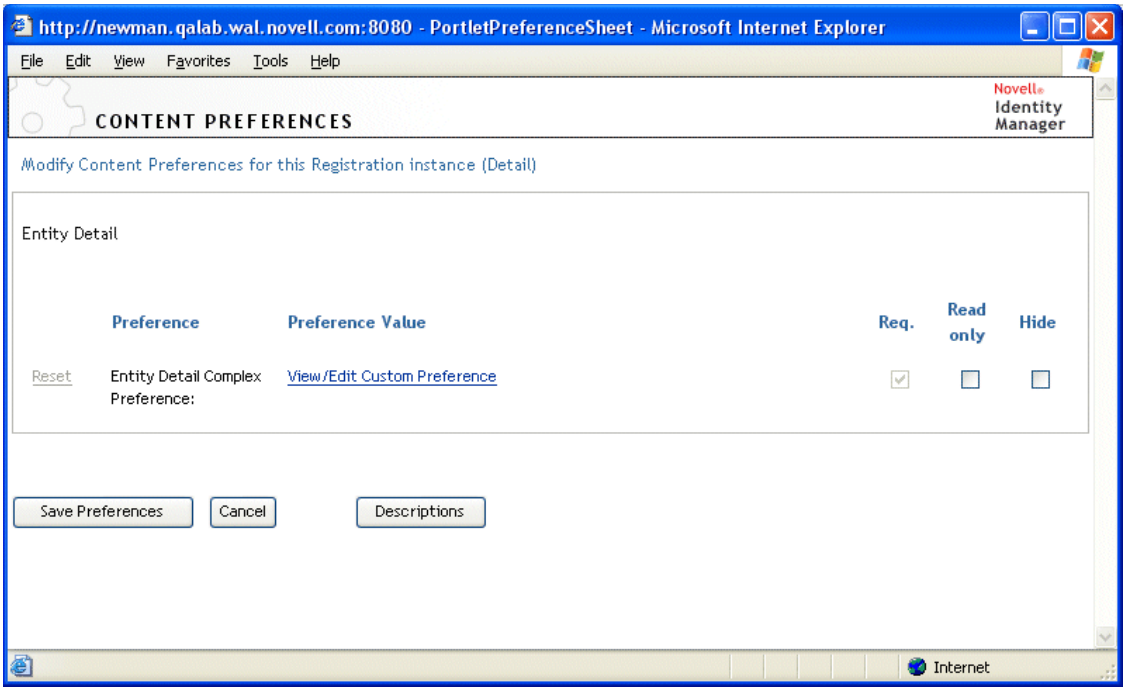
You set preferences to define contents and appearance of the Detail portlet. The way you use the Detail portlet determines where you set its preferences:

To learn about accessing portlet preferences from a shared or container page, see Chapter 7, "Page Administration," on page 131.

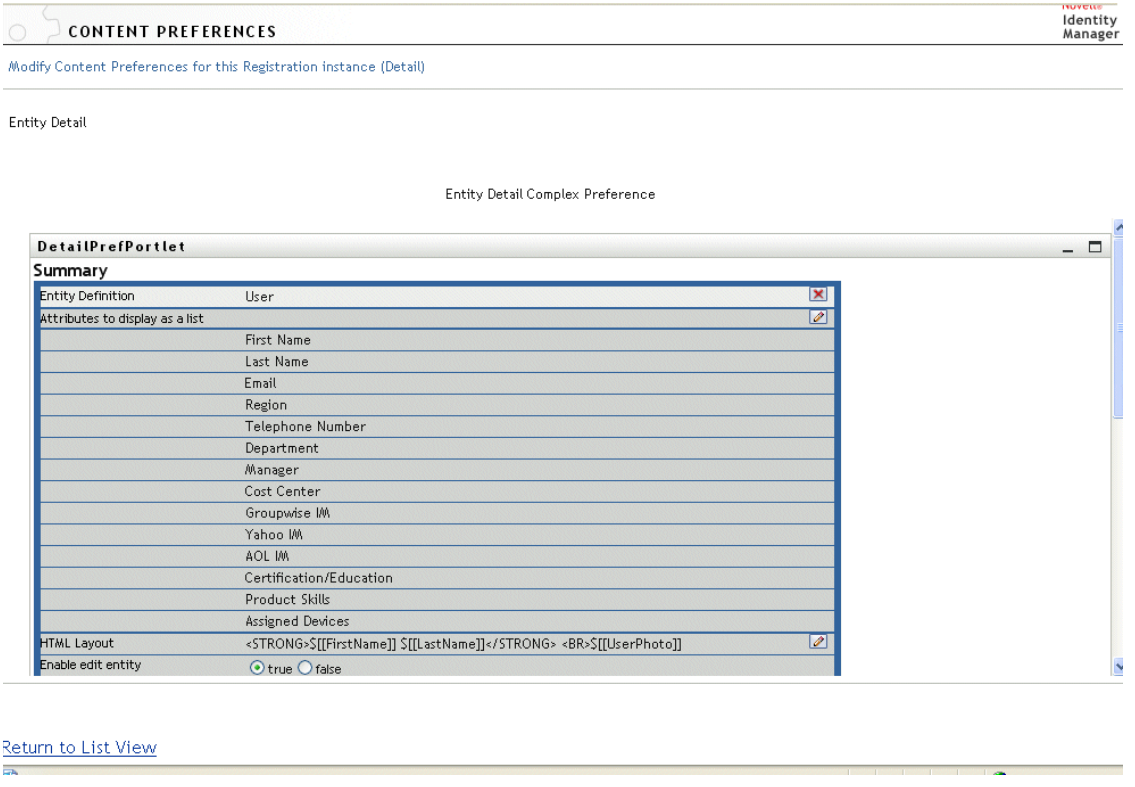
To learn about accessing portlet preferences for a portlet registration see Chapter 9, "Portlet Administration," on page 171.

17.5.1 About the preferences

The Detail preferences are all contained under a single *Detail Complex Preference*:



When you open this complex preference, the individual Detail preferences are presented:



These preferences *apply only to display mode* (not edit mode). They include the following:

Preference	Details
Entity Definition	Specifies the attribute list and HTML layout to display when Detail is used for a particular entity type (such as User, Device, or Group). You can click **Add Entity Definition** to specify Detail support for additional entity types.
Attributes to display as a list	Specifies which attributes of the selected entity you want the portlet to display. These attributes will be listed in the order you choose. A button is provided to let you add or remove attributes as needed.
HTML Layout	Provides a button to open the **HTML Layout Editor**, where you can design the heading area that the Detail portlet is to display for the selected entity. For details, see “Determining what the heading area displays” on page 239.

Org Chart Portlet Reference

18

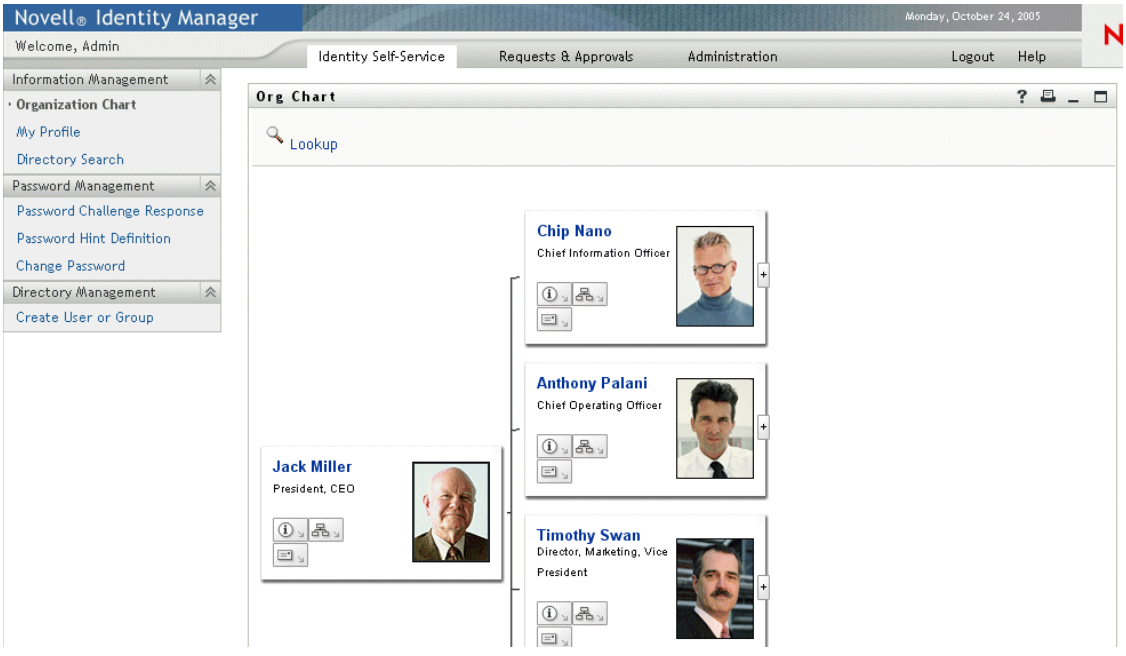
This chapter tells you how to modify existing or add new org chart features to your Identity Manager user application. Topics include:

18.1 About Org Chart

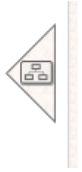
The Org Chart portlet allows end users to view and browse a graphical representation of the hierarchical relationships between objects in the identity vault. For example, you can define Org Chart portlets that show the hierarchy of:

- An organization (such as employees and managers)
- A group’s membership (such as all of the employees in a group)
- Devices assigned to a user (such as cell phones and laptops)

The default configuration of the Identity Manager user application Identity Self-Service tab includes an Organization Chart action. This action is an Org Chart portlet configured to show relationships among user objects in the identity vault. The following example shows how the default Org Chart portlet renders this relationship (using sample data).



Built-in links The Org Chart portlet includes these built-in links.

Link	Description
	Allows the user to navigate to the next upper level. This is only available when viewing a relationship where the parent and child entities are the same.
	Launches the Detail portlet. This built-in link is configurable via the org chart layout preferences described in “Org chart layout preferences” on page 258
	Displays a list of org charts. Lets users choose an org chart to view. This list of org charts is dynamic. It displays other org charts that share the same parent entity type. For example, if you are viewing a manager/employee org chart (the parent entity is user) and you click this icon, then the list of org charts you can view will only contain relationships where the parent entity is also user. This built-in link is configurable via the org chart layout preferences described in “Org chart layout preferences” on page 258
	Launches an email tool to: • Send the identity details of the currently selected user • Compose an email This built-in link is configurable via the org chart layout preferences described in “Org chart layout preferences” on page 258
	The Lookup Link allows users to perform entity searches. The searches result in the found entity becoming the top node of the chart displayed.
	Lets users drill down to the next level.

For more information about adding and restricting the built-in links on your org charts, see “Org chart layout preferences” on page 258.

18.1.1 About Org Chart Relationships

The Org Chart portlet displays relationships that are defined in the directory abstraction layer. The following relationships are available after the Identity Manager user application is installed.

- Group’s membership
- Manager-Employee
- User Groups

To learn more about creating or modifying Org Chart relationships, see Chapter 4, “Configuring the Directory Abstraction Layer,” on page 75.

NOTE: Dynamic groups are not fully supported by the org chart portlet. You cannot define a dynamic group as the Parent entity of a relationship, but you can define a dynamic group as the child entity in a relationship.

18.1.2 About Org Chart display

By default, the org chart displays within the portlet's frame in an area defined by the Portlet Width and Portlet Height preferences. If the contents require more than the defined area, the portlet borders will expand and the page height and width will do so as well. Users can get a fully displayed org chart by clicking the maximize icon available through the portlet's title bar. (The org chart is displayed in fully maximized mode, by default, when launched from the Detail portlet.)

User images By default, the org chart layout for the User object includes the User Photo attribute. However, if your identity vault does not include this attribute or it is not populated, the org chart ignores this attribute at runtime. If you store your photos in a different location, you can configure the org chart to display those photos instead.

For more information, see Section 18.2.3, "Dynamically loading images," on page 264.

18.2 Configuring the Org Chart portlet

To configure the org chart portlet you'll need to:

Step	Task	Description
1	Define the relationship that you want to display	You can use one of the predefined relationships that are installed with the Identity Manager user application, or you can create your own. For more information about defining a relationship, see Chapter 4, "Configuring the Directory Abstraction Layer," on page 75.
2	Verify that the entities and attributes that you want to use in the relationship are available in the directory abstraction layer	For more information about defining a relationship, see Section 18.2.1, "Directory abstraction layer setup," on page 254.
3	Determine where you want to display this relationship	Do you want to create a new page for launching the org chart? Or, do you want to launch it from the Detail portlet or from another org chart? For more information about creating pages and adding portlets to those pages, see Chapter 7, "Page Administration," on page 131.
4	Set preferences for the portlet	Preferences let you define: • Which attributes to display • How to display them (their HTML layout) For more information, see Section 18.2.2, "Setting Org Chart Preferences," on page 254.
5	Test	Test the relationship definitions and layout

Step	Task	Description
6	Set eDirectory rights and establish any indexes needed to enhance performance	**Effective rights**—To display attributes defined by the portlet, users must have **Read** rights to the attributes. **Performance enhancement—**The performance of the org chart display can be enhanced by adding an eDirectory value index to the relationship's child attribute because the child attribute is used to do an LDAP search.

18.2.1 Directory abstraction layer setup

The entities and attributes displayed within an Org Chart must be defined in the directory abstraction layer. The following table shows the attributes and properties that you must set for each entity and attribute displayed in an Org Chart.

Definition type	Setting	Value
entity	view	Selected (true)
attribute	read	Selected (true)
	search	Selected (true)

Lookup Link requirements The Lookup Link allows users to navigate the org chart by performing searches for other objects of the same type as the Parent Entity key. It requires that the parent entity key have at least one attribute with the *require* and *search* access properties set to true (selected in the directory abstraction layer editor). If not, the Lookup Link's Object Lookup dialog cannot be populated and it displays an empty dialog.

For more information on entity and attribute configuration, see Chapter 4, "Configuring the Directory Abstraction Layer," on page 75.

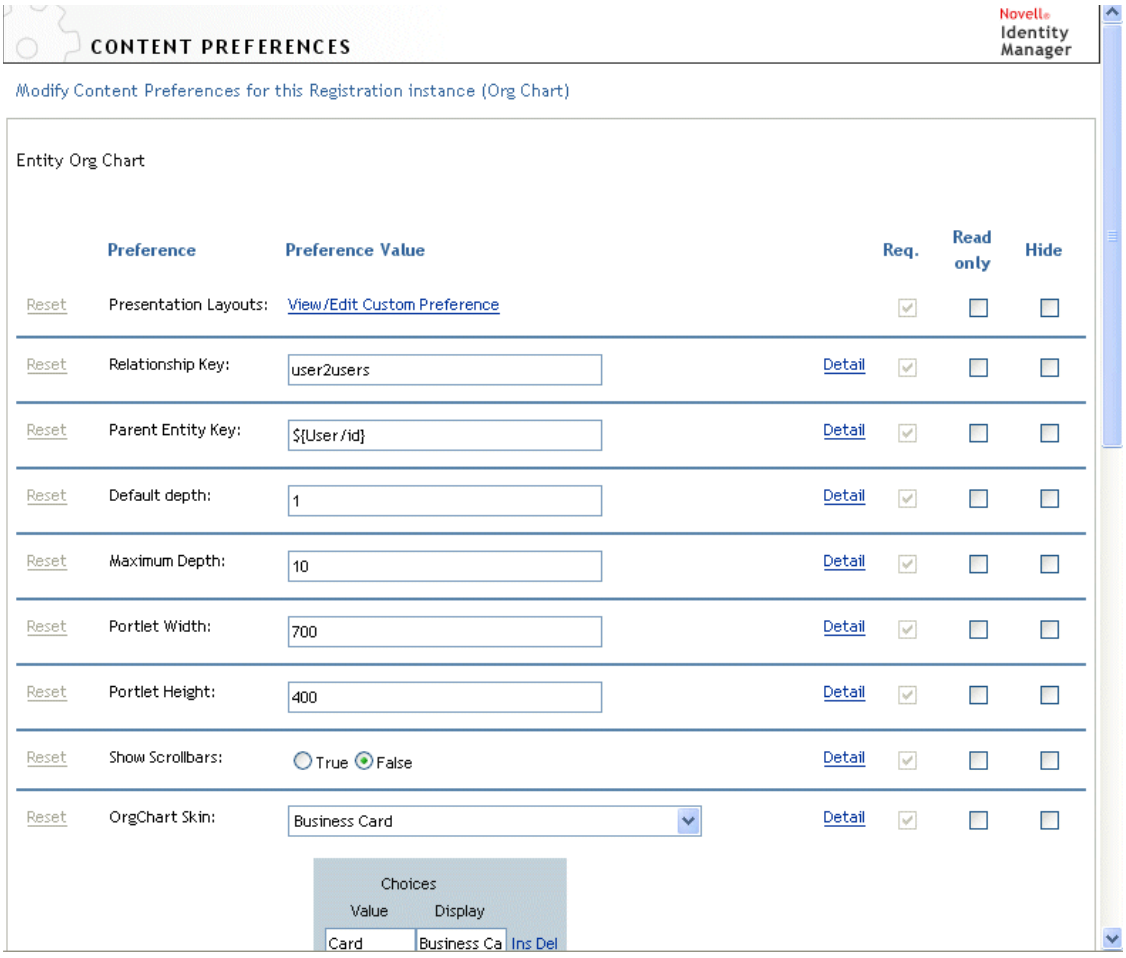
18.2.2 Setting Org Chart Preferences

You define two types of preferences:

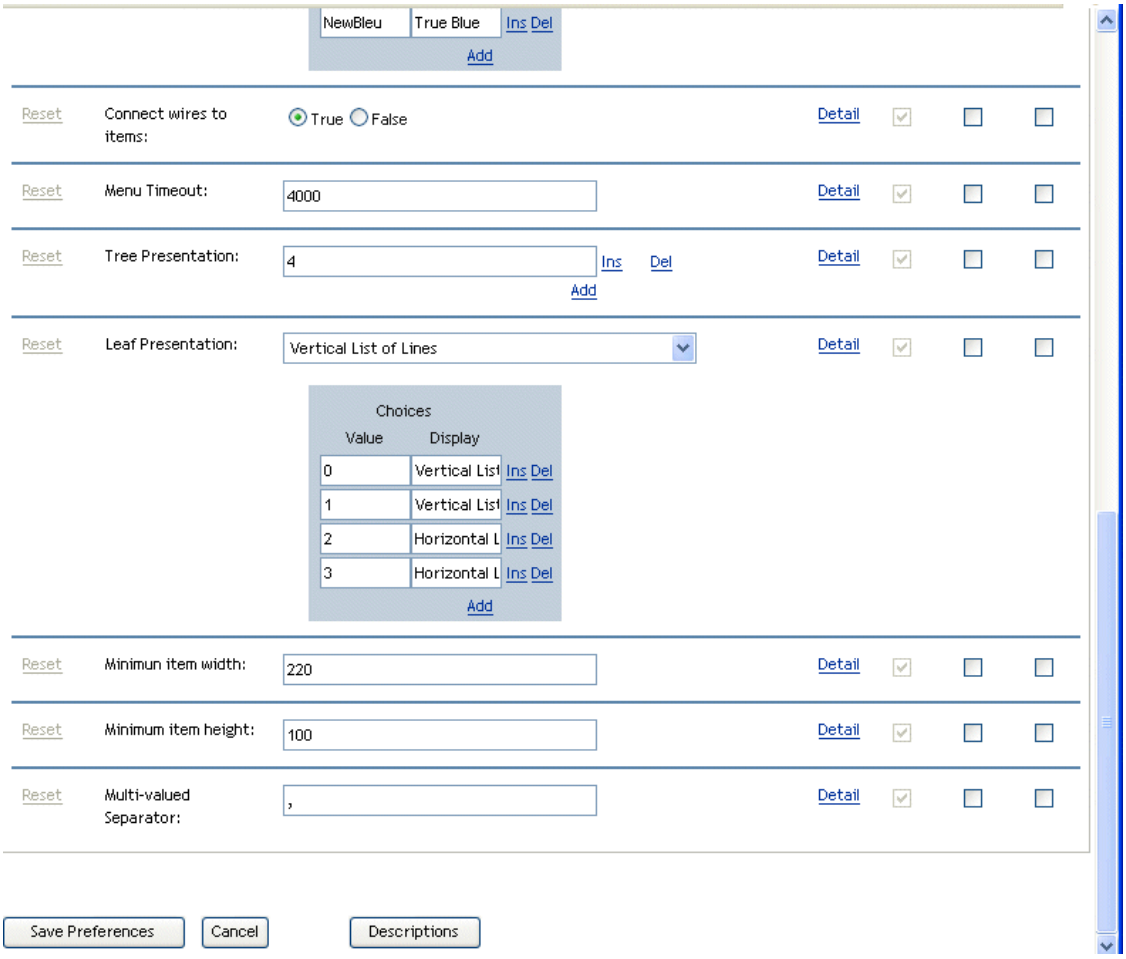
- "Org Chart relationship preferences" on page 254
- "Org chart layout preferences" on page 258

Org Chart relationship preferences

The Org Chart relationship preferences are contained in a single preferences page.


CONTENT PREFERENCES
Novell Identity Manager
Modify Content Preferences for this Registration instance (Org Chart)
Entity Org Chart
Preference
Preference Value
Req.
Read only
Hide
Reset
Presentation Layouts:
View/Edit Custom Preference
Relationship Key:
user2users
Parent Entity Key:
${User/id}
Default depth:
1
Maximum Depth:
10
Portlet Width:
700
Portlet Height:
400
Show Scrollbars:
True
False
OrgChart Skin:
Business Card
Detail
Choices
Value
Display
Card
Business Ca
Ins Del

-
-
-



Preference	What to do
Presentation Layouts	Click View/Edit Custom Preferences to access the layout preferences. They are described in “Org chart layout preferences” on page 258.
Relationship Key	Type the relationship key. This value must correspond to one of the relationship keys specified in directory abstraction layer.

Preference	What to do

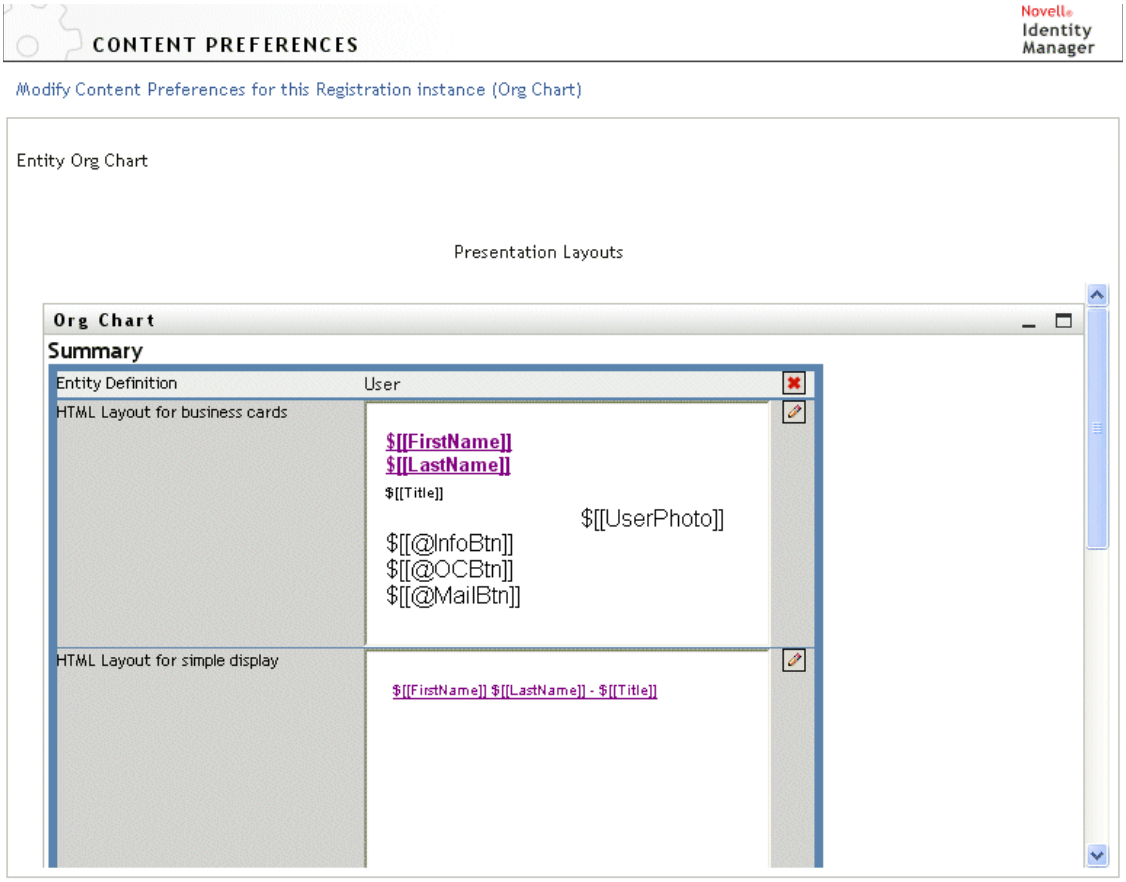
Parent Entity Key	Type the DN of the entity representing the root node of the org chart you want to display, or to display the current user's org chart, type ${User/id}. (The ${User/id} parameter resolves to the current user's DN.)
	
	This value must be within the nodes specified by the search-root property in the directory abstraction layer or the LDAP search fails.
	
	Here are some examples of valid DNs (using sample data):
	
	• To display the user2users Relationship Key with the employee named Jack Miller as the root of the org chart, you'd specify:
	
	`cn=jmiller,ou=users,ou=sample,o=novell`
	
	• To display the group2users Relationship Key with the Accounting group as the root node, you'd specify:
	
	`cn=Accounting,ou=groups,ou=sample,o=novell`

Default depth	Specifies the depth of the org chart when first displayed. • 0—Show only the root • 1—Show the root and its children • 2—Show the root, its children, and grandchildren and so on. If this value is incremented to a value higher than Maximum depth (below), then the Maximum Depth value takes precedence.
Maximum Depth	Defines the maximum depth the user can drill down in an org chart. This is not the same as the ability to navigate through an org chart which is restricted by effective rights.
OrgChart Skin	Business Card eGuide Novell.com Wired True Blue
Connect wires to items	Specifies whether the org chart cards are connected by wires. False means not connected.
Menu Timeout	Number of milliseconds before the current displayed menu (for the built-in links) disappears.

Preference	What to do
Tree Presentation	Defines the OrgChart orientation, distribution and appearance per level of depth. The *n* first values will define orientation, distribution and appearance for the levels from 0 to *n-1*. The last value is used over and over for level of depths greater than *n-1*. Values must be between 0 and 5. Values are: 0: Place card above a vertical list of items 1: Line above a vertical list of items 2: Place card above a horizontal list of items 3: Line above a horizontal list of items 4: Place card before a vertical list of items 5: Line before a vertical list of items
Leaf Presentation	Defines the OrgChart orientation, distribution and appearance for the highest depth of one OrgChart branch
Minimum item width	This value should equal to round('item min height' * 1.618)
Minimum item Height	This value should equal to round('item min width' / 1.618)
Separator for multi-valued attributes	The character used as a separator for attributes with more than one value.

Org chart layout preferences

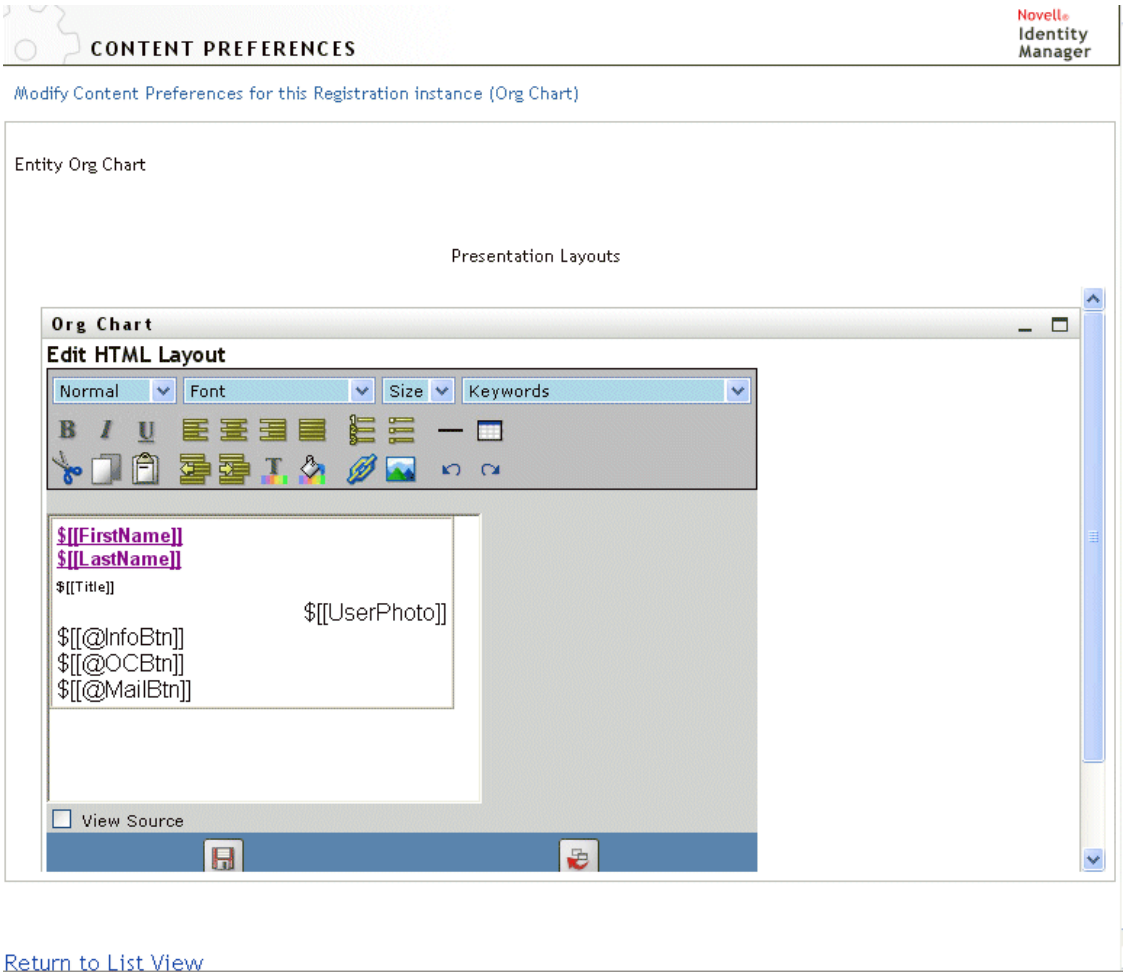
The Org Chart Layout preferences let you define the HTML layout for the display of the org chart entries. You can use the HTML editor of your choice for more precise editing. See “To use an external editor” on page 264.



HTML Layout for business cards—The default layout.

HTML Layout for simple display—The layout displayed when the tree presentation preference is set to 1.

HTML Editor You access the HTML editor by clicking the edit button. The HTML editor looks like this:



Using the HTML Editor

The HTML Editor provides a WYSIWYG interface for defining the layout of the leaves of the org chart. It provides the typical features of an HTML editor for defining text formatting and lists, and specifying anchors and images, and so on. Use the *Keywords* dropdown to place attributes, commands, and navigation URLs within the layout area. When you choose a keyword from the dropdown, it is inserted with the proper syntax, but you can also add HTML within the layout area.

Keywords When designing your layout, you can use the Keywords dropdown to insert variables to be replaced at runtime with specific attribute values. Or you can type references to them using this syntax:

`$[[keyword]]`

Where *keyword* is the value of an entity attribute such as LastName.

You can concatenate attributes using this syntax:

`$[[keyword+keyword]]`

For example:

`$[[FirstName+LastName]]`

You can concatenate as many attributes as you want and can also include quoted strings like this:

`$[[keyword+"sample text"+keyword]]`

This will render the values of the keywords and the quoted text.

NOTE: When a keyword is mistyped in a layout, it will be rendered as-is in the org chart (including the $[[]]).

HTML Editor Features and Keyword usage To use the HTML Editor features and Keywords dropdown list:

Feature	Tip
Insert Link button	To insert a link: **In Mozilla:** 1. Highlight the text you want to hyperlink and click **Insert Link**. 2. Type the URL and click **Create Link**. 3. Save the preferences. **In IE:** 1. Click Insert Link. 2. Type the URL in the popup window. 3. Highlight the text you want to hyperlink, and click **Create Link** (in the popup window). 4. Save the preferences. **NOTE:** If your image or URL is located in the the upper-left quadrant of the HTML Editor, the popup window will overlap it. Since the popup cannot be moved, you'll have to create the text you want elsewhere in the editor and cut and paste it to the correct location.

Feature	Tip
Add Image button	**In Mozilla:** 1. Place mouse focus where you want to insert an image, and click **Add Image**. 2. Type the URL and text then click **Create Image** in the popup window. 3. Save the preferences. **In IE:** 1. Click **Add Image**. 2. Type the URL and text in the popup window, then place mouse focus where you want to insert an image, and click **Create Image** in the popup window. 3. Save the preferences. **NOTE:** If your image or URL is located in the the upper-left quadrant of the HTML Editor, the popup window will overlap it. Since the popup cannot be moved, you'll have to create the text you want elsewhere in the editor and cut and paste it to the correct location.
Keyword Dropdown: Attributes	These are the set of attributes that are available for this entity.
Keyword Dropdown: Commands	These commands allow the Org Chart portlet to launch other identity portlets or built-in features such as IM or email tools. • **IM Action button**—Creates a button to send IMs • **Mail Action button**—Creates a button to send emails • **Org Chart Action button**—Creates a button to switch to another relationship, with the selected entity instance being the parent • **Info Action button**—Launches the Detail portlet For examples of the buttons that are generated, see "Built-in links" on page 251.

Feature	Tip

URLs	**OrgChart Navigation URL Link**—Lets you specify an URL or entity attribute that will display as a link. When the user clicks the link, the Org Chart portlet redisplays with the clicked entity becoming the root node.
	
	Restriction:
	
	This is only valid when the Parent and Child entities in a relationship are of the same object type. For example, in the Manager-Employee relationship, both are users.
	
	Usage Tips:
	
	To use this keyword, you must:
	
	1. Click View Source.
	2. Type the @NavUrl keyword using this syntax:
	
	`<a href="javascript:$[[@NavUrl]]">someText</a>`
	
	where *someText* is the link to display at runtime or an entity attribute. In the following example, **Click here** becomes a clickable link.
	
	`<a href="javascript:$[[@NavUrl]]">Click here</a>`
	
	In this example, the FirstName attribute becomes the clickable link:
	
	`<a href="javascript:$[[@NavUrl]]">$[[FirstName]]</a>`
	
	Usage Restriction:
	
	With Internet Explorer, you **cannot** use this following syntax.
	
	`<a href="$[[@NavUrl]]">someText</a>`
	
	During a save operation Internet Explorer adds:
	
	`http://context before $[[@NavUrl]]`
	
	This means that
	
	`<a href="$[[@NavUrl]]">someText</a>`
	
	becomes
	
	`<a href="http://localhost/.../$[[@NavUrl]]">someText</a>`
	
	and this will **not** display correctly at runtime.

Feature	Tip
	Org Chart Navigation Click Link—Use this keyword for an onClick event. (Enables only the org chart portlet area to be refreshed rather than the whole page.) **Usage Tips:** To use this keyword, you must: 1. Click View Source. 2. Type the @NavClick keyword using this syntax: `<A href="javascript:return false;" onClick="$[[@NavClick]]">$[[SomeAttribute]]</A>` where *SomeAttribute* is an entity attribute that becomes a clickable link. The "javascript:return false;" is required. Omitting it will cause an error.

To save the layouts you define, click *Submit*.

To use an external editor You can use an HTML external editor by:

1. Creating the HTML source for the entity attributes, commands, and keywords using HTML Layout Editor available in the preferences.
2. Copying the HTML source to the editor of your choice.
3. Making the changes that you want.
4. Copying the HTML source back to the HTML Layout Editor preference when you are done editing it.

18.2.3 Dynamically loading images

To display images, such as user photos, that are stored in your identity vault, you can add the attribute name to the business card. For example, adding the User Photo attribute to the business card layout displays the user's photo.

If you store images outside the identity vault, you'll need to use the IMG: tag within the *View Source mode* of the HTML Editor as follows:

1. Go to the Org Chart portlet's preferences and access the HTML Editor.
2. Click *View Source*.
3. Use the IMG: tag to combine a location, an attribute key, and a file extension using a syntax like this:

```
$[[IMG:”URL” + attribute-key-name + ”fileextension”]]
```

The following example shows the syntax you'd use if you stored employee photos as JPG images by Last Name in the /images subdirectory of your application server:

```
$[[IMG:"http://myhost:8080/images/"+LastName+".jpg"]]
```

At runtime, the org chart concatenates the URL with the LastName attribute and the file extension .jpg.

Note that the HTML Editor supports a flexible syntax. It supports any combination of text and attributes so that the syntax is:

```
$[[IMG:”some text” + attribute-key-name + ...]]
```

Password Management Portlets Reference

19

This chapter tells you how to add password self-service and user authentication features to your Identity Manager user application. Topics include:

19.1 Preparing for password management

To get ready to support password self-service and user authentication in an Identity Manager user application, you need to know the following:

19.1.1 About password management features

The password management features supported by an Identity Manager user application encompass *user authentication* and *password self-service*. When you put these features into use, they enable your application to:

- Prompt for *login* information (user name and password) to authenticate against Novell eDirectory
- Provide users with *password change* self-service
- Provide users with *forgotten password* self-service (including prompting for challenge responses, displaying a password hint, or allowing a password change, as needed)
- Provide users with *challenge question* self-service
- Provide users with *password hint* self-service

19.1.2 Required setup in eDirectory

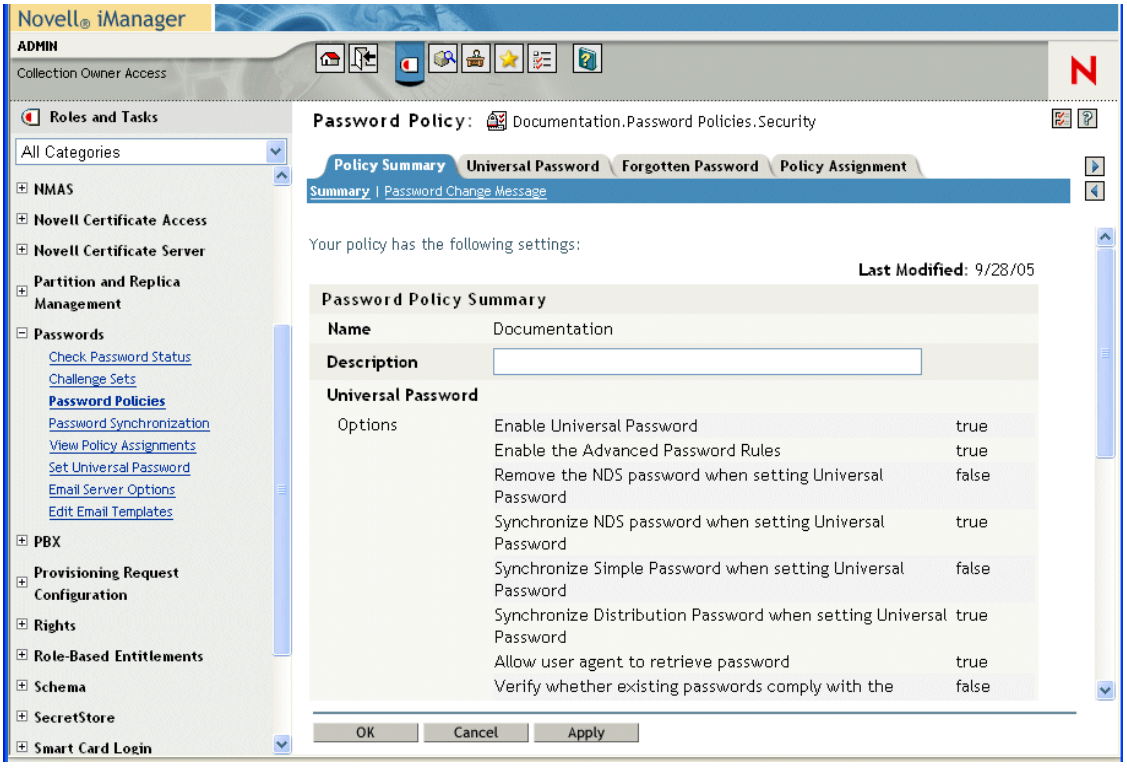
Before you can use most of the password self-service and user authentication features, you need to do the following in eDirectory:

- Enable *Universal Password*
- Create one or more *password policies*

- Assign the appropriate password policies to *users*

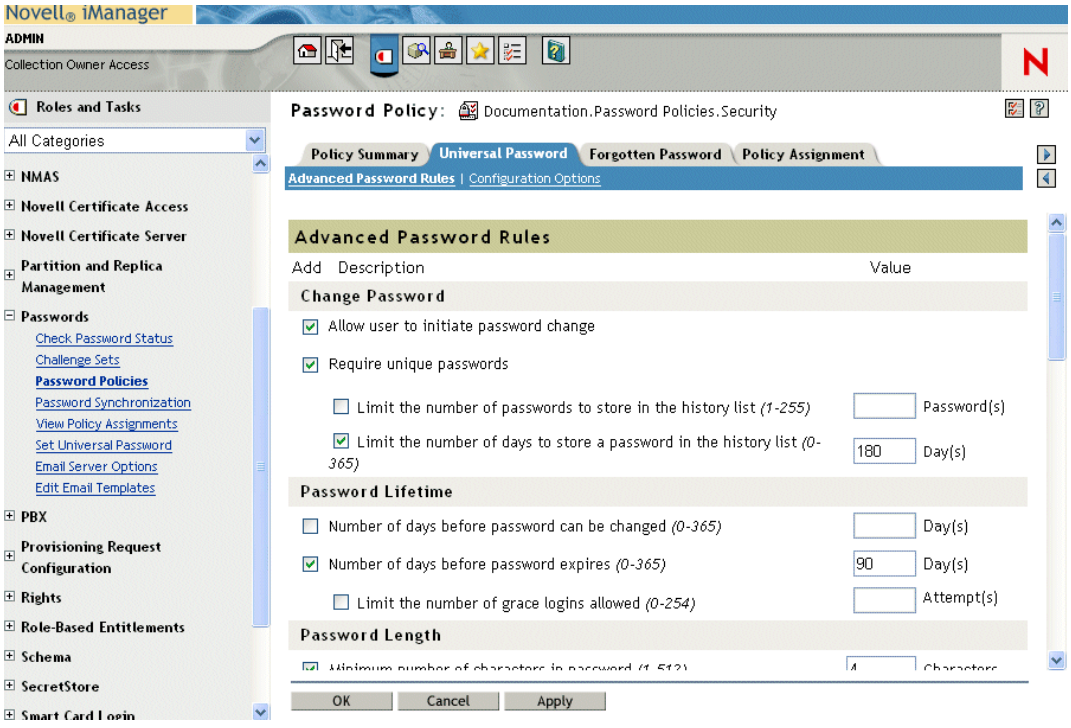
A password policy is a collection of administrator-defined rules that specify the criteria for creating and replacing user passwords. Novell Identity Manager takes advantage of *NMAS* (Novell Modular Authentication Service) to enforce password policies that you assign to users in eDirectory.

You can use *Novell iManager* to perform the required setup steps. For example, here's how someone defined the DocumentationPassword Policy in iManager.

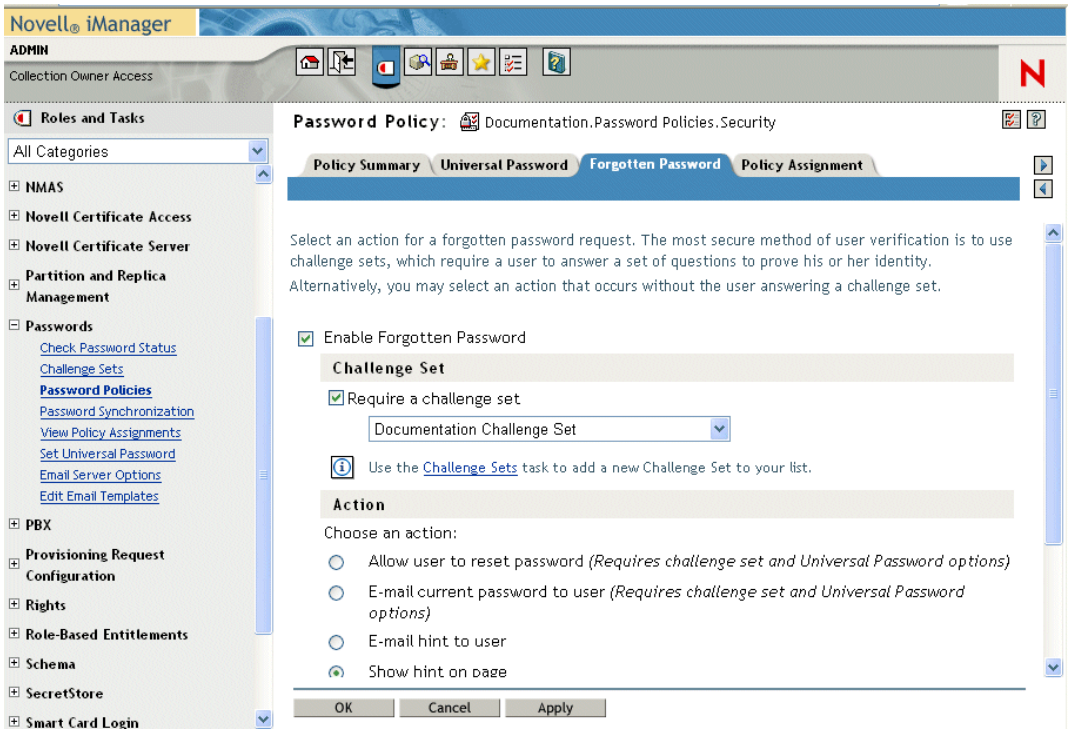


This password policy specifies:

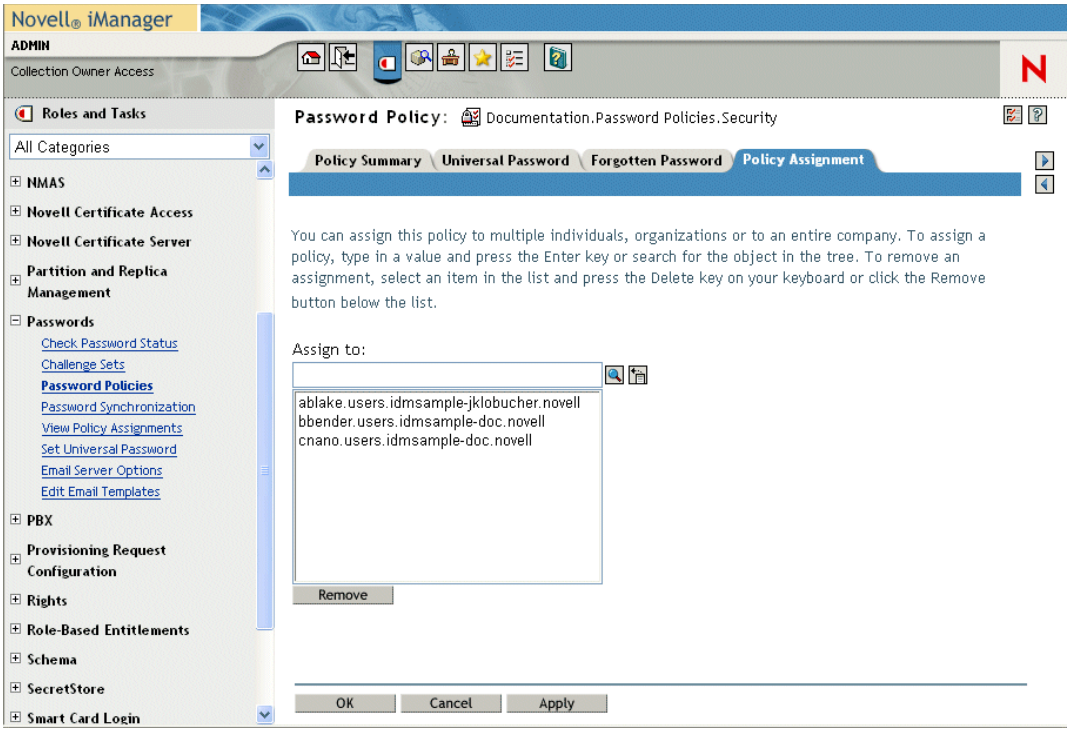
- *Universal Password* settings



- Settings to deal with *forgotten-password* situations



- *Assignments* that apply the policy to specific users



For more information on setting up Universal Password and password policies in eDirectory, see the Novell Identity Manager Administration Guide (http://www.novell.com/documentation/dirxml20/index.html).

19.2 About the password portlets

To implement password self-service and user authentication features in your Identity Manager user application, you'll use the following portlets:

Portlet	Description
Section 19.3, "IDM Login Portlet," on page 271	The IDM Login provides robust user authentication supported by Identity Manager (through Universal Password, password policies, and NMAS). The IDM Login portlet redirects to the other password portlets as needed during the login process.
Section 19.4, "IDM Challenge Response portlet," on page 273	This self-service portlet lets users: • Set up the valid responses to administrator-defined challenge questions, and set up user-defined challenge questions and responses • Change the valid responses to administrator-defined challenge questions, and change user-defined challenge questions and responses
Section 19.5, "IDM Hint Definition portlet," on page 274	This self-service portlet lets the user set up or change their password hint (which may be displayed or emailed as a clue in forgotten password situations).

Portlet	Description
Section 19.6, “IDM Change password portlet,” on page 276	This self-service portlet lets the user change (reset) their Universal Password, according to the assigned password policy. It uses that policy to display the rules that the new password must conform to. If Universal Password is not enabled, this portlet changes the user’s eDirectory (simple) password, as permitted in the user's Password Restrictions.
Section 19.7, “IDM Forgot Password portlet,” on page 278	This self-service portlet uses challenge/response authentication to let the user get information about their password (from NMAS). The result, which depends on the assigned password policy, may include: • Displaying the user’s password hint on the screen • Emailing the hint to the user • Emailing the password to the user • Prompting the user to reset (change) the password

19.2.1 Password self-service portlet modes

The password self-service portlets (IDM Challenge Response, IDM Hint Definition, and IDM Change Password) operate in two modes:

Mode	Description	Runtime behavior
Standalone mode	Portlets run standalone on shared pages.	• If portlet runs **successfully**, it displays a success message with a link to perform the operation again. • If portlet is **unsuccessful**, it displays an error message in the existing form.
Delegation mode	Portlets are displayed on a page as a result of a validation check during login.	• If portlet runs **successfully**, user is redirected to a new portlet or to the user application main page. No success message is displayed. • If portlet is **unsuccessful**, it displays an error message in the existing form.

19.3 IDM Login Portlet

The IDM Login portlet performs a very robust user authentication supported by Identity Manager (through Universal Password, password policies, and NMAS). The IDM Login portlet redirects to the other password portlets as needed during the login process.

19.3.1 Requirements

The IDM Login portlet has the following requirements:

Topic	Requirements
Password policy	This portlet does not require a password policy, unless you want to use advanced password rules or let users click the **Forgot Password** link.
Universal Password	This portlet does not require Universal Password to be enabled, unless you want to use a password policy with advanced password rules.
SSL	This portlet uses SSL, so make sure that your application server is properly configured to support SSL connections to your LDAP realm.

19.3.2 Usage

To use the IDM Login portlet, you need to know about the following:

- “How IDM Login redirects to other portlets” on page 272
- “Using grace logins” on page 273

How IDM Login redirects to other portlets

At runtime, the IDM Login portlet redirects to other password portlets depending on what’s needed to complete the login process. For example:

If the user	IDM Login redirects to
Clicks the **Forgot Password** link	Section 19.7, “IDM Forgot Password portlet,” on page 278
Needs to set up challenge questions and responses	Section 19.4, “IDM Challenge Response portlet,” on page 273
Needs to set up their password hint	Section 19.5, “IDM Hint Definition portlet,” on page 274

If the user	IDM Login redirects to
Needs to reset an invalid password	Section 19.6, “IDM Change password portlet,” on page 276

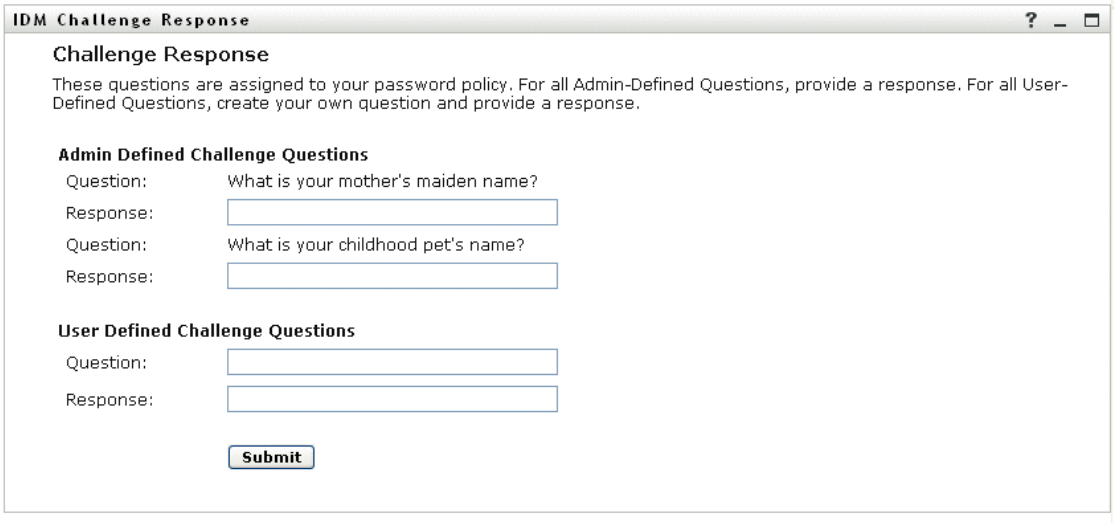
Using grace logins

If you use a grace login, the IDM Login portlet displays a warning message that asks you to change your password and indicates the number of grace logins that remain. If you are on your last login, the IDM Login portlet redirects you to the IDM Change Password portlet.

19.4 IDM Challenge Response portlet

This self-service portlet lets users:

- Set up the valid responses to administrator-defined challenge questions, and set up user-defined challenge questions and responses
- Change the valid responses to administrator-defined challenge questions, and change user-defined challenge questions and responses



19.4.1 Requirements

The IDM Challenge Response portlet has the following requirements:

Topic	Requirements
Password policy	This portlet requires a password policy with forgotten password enabled and a challenge set.
Universal Password	This portlet does not require Universal Password to be enabled.

Topic	Requirements
eDirectory configuration	This portlet requires that you grant supervisor rights to the User Application Administrator for the container in which the logged-in user resides. Granting these privileges allows the user to write a challenge response to the secret store. For example, suppose the LDAP realm administrator is cn=admin, ou=sample, n=novell and you log in as cn=user1, ou=testou, o=novell. You need to assign cn=admin, ou=sample, n=novell as a trustee of **testou**, and grant supervisor rights on **[All attribute rights]**.

19.4.2 Usage

To use the IDM Challenge Response portlet, you need to know about the following:

- "How IDM Challenge Response is used during login" on page 274
- "How IDM Challenge Response is used in the user application" on page 274

How IDM Challenge Response is used during login

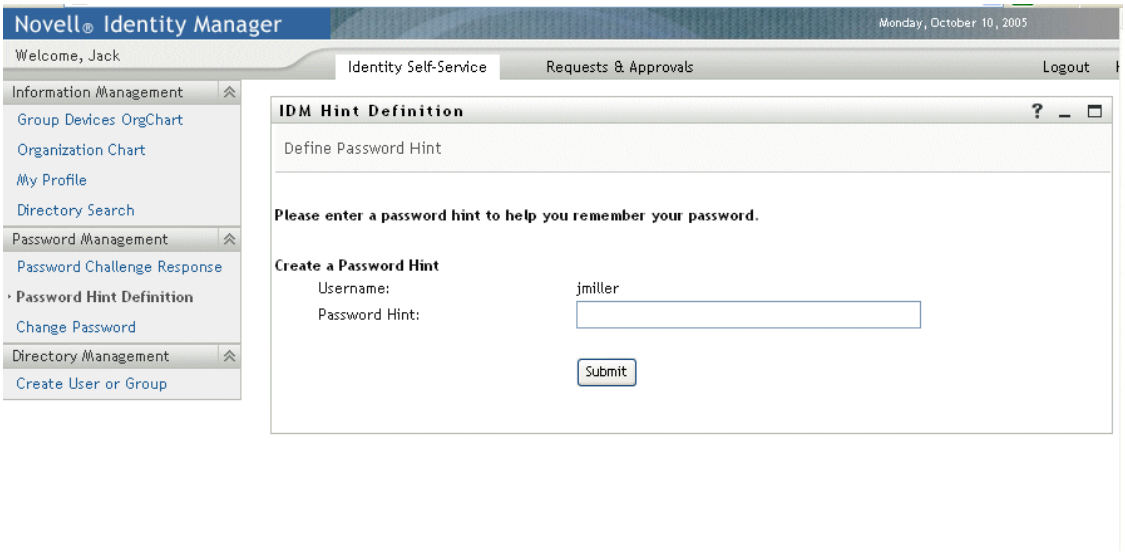
During the login process, the IDM Login Portlet (page 271) automatically redirects to the IDM Challenge Response portlet whenever the user needs to set up challenge questions and responses (for example, the first time a user attempts to log in to the application after an administrator assigns the user to a password policy in iManager. The password policy must have forgotten password enabled and include a challenge set).

How IDM Challenge Response is used in the user application

By default, the user application provides users with self service for changing challenge questions and responses.

19.5 IDM Hint Definition portlet

This self-service portlet lets the user set up or change their password hint (which may be displayed or emailed as a clue in forgotten password situations).



19.5.1 Requirements

The IDM Hint Definition portlet has the following requirements:

Topic	Requirements
Password policy	This portlet requires a password policy with forgotten password enabled and a challenge set.
Universal Password	This portlet does not require Universal Password to be enabled.

19.5.2 Usage

To use the IDM Hint Definition portlet, you need to know about the following:

- “How IDM Hint Definition is used during login” on page 275
- “Using IDM Hint Definition in the user application page” on page 275

How IDM Hint Definition is used during login

During the login process, the IDM Login Portlet (page 271) automatically redirects to the IDM Hint Definition portlet whenever the user needs to set up their password hint (for example, the first time a user attempts to log in to the application after an administrator assigns the user to a password policy in iManager. The password policy will have forgotten password enabled and will have the action set to *Email hint to user* or *Show hint on page*).

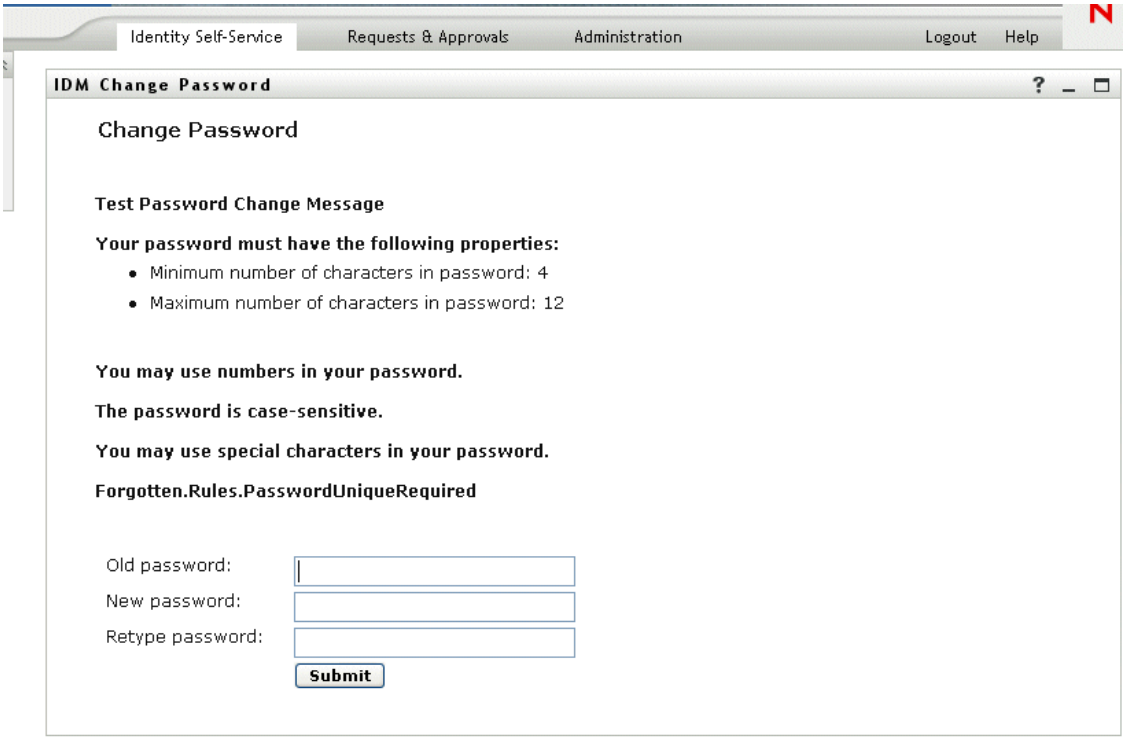
Using IDM Hint Definition in the user application page

By default, the user application provides users with self service for changing their password hint.

19.6 IDM Change password portlet

This self-service portlet lets the user change (reset) their Universal Password, according to the assigned password policy. It uses that policy to display the rules that the new password must conform to.

If Universal Password is not enabled, this portlet changes the user's eDirectory (simple) password, as permitted in the user's Password Restrictions.



19.6.1 Requirements

The IDM Change Password portlet has the following requirements:

Topic	Requirements
Directory Abstraction Layer configuration	No directory abstraction layer configuration is required for this portlet.
Password policy	This portlet does not require a password policy, unless you want to use advanced password rules (with Universal Password enabled).

Topic	Requirements
Universal Password	To use this portlet for a Universal Password, the setting **Allow user to initiate password change** must be enabled in the Advanced Password Rules of the user's assigned password policy. To use this portlet for an eDirectory (simple) password, the setting **Allow user to change password** must be enabled in the user's Password Restrictions.

19.6.2 Usage

To use the IDM Change Password portlet, you need to know about the following:

- "How IDM Change Password is used during login" on page 277
- "Using IDM Change Password in the user application" on page 277

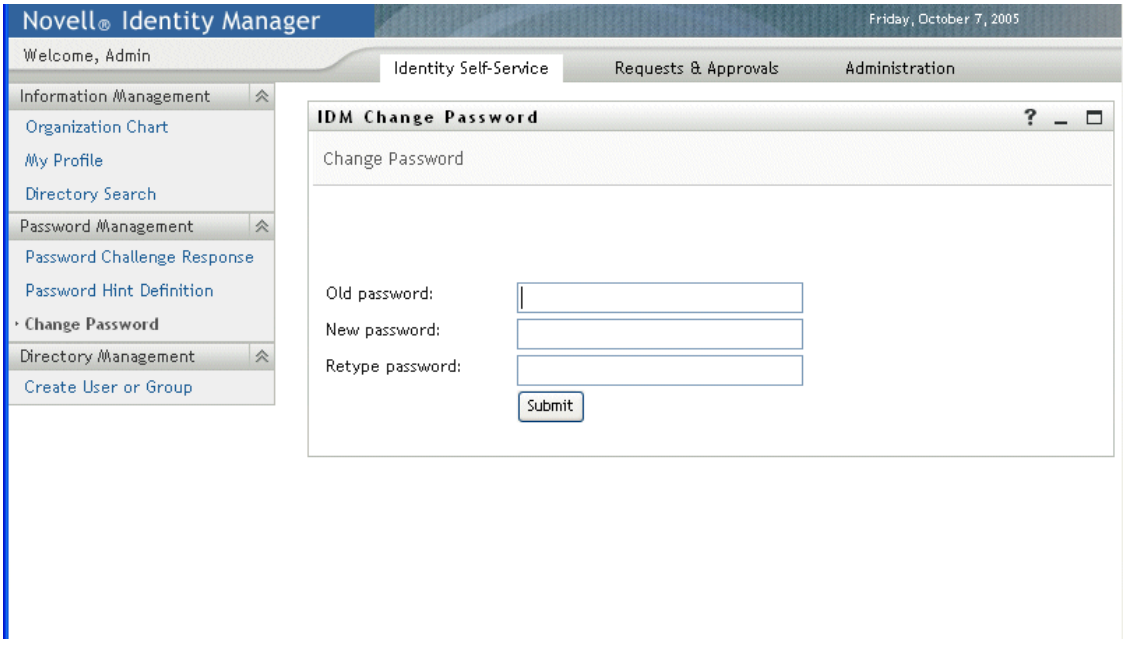
How IDM Change Password is used during login

During the login process, the IDM Login Portlet (page 271) automatically redirects to the IDM Change Password portlet whenever the user needs to reset an invalid password (for example, the first time a user attempts to log in to an application after an administrator implements a password policy that requires users to reset their passwords).

The IDM Forgot Password portlet (page 278) also redirects to IDM Change Password automatically if the user's assigned password policy specifies reset password as the action for forgotten password situations.

Using IDM Change Password in the user application

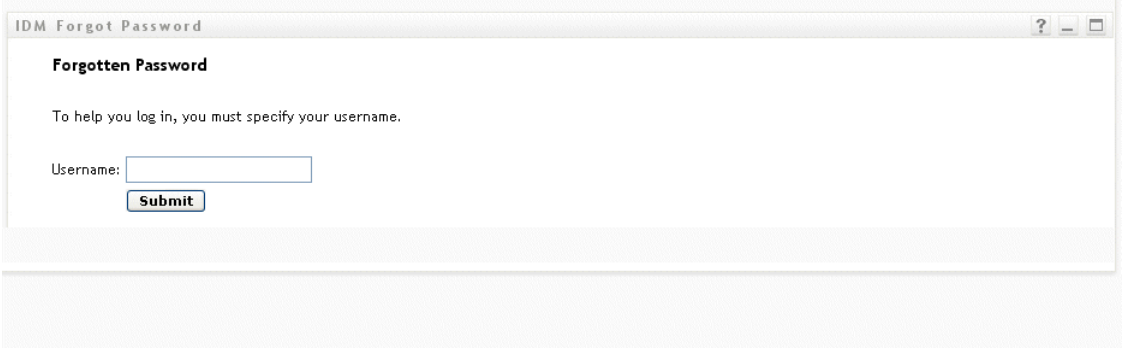
By default, the user application provides users with the password change self-service using the IDM Change Password portlet. For example:



19.7 IDM Forgot Password portlet

This self-service portlet uses challenge/response authentication to let the user get information about their password. The result, which depends on the assigned password policy, may include:

- Displaying the user's password hint on the screen
- Emailing the hint to the user
- Emailing the password to the user
- Prompting the user to reset (change) the password



19.7.1 Requirements

The IDM Forgot Password portlet has the following requirements:

Topic	Requirements
Password policy	This portlet requires a password policy with forgotten password enabled and a challenge set.

Topic	Requirements
Universal Password	This portlet does not require Universal Password to be enabled (unless you want to support the following forgotten password actions: reset password or email password to user).

19.7.2 Usage

To use the IDM Forgot Password portlet, you need to know about the following:

- "How IDM Forgot Password is used during login" on page 279
- "Configuring your environment for email actions" on page 279
- "Preferences for IDM Forgot Password" on page 280

How IDM Forgot Password is used during login

During the login process, the IDM Login Portlet (page 271) redirects to the IDM Forgot Password portlet if the user clicks the *Forgot Password* link. When IDM Forgot Password displays, it does the following:

1. Prompts for *username*.
2. Redirects to the IDM Login Portlet (page 271) to perform *challenge/response authentication* for that user.
3. Performs the *forgotten password action* specified in the authenticated user's assigned password policy. It does one of the following:
   - *Redirects* to the IDM Change password portlet (page 276) so the user can reset their password
   - *Emails* the password or hint to the user
   - *Displays* the hint

NOTE: The IDM Forgot Password portlet is not intended for stand-alone use. That means you should not plan to add it to a shared page in the user application. Placing this portlet on a page creates the potential security risk of persons changing the password on an unattended machine without the user's knowledge or permission.

Configuring your environment for email actions

If you want to support the forgotten password email actions, you need to make sure your *email notification server* is set up properly:

1. Use a Web browser to access *iManager* on your eDirectory server and log in as an *administrator*.
2. Go to *Roles and Tasks>Passwords* and select *Email Server Options*.
3. Specify the appropriate settings, then click *OK*.

The IDM Forgot Password portlet uses two *email templates*. In iManager you will find them in *Roles and Tasks>Passwords>Edit Email Templates*. They are named:

- Password hint request

- Your password request

You can change the content of these templates as needed for your application (but don't change the structure).

Preferences for IDM Forgot Password

The IDM Forgot Password portlet provides the following preferences:

Preference	Details
login-sequence	The NMAS login sequence to use. In this version, the portlet supports only **Challenge Response**.
ldap-sslport	The secure ldap port to use. The default is **636**.
allow-wildcard	Whether the user can type wildcards when entering the username. The default is **false**.
encoding	The character encoding to use. The default is **utf-8**.

Search List Portlet Reference

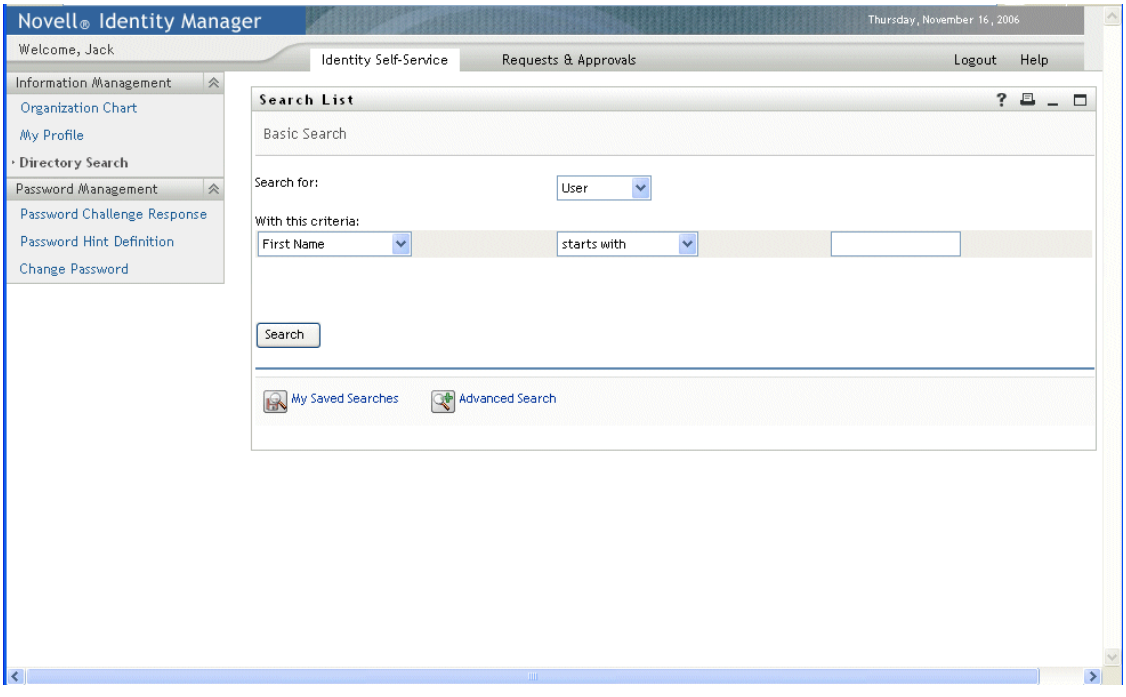
20

This chapter describes how to set up and customize the *Search List* portlet for use with the Identity Manager user application. Topics include:

20.1 About Search List

The *Search List* portlet allows users to search and display the contents of the identity vault. It is the basis for the *Directory Search* action of the Identity Manager user application Identity Self-Service tab. The Directory Search action is configured to allow users to search for users, groups, and task groups, but you can modify it to change the scope of searchable objects and attributes.

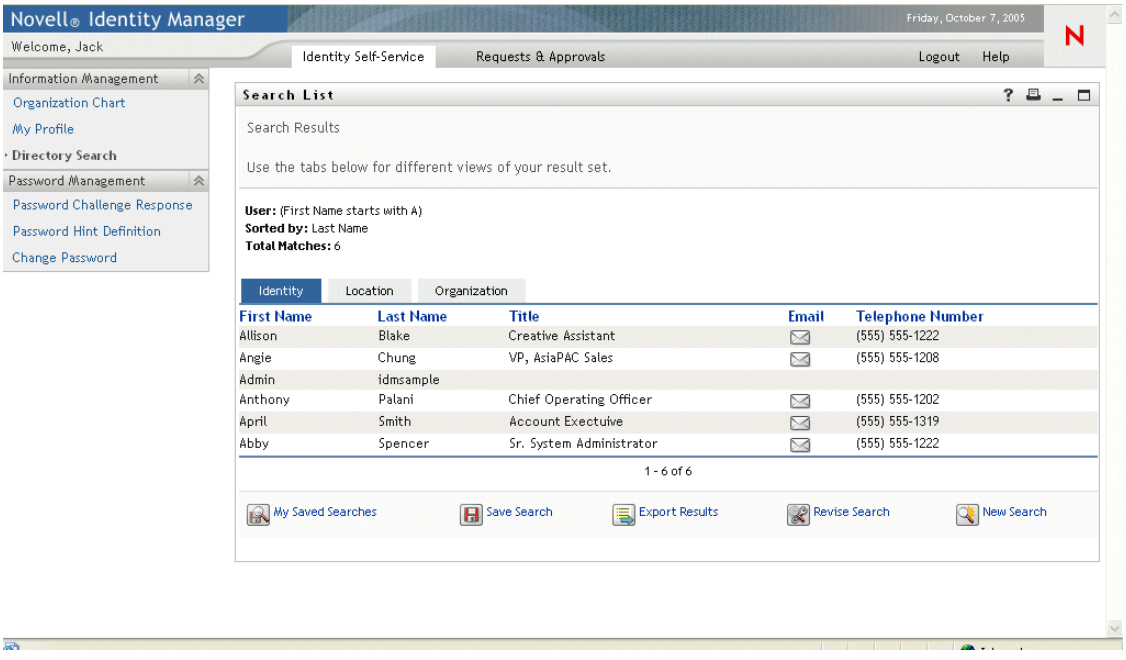
The following example shows how the Directory Search action allows users to define the search criteria.



User interface element	Description
Search for	Users select the object type to search. For more information on defining the contents of this list, see Section 20.2.2, “Setting Search List preferences,” on page 287.

User interface element	Description
With this criteria	Users define the search criteria by selecting attributes and search operators from the dropdown. When the user selects Advanced Search, they are able to specify multiple rows and multiple blocks of search criteria groupings that can be made inclusive (AND) or exclusive (OR). For more information on defining the searchable attributes, see Section 20.2.2, “Setting Search List preferences,” on page 287.
Search	Runs the specified search criteria. For more information on defining the default search, see Section 20.2.2, “Setting Search List preferences,” on page 287.
My Saved Searches  Saved Searches	Allows the user to run, edit, or delete a select a previously saved search.
Advanced Search  Advanced Search	Like the Search button, it lets the user add rows or blocks of search criteria, but in an advanced search, they are able to specify multiple rows and multiple blocks of search criteria groupings that can be made inclusive (AND) or exclusive (OR). For more information on defining the searchable attributes, see Section 20.2.2, “Setting Search List preferences,” on page 287.

This example shows how the portlet displays (using sample data) after the search criteria *First name starts with A* is entered:



You can configure the Search List portlet to use any of these features:

User interface element	Description
Identity, Location, Organization tabs	Users click one of these tabs to see the results list displayed in different ways. For more information on formats, see Section 20.1.1, “About results list display formats,” on page 283.
My Saved Searches  Saved Searches	Allows the user to select a previously saved search.
Save Search  Save Search	Allows users to save search criteria and rerun the saved searches as needed. The searches are saved to the currently logged on user’s srvprvQueryList attribute.
Export Results  	Lets users export the search results to a different format.
Revise Search  	Lets users change the search criteria.
New Search  New Search	Lets the user define a new search.

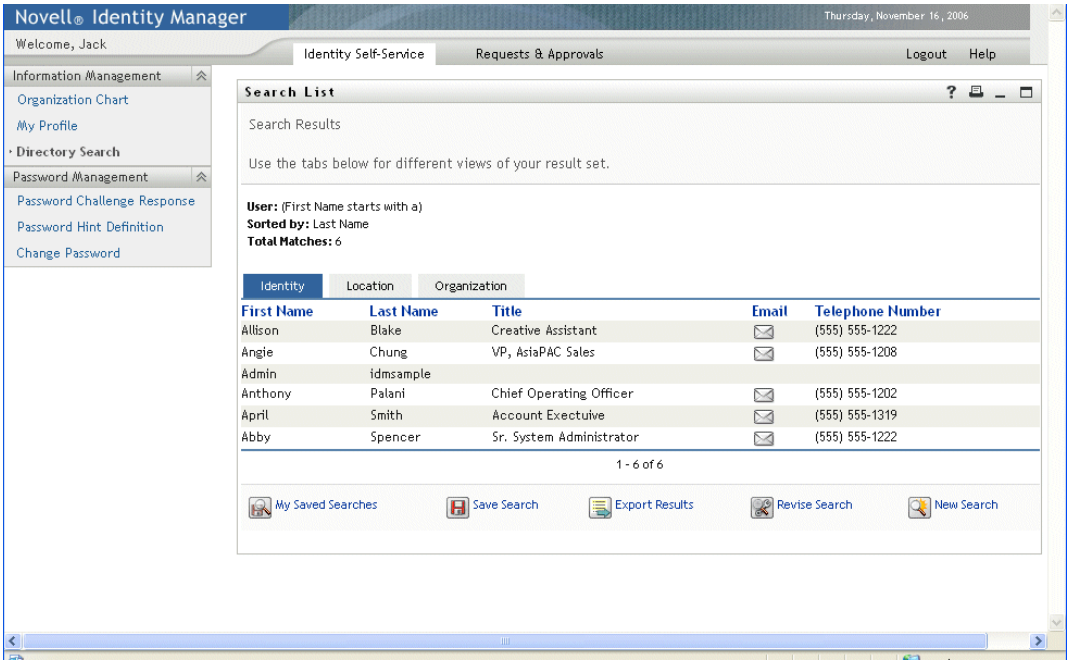
By default, Search List also allows end users to:

- Print the search results
- Launch email from the results list
- Launch the Detail portlet from the results list

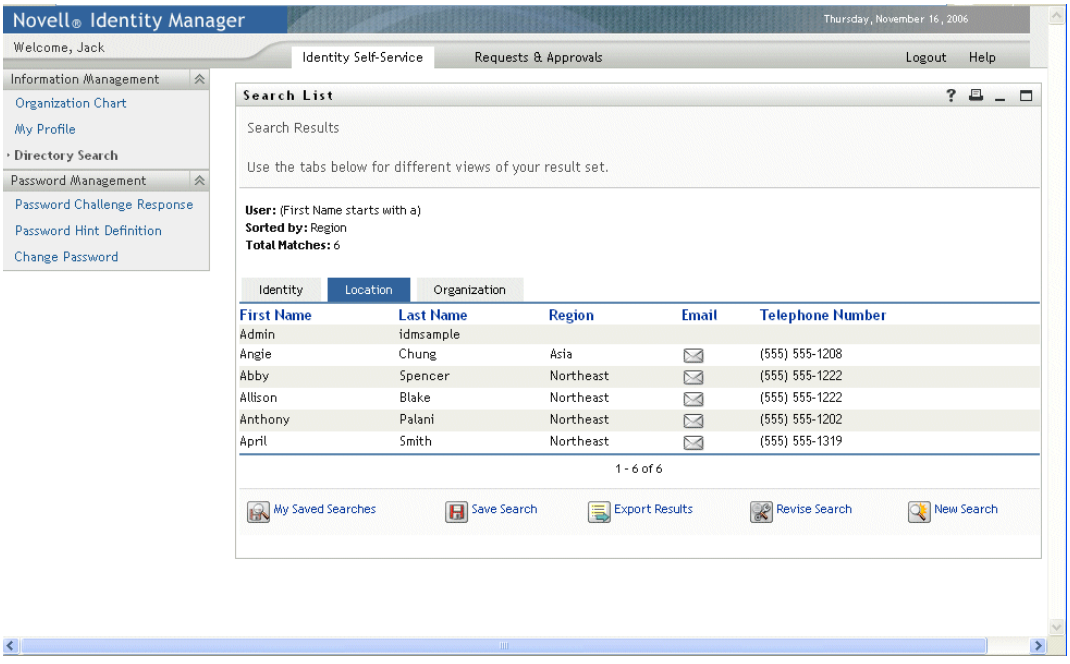
20.1.1 About results list display formats

You can define how data that is returned from the identity vault search is displayed to end users. The data can be organized in one or more of these page types:

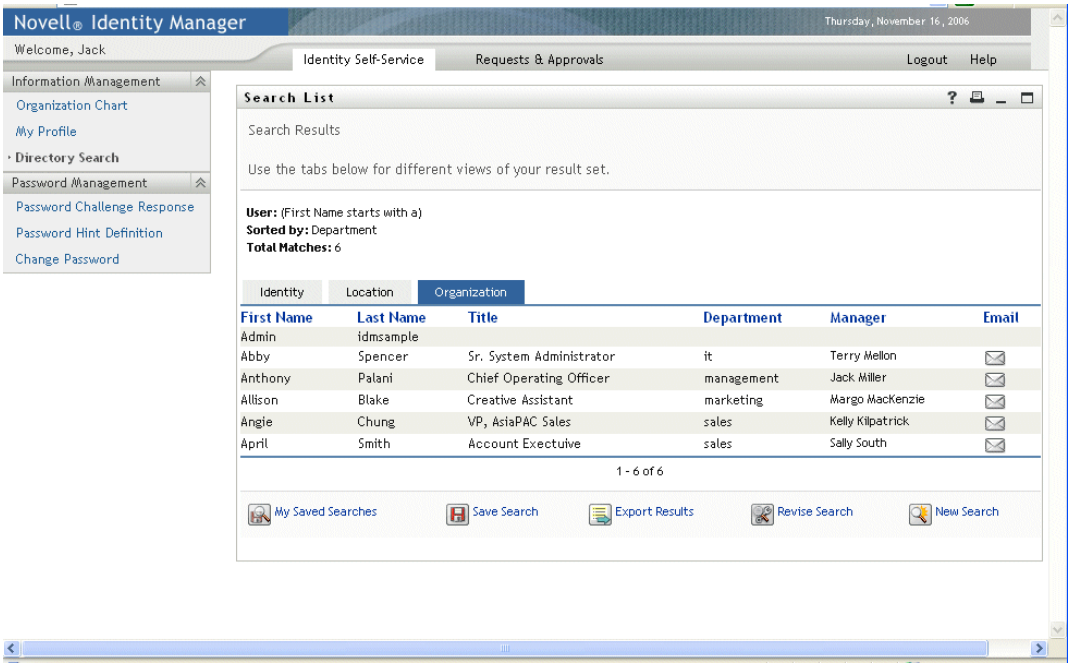
- *Identity Pages*—Typically includes contact information, as shown here:



- *Location Pages*—Typically includes location information, as shown here:



- *Organization Pages*—Typically includes organization hierarchy information, as shown here:



You can define other result list formats using the portlet's complex preferences. For example, if your identity vault schema included information about employee skills or certifications, you could set up a results list to display this information.

Depending on how you configure the portlet, end users are able to:

- Choose the types of identity vault objects to search (such as, users and groups)
- Specify the *criteria* that they want to search (such as, First name starts with, Last name includes, and so on)
- Choose the *display format* that they want to view the search results
- Change the *sort order*

20.2 Configuring the Search List portlet

To configure the Search List portlet, you'll follow a set of steps like these:

Step	Task	Description
1	Define: • The entities and attributes you will allow users to search • How you will display the results list	You can use the predefined Directory Search action that gets installed with the Identity Manager user application as-is. You can modify it, or you can create your own. For more information, see Section 20.2.2, "Setting Search List preferences," on page 287.
2	Verify that the set of entities and attributes for searching are defined in the directory abstraction layer.	For more information, see Chapter 4, "Configuring the Directory Abstraction Layer," on page 75.

Step	Task	Description
3	Determine how you want users to access the portlet.	Do you want users to launch this portlet from an existing or a new page? For more information about pages, see Chapter 7, “Page Administration,” on page 131.
4	Set preferences for the portlet	Preferences for the search list portlet let you define: • The attributes displayed for each results list format • What results list display format a search will produce • The default sort order for the results list formats For more information, see Section 20.2.2, “Setting Search List preferences,” on page 287
5	Test your settings	Verify that the results lists show the desired attributes.
6	Set eDirectory rights and establish any indexes needed to enhance performance	eDirectory rights: To execute a search • The user performing the search needs **Browse** rights to any users or objects being searched. To save a search (for non-Administrative users): • **Trustee** of the organizational unit and the organization where they will be executing the search. • **User** requires write, self, and supervisor rights. **Performance enhancement**—The performance of the search can be improved by adding an eDirectory value index to the attribute on which the search is based.

For more information on defining different results list display formats, see Section 20.2.2, “Setting Search List preferences,” on page 287.

20.2.1 Directory abstraction layer setup

The entities and attributes that can be selected from the search criteria dropdown and data returned from the identity vault searches must be defined in the directory abstraction layer. The following table shows the properties that you should set for the entities and attributes used by search list.

Definition type	Setting	Directory abstraction layer value
entity	view	Selected (true)

Definition type	Setting	Directory abstraction layer value
attribute	enable	Selected (true)
	search	Selected (true)
		When false, you cannot define a search on this attribute or include it in a results list format
	hide	Unselected (false)
		Any attribute where search is selected (true) must also have hide set to unselected (false) because the Search List portlet does not examine the value of the hide property during the search (because it hinders performance).
		Suppose that User1 sets the HomePhone attribute to hide=true (in eDirectory). HomePhone is searchable so Search List retrieves the record, but Search List does not examine the values of the other attributes (because it would impact performance). If another user searched on an exact match for the HomePhone attribute, the hidden record would be displayed in the results list.

Other Directory abstraction layer settings The directory abstraction layer data type, format type, filters, and search scope will also impact the Search List portlet. The data type and format type affect the appearance, the filter and search scope will affect how much data is returned.

For more information, see Section 4.3, “Working with entities and attributes,” on page 86.

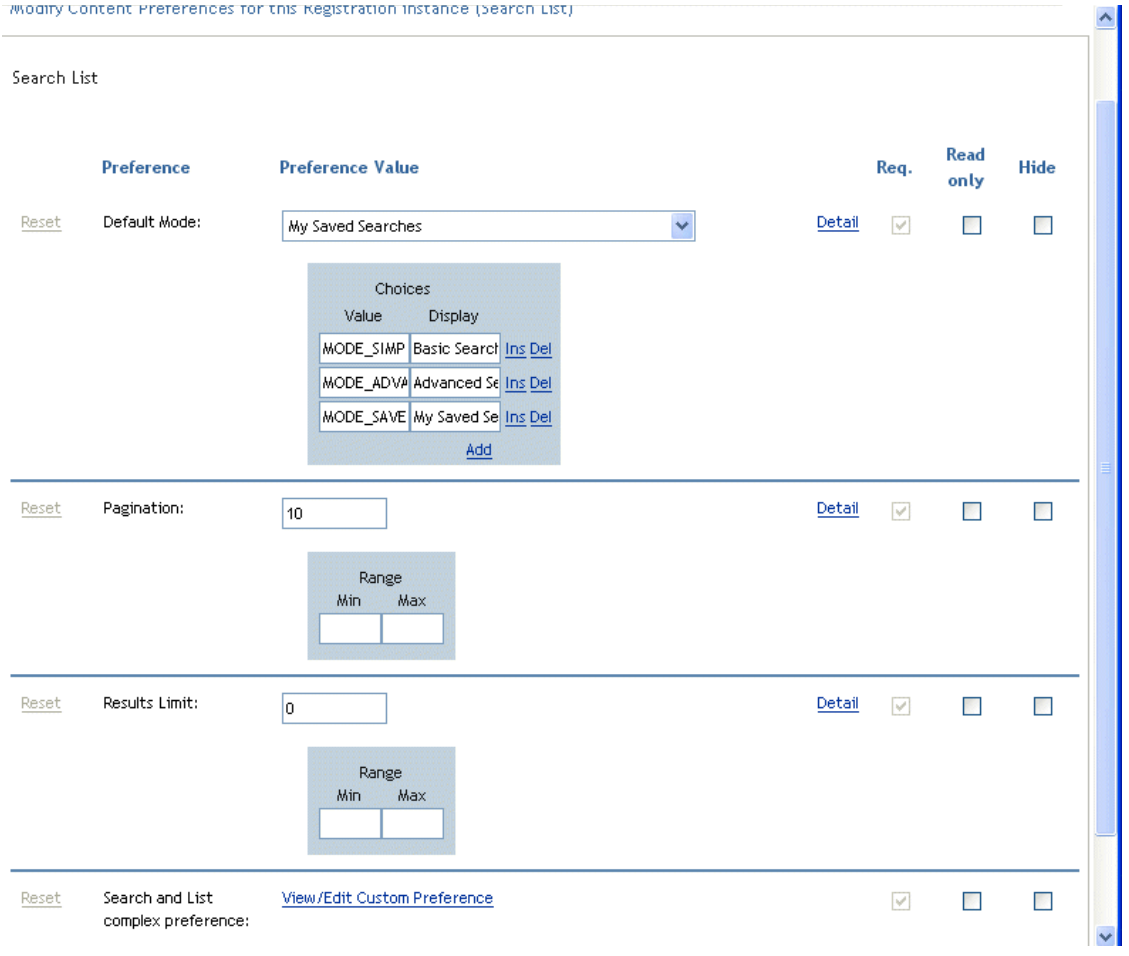
20.2.2 Setting Search List preferences

You define two types of preferences:

- “Search preferences” on page 287
- “Results List format preferences” on page 289

Search preferences

The search preferences are contained in a single preference page:



The search preferences are defined below:

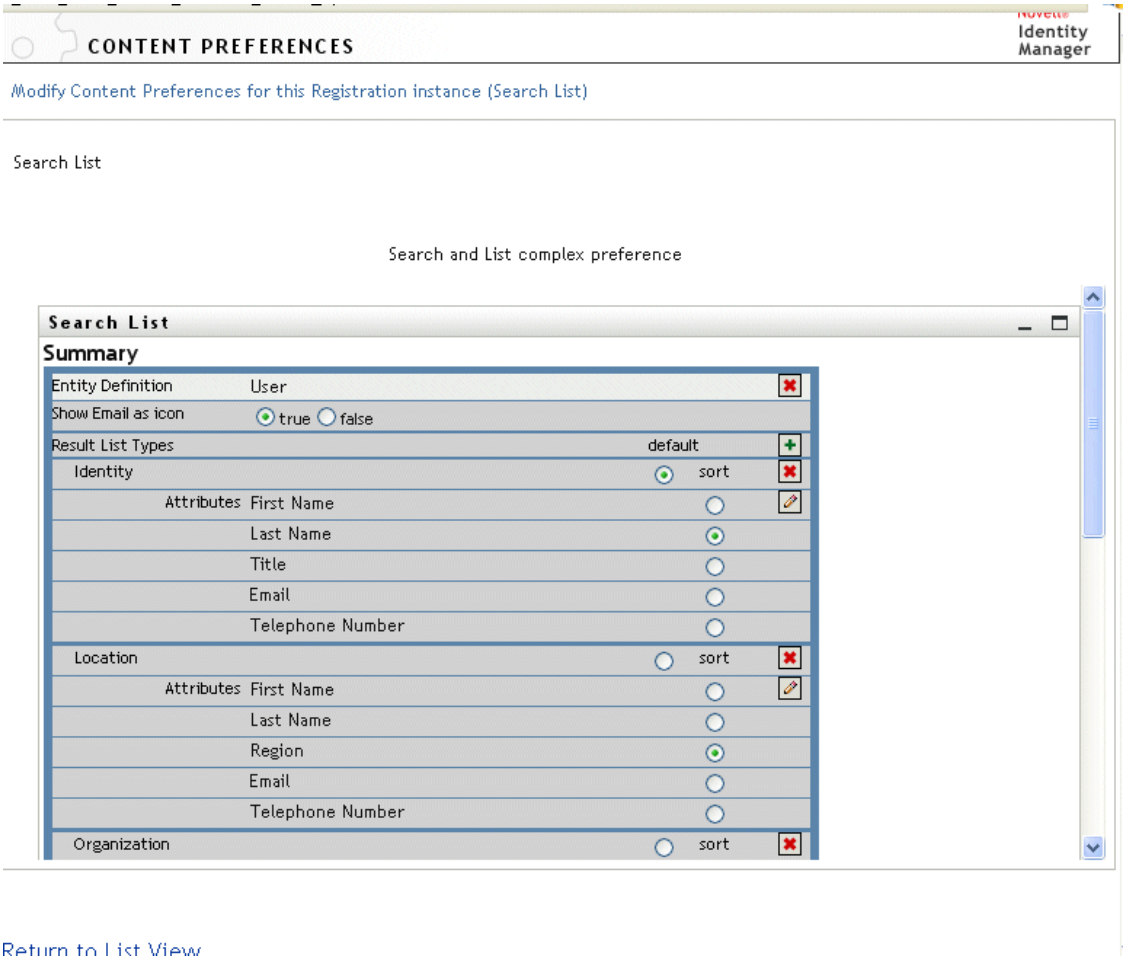
Preference	What to do

Default Mode	Specify how you want the portlet to display when a user first accesses it. Values are:
	
	Basic Search—Allows users to enter a single search criteria. For example:
	
	`First Name starts with A`
	
	Advanced search—Allows users to define multiple search criteria in one or more search blocks. Users can use the and/or logical operators within the search criteria or among the search blocks. For example, users can create a search like this:
	
	`(First Name starts with A or First Name starts with B) and (Region = Northeast or Region = Southeast)`
	
	OR
	
	`(First Name starts with A and Last Name starts with B) or (First Name starts with B and Last Name starts with A)`
	
	My Saved Searches—Displays a list of searches saved by the currently logged in user. The searches are saved in the user's srvprvQueryList attribute.
	
	NOTE: Users can access any of these modes at runtime by executing or editing a search or clicking a button at the bottom of the portlet.

Pagination	The maximum number of rows shown at a time.
Results Limit	The maximum number of matches returned by the search. If set to 0, then the maximum defers to the directory abstraction layer setting.
Search and List complex preference	Click to refine the: • Entities to search • Result set type • Attributes to include in the pages and the order in which they appear By default, any object listed in the directory abstraction layer with the attribute view=true are included in the search. The entity's attribute list is derived from the attributes listed in the directory abstraction layer and whose defined as enable=true.

Results List format preferences

The complex preferences page lets you define the entities to include in the search and how to format the results list. The default preferences page looks like this:



The complex preferences include:

Preference	What to do
Entity Definition	Each object that is valid for searching (view=true) has a corresponding Entity Definition block on this preferences page. Use these preferences to: • Define the objects included in the search. • Modify the results list format definitions (such as adding and removing the attributes that are displayed and their default sort order). • Remove any objects that you do not want included in the search by clicking the delete button shown on the Entity Definition line. This deletes the entire entity definition block. You can add the object back to the search later by clicking **Add Entity Definition** (located at the bottom of the page) and completing the wizard selection panels. **TIP:** If an object does not appear in this list, but is listed in the directory abstraction layer, check the **view** modifier (on the entity object). If set to false then the entity cannot be used by the identity portlets.
Show email as Icon	When true and an Email attribute is specified in the results list, it will display as an icon. When false the Email attribute displays the full email address. The email attribute (whether text or icon) is a clickable mailto: link.
Results List Types (default)	Specifies the results list default format for the current entity. The default is used only when a different format is not selected by the current user.
Results List display format block	Specifies the display format (such as Identity, Location, or Organizational pages) and includes the set of attributes to include for the type. **To remove a Results List Type:** • Click the delete button next to the Results List Type. This deletes the page type and all of its associated attributes from the search. **To add a result set page:** • Click the expand button and select the result set format from the list of choices.

Preference	What to do
Attributes	Specifies the set of attributes that will be displayed for the particular display format. **To add or remove an attribute:** • Click the modify attributes button. • To add an attribute, select it (from the list of Available attributes). • Click the arrow to move it to the Selected list. Do the reverse to remove an attribute from the Results List. • To reorder the attributes list, click the up and down arrows to the right of the selected list. • Click **Submit**. **Attributes and data types**—The attribute's data type affects the way it is displayed. For example, if an attribute is defined as a sub-type of local list or global list then possible values are displayed in a dropdown list box in the Basic or Advanced Search Criteria screens. If the type is DN then a finder and history button are displayed to allow users to select a value in the Basic or Advanced Search Criteria screens, and the DN will be resolved to a user-friendly display in the results list. The data type and sub-type also restrict the comparison operator displayed for the user to ensure that only valid comparisons are constructed. For more information, see Chapter 4, "Configuring the Directory Abstraction Layer," on page 75.
Results List display format block Sort	The sort order for the Results List is based on this attribute. The default sort order only takes effect if the Result Set Type is not the display format for the current user session. **Multi-valued attributes and single-valued attributes**—The number of records displayed in a results list will vary depending on whether the sort attribute is single- or multi-valued. Sorting on multi-value attributes will generally appear to result in more records although the total number of matches remains the same. This is because each value of a multi-valued attribute is shown on a line by itself.

Completing the preferences panel

To verify that you've submitted valid entries, click *Submit*. If an entry is invalid, you'll see an error message displayed at the top of the preferences page. Once you are able to resolve all of the errors, click *Return to List View*, then click *Save Preferences*.

Designing and Managing Provisioning Requests

V

These chapters tell you how to use the features of the Provisioning Module of Identity Manager.

Introduction to Workflow-Based Provisioning

21

This chapter provides an overview of workflow-based provisioning. Topics include:

21.1 About workflow-based provisioning

A key feature of Identity Manager is *workflow-based provisioning*, which is the process of managing user access to secure resources in an organization. These resources may include digital entities such as user accounts, computers, and databases. In this release, provisioned resources are mapped to Identity Manager entitlements.

Identity Manager can service a wide range of *provisioning requests*. Provisioning requests are user or system actions intended to grant or revoke access to organizational resources. They can be initiated directly by the end user through the Identity Manager user application, or indirectly in response to events occurring in the identity vault (eDirectory).

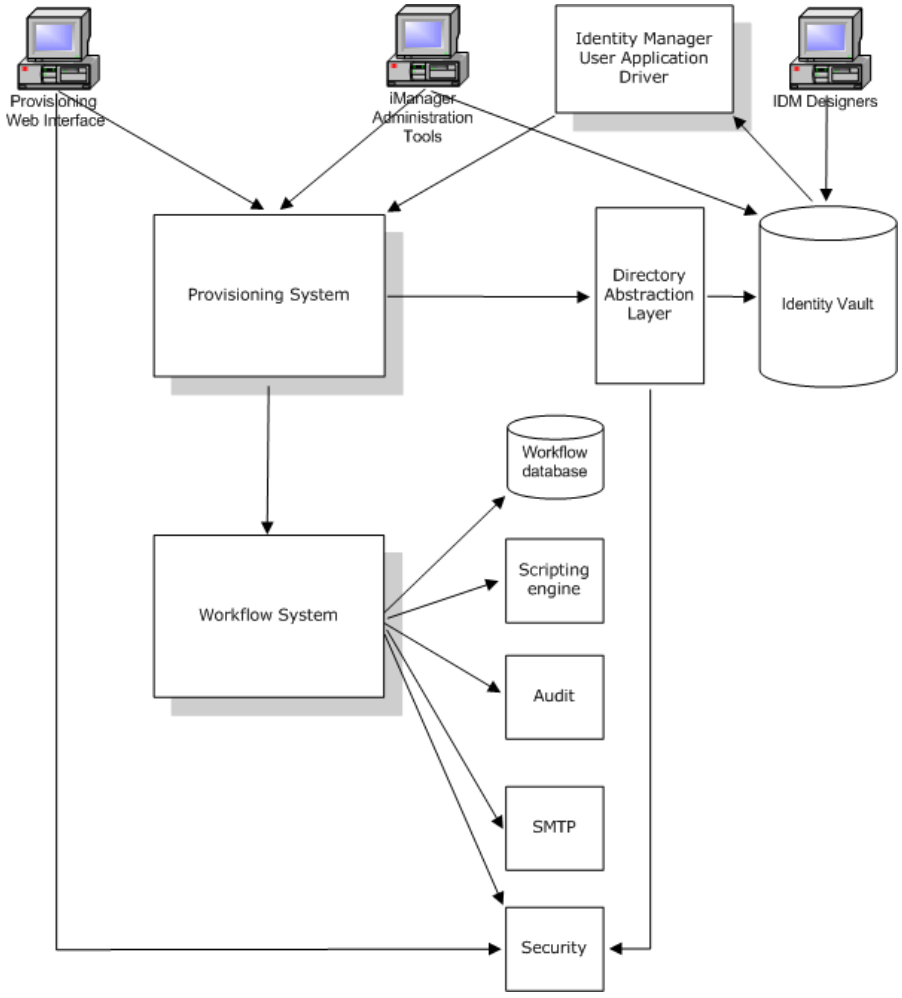
When a provisioning request requires permission from one or more individuals in an organization, the request starts a workflow. The workflow coordinates the *approvals* needed to fulfill the request. Some provisioning requests require approval from a single individual; others require approval from several individuals. In some instances, a request can be fulfilled without any approvals.

Some workflows require that processing proceed in a *sequential* fashion, with each approval step being performed sequentially. Other workflows provide support for *parallel* processing. When you define a provisioning request, you specify whether you want the workflow to support sequential or parallel processing.

Identity Manager provides a set of Web-based tools that the administrator can use to build provisioning capabilities into the user application. These tools give you the ability to configure provisioning requests and also manage workflows that are in process. To configure a provisioning request, the administrator creates a *provisioning request definition*, which binds the resource to a workflow.

21.1.1 High-level architecture

The following diagram shows the high-level architecture of the workflow-based provisioning system included with Identity Manager:



The sections below describe each component of this architecture.

Provisioning Web interface

The Identity Manager user application provides a Web interface through which end users submit provisioning requests and manage these requests once they've been submitted. The user application also provides the User Application Administrator or an Organizational Manager with the ability to assign delegates and proxies for provisioning workflows.

TIP: The provisioning and workflow actions are available on the *Requests & Approvals* tab of the Identity Manager user application.

For more information on delegates and proxies, see Section 21.3, "Provisioning security," on page 305. For complete details on working with the user application, see the *Identity Manager User Application: User Guide*.

iManager Administration Tools

iManager provides plug-ins you can use to configure and manage provisioning requests and their associated workflows.

To configure a provisioning request, you bind it to a provisioned resource, specify the runtime characteristics of the associated workflow, and enable it for use. Once a provisioning request has been initiated, you can use iManager to view the status of the workflow process, reassign activities within the workflow, or terminate the workflow in the event that it is stuck.

Identity Manager User Application Driver

In addition to supporting end user requests to provision resources, Identity Manager allows you to initiate provisioning requests in response to events occurring in eDirectory. The Identity Manager *User Application Driver* listens for events and responds by initiating the corresponding provisioning requests. These requests may in turn initiate workflows to handle the approval process. For example, if configured to do so, Identity Manager will support a scenario in which the addition of a new user in eDirectory automatically kicks off a predesignated provisioning request and workflow.

Provisioning System

The Provisioning System performs all processing required to initiate and fulfill provisioning requests. If a request requires one or more approvals, the Provisioning System in turn calls the Workflow System to start the workflow process. Once the necessary approvals have been given, the Provisioning System provisions the resource as requested.

The Provisioning System maintains information about available and outstanding provisioning requests in the identity vault (eDirectory).

To initiate a request or perform the processing required to fulfill a request, the system accesses the identity vault through the Directory Abstraction Layer.

For details on the Directory Abstraction Layer, see Chapter 4, “Configuring the Directory Abstraction Layer,” on page 75.

Workflow System

When a provisioning request requires one or more approvals, the Workflow System coordinates the approval process. During the course of processing, it interacts with these components:

- Workflow database
- Scripting engine
- Audit
- SMTP
- Security system

Workflow database

To track the state of workflows in process, the Workflow System stores information in a database. This database maintains information about workflow process instances, work lists (queues), and workflow addressees. In addition, it stores any comments added during the execution of a workflow process.

Scripting engine

The Workflow System calls the Scripting engine whenever a workflow includes a dynamic expression that must be evaluated. Dynamic expressions can include variables, functions, and operators, as well as references to entities in the Directory Abstraction Layer.

Novell Audit

To log information about the state of a workflow process, the Workflow System interacts with Novell Audit. During the course of its processing, a workflow may log information about various events that have occurred. Users can then use the Novell Audit reporting tools to look at logging data.

For details on setting up logging, see Chapter 5, “Setting up Logging,” on page 115. For details on controlling the levels of logging messages you want the Identity Manager user application to generate, see Chapter 12, “Logging Configuration,” on page 201.

SMTP

A workflow process often sends e-mail notifications at various points in the course of its execution. For example, an e-mail might be sent when a workflow activity is assigned to a new addressee.

An administrator can edit an e-mail template in iManager and then use this template in a workflow process. At runtime, the Workflow System retrieves it from eDirectory and replaces tags with dynamic text suitable for the notification.

E-mail notifications are handled through the Simple Mail Transfer Protocol (SMTP).

For basic setup steps required for e-mail notification, see Section 23.3, “Configuring the e-mail server,” on page 339 and Section 23.4, “Working with the installed e-mail template,” on page 340. For details on configuring e-mail notification for a workflow, see “Configuring the workflow activities” on page 320.

Security

The Security system handles all aspects of security for a workflow-based provisioning application.

For more information on workflow security, see Section 21.3, “Provisioning security,” on page 305.

21.1.2 Provisioning and workflow example

Suppose a user needs an account on an IT system. To set up the account, the user initiates a request through the Identity Manager user application. This request starts a workflow, which coordinates an approval process. Once the necessary approvals have been granted, the request is fulfilled. There are three basic steps in the process, as outlined below.

Step 1: Initiating the request

In the Identity Manager user application, the user browses a list of resources by *category* and selects one to provision. In the identity vault, the *provisioned resource* selected is associated with a *provisioning request definition*. The provisioning request definition is the most prominent object in a provisioning system. It binds a provisioned resource to a *workflow*, and acts as the means by which the workflow process is exposed to the end user. The provisioning request definition provides all the

information required to display the *initial request form* to the user, and to start the flow that follows the initial request.

In this example, the user selects the New Account resource. When the user initiates the request, the Web application retrieves the initial request form and the description of the associated *initial request data* from the Provisioning System, which gets these objects from the provisioning request definition.

When a provisioning request is initiated, the Provisioning System tracks the initiator and the recipient. The *initiator* is the person who made the request. The *recipient* is the person for whom the request was made. In some situations, the initiator and the recipient may be the same individual.

Each provisioning request has an *operation* associated with it. The operation specifies whether the user wants to *grant* or *revoke* the resource.

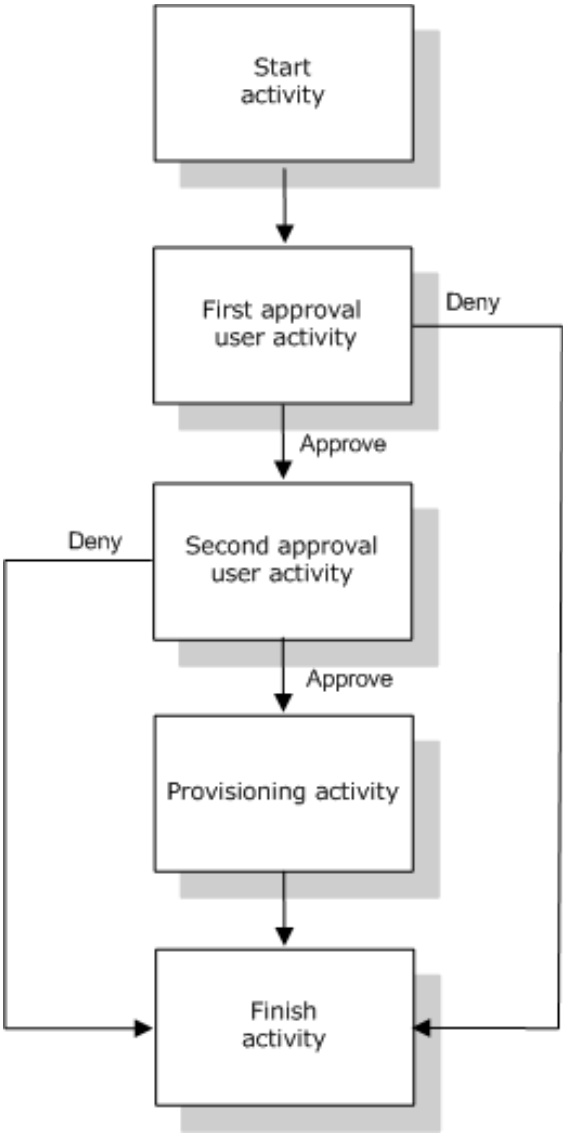
Step 2: Approving the request

Once the user has initiated a request, the Provisioning System starts the workflow process. The *workflow process* coordinates the approvals. In this example, two levels of approvals are required, one from the user's manager, and a second from the manager's supervisor. If approval is denied by any user in a workflow, the flow terminates and the request is denied.

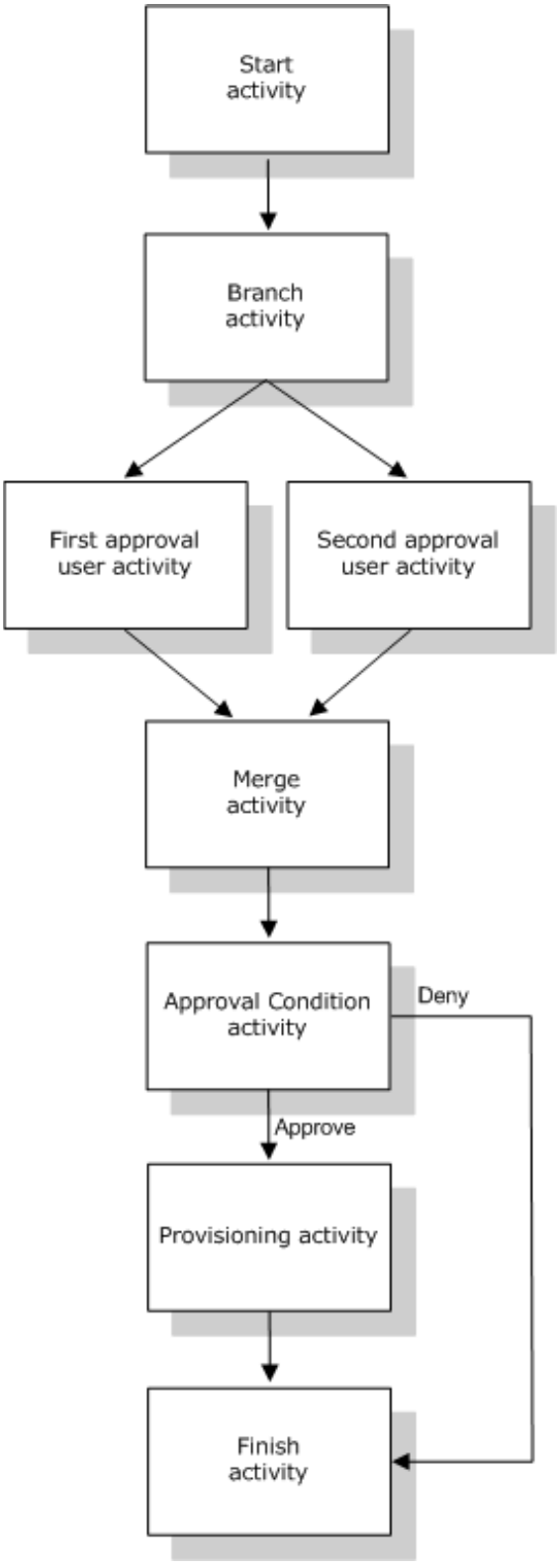
NOTE: Identity Manager ships with a set of provisioning request templates that support up to five levels of workflow approval. In a follow-on release of Identity Manager, the Eclipse-based design environment will provide tools that let you create your own custom workflow processes. For more information on the templates that ship with this release, see Section 22.2, "Working with the installed templates," on page 310.

Workflows can process approvals in a sequential manner, or in a parallel manner. In a *sequential workflow*, each approval task must be processed before the next approval task begins. In a *parallel workflow*, users can work on the approval tasks simultaneously.

Sequential flow Here's the basic design pattern of a sequential workflow with two approvals:



Parallel flow Here's the basic design pattern of a parallel workflow with two approvals:



NOTE: The display labels (First approval, Second approval, and so on) can easily be changed to suit your application requirements. For parallel flows, you may want to specify labels that do not

imply sequential processing. For example, you might want to assign labels such as One of Three Parallel Approvals, Two of Three Parallel Approvals, and so on.

The workflow definition is made up of these components

Process components	Description
Activities	An activity is an object that represents a task. An activity can present information to the user and respond to user interactions, or perform background functions that are not visible to the user. In the workflow examples shown above, the activities are represented by boxes. In the Identity Manager user application, the user activities that handle the approval process are referred to as **tasks**. An end user can see the list of tasks in his/her queue by clicking **My Tasks** in the **My Work** group of actions. To see which workflow activities have been processed for a particular task, the user can select the task and click the **View Comment History** button on the Task Detail form. To see which workflow activities have been processed for a particular provisioning request, the user can click **My Requests**, select the request, and click the **View Comment and Flow History** button on the Request Detail form. For more information on the **My Tasks** and **My Requests** actions, see the *Identity Manager User Application: User Guide*.
Links	Links are what tie the activities in a workflow together. A link represents a path to be followed between two activities. An activity can have multiple incoming links and multiple outgoing links. When an activity has more than one outgoing link, the link selected depends on the **outcome** of the activity. The outcome is the end result of processing performed by the activity. For example, a User activity may have an outcome of approved or denied, depending on the **action** taken by the user. In the workflow examples shown above, the links are represented by arrows.

Start activity The workflow process begins with the execution of the *Start activity*. This activity initializes a work document using the initial request data. It also binds several system values such as the initiator and recipient, so that these can be used in script expressions.

User activities After the Start activity finishes execution, the Workflow System forwards processing to the first *User activity* in the flow. A User activity is an activity that supports user interactions. To handle these interactions, the activity displays a form, which gives the user the ability to act on the request. In the workflow examples shown above, `First approval` and `Second approval` are examples of User activities. The display labels for User activities can be localized to satisfy international requirements.

A User activity may support one or more of these *actions*:

- Claim
- Approve

- Deny
- Refuse
- Reassign (available to Organizational Managers and User Application Administrators only)

NOTE: The fields and buttons on the form will vary depending on which resource is requested and how the workflow is configured. The *Refuse* action, for example, is not supported by many of the templates shipped with the product.

A User activity has five possible *outcomes*:

- Approved
- Denied
- Refused
- Error
- Timeout

NOTE: The Error and Timeout outcomes may occur without any action being taken by the user.

If the user approves the request, the workflow forwards control to the next activity in the flow. If no further approvals are needed, the resource is provisioned. If the user denies the request, the work item is forwarded to the next activity in the workflow and the request is denied. Alternatively, the user can reassign the task (if he/she is an Organizational Manager or User Application Administrator), which puts the work item in another user's queue.

NOTE: The provisioning request templates that ship with the product are configured to terminate a workflow process when a request is denied. When a request is denied, the workitem is forwarded to the Finish activity, which terminates the flow.

The user to whom a User activity has been assigned is referred to as the *addressee*. The addressee for an activity may be notified of the assigned task via e-mail. To perform the work associated with the activity, the addressee can click the URL in the e-mail, find the task in the work list (queue), and claim the task.

The addressee must respond to a User activity within a specified amount of time, or the activity times out. Typically the *timeout interval* is expressed in hours or days, to allow the user sufficient time to respond.

When an activity times out, the workflow process may try to complete the activity again, depending on the *retry count* specified for the activity. In some situations, the workflow process may be configured to escalate an activity that has timed out to another user. In this case, the activity is reassigned to a new addressee (the user's manager, for example) to give this user an opportunity to finish the work of the activity. In the event that the last retry times out, the activity may be marked as approved or denied, depending on how the workflow was configured.

Conditional activities During the course of execution, a workflow process may perform a test and check the outcome to see what to do next. The *Conditional activity* provides this capability. Conditional activities use a scripting expression to define the condition to evaluate. In the workflow examples shown above, `Approval Condition` is an example of a Conditional activity.

Conditional activities support three possible *outcomes*:

- True
- False
- Error

Branch and Merge activities In a workflow that supports parallel processing, the *Branch activity* allows two users to act on different areas of the workitem in parallel. Once the users have completed their work, the *Merge activity* synchronizes the incoming branches in the flow.

Provisioning activity The *Provisioning activity* fulfills the provisioning request. This activity is executed only if all of the necessary approvals have given.

For details on the provisioning step, see “Step 3: Fulfilling the request” on page 304.

Finish activity The *Finish activity* is the final activity in a workflow. When all the activities in a flow have been completed and the final result of the flow is available, the Finish activity can be executed. The Workflow System can determine the final state of the process by examining the links into the Finish activity. The overall flow state is *approved* when an approval link reaches the Finish activity. If any other outcome (deny, timeout, or error) leads into the Finish activity, the overall flow state is *denied*.

When a workflow process reaches the Finish activity with an approved state, the approval process is complete, and the provisioning request can be fulfilled.

Step 3: Fulfilling the request

When a provisioning request has been approved, the Workflow System can begin the *provisioning* step. At this point, control passes back to the Provisioning System.

To fulfill the provisioning request, the Provisioning System may execute an Identity Manager entitlement or directly manipulate an eDirectory object and its attributes. During the provisioning step, it creates any related objects and records the results of the provisioning action on the recipient, as described in the provisioning data definition. Depending on whether the user requested a grant or revoke operation, this may involve setting or removing the value of an attribute on the recipient, or adding an item to or removing an item from a multi-valued attribute on the recipient. The attributes involved are eDirectory attributes (possibly made available by adding an auxiliary class to the recipient). The attribute values themselves may be simple or they may be of a complex type that allows the Provisioning System to specify the value of internal sub-attributes.

21.2 Provisioning configuration and administration

To configure a provisioning request definition, you use iManager to bind it to a provisioned resource, specify the runtime characteristics of the associated workflow, and enable it for use. Identity Manager ships with a set of predeployed provisioning request definitions and workflows. You can use these as *templates* for building your own provisioning system. The installed templates are easy to use and, yet, flexible enough to address the requirements of a wide range of business environments. To set up your system, you define new objects based on the installed templates and customize these objects to suit the needs of your organization.

Once a provisioning request definition has been configured, you can use iManager to view the status of running workflow processes, reassign activities within the workflows, or terminate a workflow in the event that it is stuck.

For more information about using iManager for provisioning configuration and management, see Chapter 22, “Configuring Provisioning Request Definitions,” on page 309 and Chapter 23, “Managing Provisioning Workflows,” on page 331.

21.3 Provisioning security

When a user logs into the Identity Manager user application, the Security system authenticates that user and sets access controls to protect provisioning and workflow objects from unauthorized use. This ensures that the user sees only those provisioning request definitions to which he or she has been granted access. In addition to performing authentication and authorization services for the user application, the Security system manages proxy and delegate assignments.

- A *delegate* is a user authorized to perform work for another user. A delegate assignment applies to a particular provisioning request definition.
- A *proxy* is a user authorized to perform any and all work for one or more users, groups, or containers. Unlike delegate assignments, proxy assignments are independent of provisioning request definitions, and therefore apply to all work and settings.

If logging is enabled, any actions taken by a proxy or delegate are logged along with actions taken by other users. When an action is taken by a proxy or delegate, the log message clearly indicates that the action was performed by a proxy or delegate for another user. In addition, each time a new proxy or delegate assignment is defined, this event is logged as well.

If a provisioning request definition is configured to generate e-mail notifications, proxies as well as addressees are notified by e-mail. Delegates are not included in e-mail notifications.

Workflow security roles The Security system recognizes the following security roles:

Role	Description	Rights
User Application Administrator	Locksmith user with full administrative rights.	The User Application Administrator is permitted to perform these tasks in **iManager**: • Configure provisioning requests • Manage workflows already in process The User Application Administrator is permitted to perform these tasks within the **user application**: • View and edit all tasks in all workflow queues. • Define proxy and delegate assignments for any user in the system. • View hidden information (hidden attributes) for any user in the system. • Create Task Group Managers and assign them to groups. The User Application Administrator is the only user who can create and assign Task Group Managers. **NOTE:** The **Administration** tab of the Identity Manager user application provides tools for assigning rights to administer the user application. To use this tab, you must first log on as the user who was specified as the User Application Administrator at installation time. For details on using the security features of the user application, see Chapter 11, “Security Configuration,” on page 197.

Role	Description	Rights
Organizational Manager	Direct report supervisor for an employee. Each user has only one Organizational Manager. **TIP:** The Organizational Manager can also be thought of as an administrative manager.	The Organizational Manager is permitted to: • View all tasks that are in his/her team's workflow queues. This capability applies to a single level in the management hierarchy; therefore, an Organizational Manager's supervisor cannot see the tasks of the Organizational Manager's direct reports. • Edit tasks for direct reports, except in the case where a direct report has a task assigned to a group whose Task Group Manager is someone other than the Organizational Manager. In this case, the Organizational Manager can view the task, but not perform any edits. Upon escalation, the task moves to the Task Group Manager, not the Organizational Manager. • Claim tasks and unclaim tasks, and reassign tasks to members of his/her team. • Define proxy and delegate relationships for himself or herself and for members of his/her team. • View hidden attributes for members of his/her team.
Task Group Manager	User given responsibility for a set of tasks associated with a task group. A task group is an extension of the LDAP Group object. Each task group can have only one Task Group Manager. Task Group Managers are assigned by the User Application Administrator. When a task is assigned to a group, the srvrprvTaskManager attribute for the group contains the DN for the user who is the designated Task Group Manager. For improved performance, Task Group Managers are also identified by an attribute on the user object. The srvprvIsTaskManager attribute is set to true for a user who is a designated Task Group Manager.	The Task Group Manager is permitted to: • View and edit all tasks that are assigned to a group for which he/she is the designated leader. The Task Group Manager is **not** permitted to: • Create resources or retract requests. • Define proxy or delegate relationships. • View hidden attributes for members of his/her team.

NOTE: Any user can view hidden attributes associated with his/her own identity.

Defining proxy and delegate relationships To define a proxy assignment for a user, you use the *Team Proxy Assignments* page on the *Requests & Approvals* tab of the Identity Manager user interface. To define a delegate assignment for a user, you use the *Team Delegate Assignments* page, which is also available on the *Requests & Approvals* tab.

Creating Task Group Managers To define a Task Group Manager for a task group, you use the *Create User or Group* page on the *Identity Self-Service* tab of the Identity Manager user interface.

For complete details on defining Task Group Managers, proxies, and delegates, see the *Identity Manager User Application: User Guide*.

Configuring Provisioning Request Definitions

22

This chapter provides instructions for configuring provisioning request definitions. Topics include:

22.1 About the Provisioning Request Configuration plug-in

To configure a provisioning request definition, you need to use the Provisioning Request Configuration plug-in to iManager. This plug-in lets you bind the provisioning request definition to a provisioned resource, specify the runtime characteristics of the associated workflow, and enable it for use. In this release, provisioned resources are mapped to Identity Manager entitlements.

NOTE: You can also run provisioning request definitions that map directly to attributes in the identity vault. However, the installed templates do not support this type of resource, since they are based on entitlements.

You can find the Provisioning Request Configuration plug-in in the *Identity Manager category* in iManager. The plug-in includes the *Provisioning Requests task* in the *Provisioning Request Configuration role*. The Provisioning Requests task consists of these panels:

Panel	Description
Provisioning Driver Selection	Gives you the opportunity to select an Identity Manager user application driver. The driver contains a set of predeployed provisioning request definitions, so you need to pick a driver before you can begin configuring your provisioning requests.
Provisioning Request Configuration	Provides tools that let you: • Browse the available provisioning request definitions and select one to configure • Create a new provisioning request definition based on an existing definition • Set the properties of a provisioning request definition • Assign the provisioning request definition to a provisioned resource • Edit the addressee and timeout settings for each activity in the associated workflow When you choose to create a new provisioning request or edit an existing one, the plug-in runs the **Provisioning Request Configuration Wizard**.

22.2 Working with the installed templates

Identity Manager ships with a set of predeployed provisioning request definitions and workflows. You can use these as *templates* for building your own provisioning system.To set up your system, you define new objects based on the installed templates and customize these objects to suit the needs of your organization.

The installed templates let you determine the number of approval steps required for the request to be fulfilled. You can configure a provisioning request to require:

- No approvals
- One approval step
- Two approval steps
- Three approval steps
- Four approval steps
- Five approval steps

You can also specify whether you want to support sequential or parallel processing, and whether you want to approve or deny the request in the event that the workflow times out during the course of processing.

For more information on workflow design patterns, see Section 21.1.2, “Provisioning and workflow example,” on page 298.

Identity Manager ships with these templates:

Template	Description
Self Provision Approval	Allows a provisioning request to be fulfilled without any approvals.
One Step Approval (Timeout Approves)	Requires a single approval for the provisioning request to be fulfilled. If an activity times out, the the activity approves the request and the workitem forwards to the next activity.
Two Step Sequential Approval (Timeout Approves)	Requires two approvals for the provisioning request to be fulfilled. If an activity times out, the the activity approves the request and the workitem forwards to the next activity. This template supports sequential processing.
Three Step Sequential Approval (Timeout Approves)	Requires three approvals for the provisioning request to be fulfilled. If an activity times out, the the activity approves the request and the workitem forwards to the next activity. This template supports sequential processing.
Four Step Sequential Approval (Timeout Approves)	Requires four approvals for the provisioning request to be fulfilled. If an activity times out, the the activity approves the request and the workitem forwards to the next activity. This template supports sequential processing.

Template	Description
Five Step Sequential Approval (Timeout Approves)	Requires five approvals for the provisioning request to be fulfilled. If an activity times out, the the activity approves the request and the workitem forwards to the next activity. This template supports sequential processing.
One Step Approval (Timeout Denies)	Requires a single approval for the provisioning request to be fulfilled. If an activity times out, the the workflow denies the request. This template supports sequential processing.
Two Step Sequential Approval (Timeout Denies)	Requires two approvals for the provisioning request to be fulfilled. If an activity times out, the the workflow denies the request. This template supports sequential processing.
Three Step Sequential Approval (Timeout Denies)	Requires three approvals for the provisioning request to be fulfilled. If an activity times out, the the workflow denies the request. This template supports sequential processing.
Four Step Sequential Approval (Timeout Denies)	Requires four approvals for the provisioning request to be fulfilled. If an activity times out, the the workflow denies the request. This template supports sequential processing.
Five Step Sequential Approval (Timeout Denies)	Requires five approvals for the provisioning request to be fulfilled. If an activity times out, the the workflow denies the request. This template supports sequential processing.
Two Step Parallel Approval (Timeout Approves)	Requires two approvals for the provisioning request to be fulfilled. If an activity times out, the the activity approves the request and the workitem forwards to the next activity. This template supports parallel processing.
Three Step Parallel Approval (Timeout Approves)	Requires three approvals for the provisioning request to be fulfilled. If an activity times out, the the activity approves the request and the workitem forwards to the next activity. This template supports parallel processing.
Four Step Parallel Approval (Timeout Approves)	Requires four approvals for the provisioning request to be fulfilled. If an activity times out, the the activity approves the request and the workitem forwards to the next activity. This template supports parallel processing.

Template	Description
Five Step Parallel Approval (Timeout Approves)	Requires five approvals for the provisioning request to be fulfilled. If an activity times out, the the activity approves the request and the workitem forwards to the next activity. This template supports parallel processing.
Two Step Parallel Approval (Timeout Denies)	Requires two approvals for the provisioning request to be fulfilled. If an activity times out, the the workflow denies the request. This template supports parallel processing.
Three Step Parallel Approval (Timeout Denies)	Requires three approvals for the provisioning request to be fulfilled. If an activity times out, the the workflow denies the request. This template supports parallel processing.
Four Step Parallel Approval (Timeout Denies)	Requires four approvals for the provisioning request to be fulfilled. If an activity times out, the the workflow denies the request. This template supports parallel processing.
Five Step Parallel Approval (Timeout Denies)	Requires five approvals for the provisioning request to be fulfilled. If an activity times out, the the workflow denies the request. This template supports parallel processing.

Workflows and provisioned resources Each of these provisioning request definitions has a preconfigured binding to a workflow and a provisioned resource. You can change the provisioned resource associated with the request definition, but not the workflow or its topology.

Categories for provisioning requests Each provisioning request template is also bound to a *category*. Categories provide a convenient way to organize provisioning requests for the end user. The default category for all provisioning request templates is *Entitlements*. The category key, which is the value of the srvprvCategoryKey attribute, is *entitlements* (lower case).

You can create your own categories by using the directory abstraction layer editor. When you create a new category, make sure the category key (the value of srvprvCategoryKey) is lower case. This is necessary to ensure that categories work properly in the Identity Manager user application.

For details on creating provisioning categories, see Section 4.4, “Working with lists,” on page 101.

22.3 Configuring a provisioning request definition

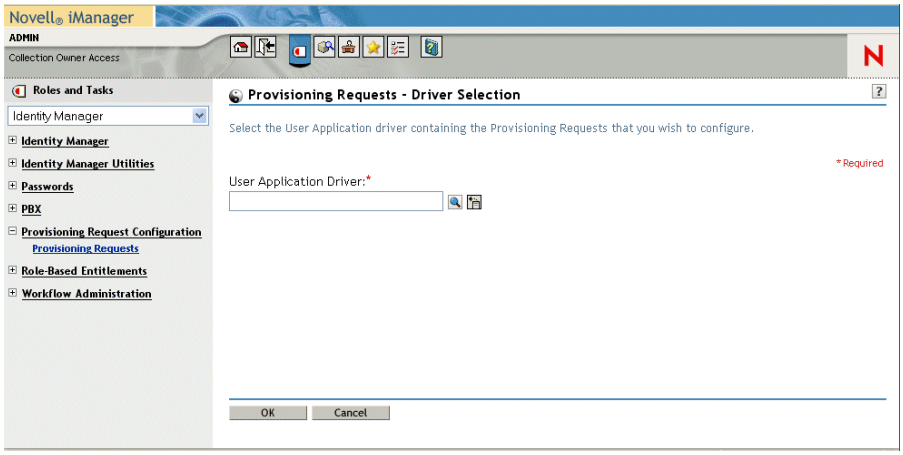
Before configuring a provisioning request definition, you need to select the Identity Manager user application driver that contains the definition. Having selected the driver, you can create a new provisioning request definition or edit an existing definition. You can also delete provisioning request definitions, change the status of a request definition, or define rights for a request definition.

22.3.1 Selecting the driver

To select an Identity Manager user application driver:

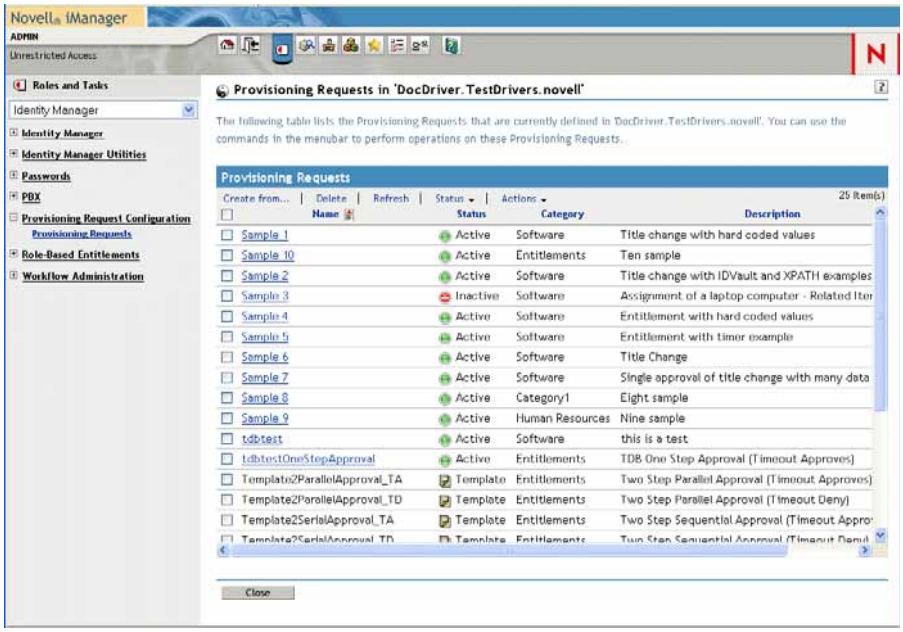
1. Select the *Identity Manager* category in iManager.
2. Open the *Provisioning Request Configuration* role.
3. Click on the *Provisioning Requests* task.

   iManager displays the User Application Driver screen.



4. Specify the driver name in the *User Application Driver* field and click *OK*.

   iManager displays the Provisioning Request Configuration panel. The Provisioning Request Configuration panel displays a list of available provisioning request definitions.



   The installed templates appear in dark text with a status of *Template*. Request definitions that are templates do not display hypertext links because they are read only.

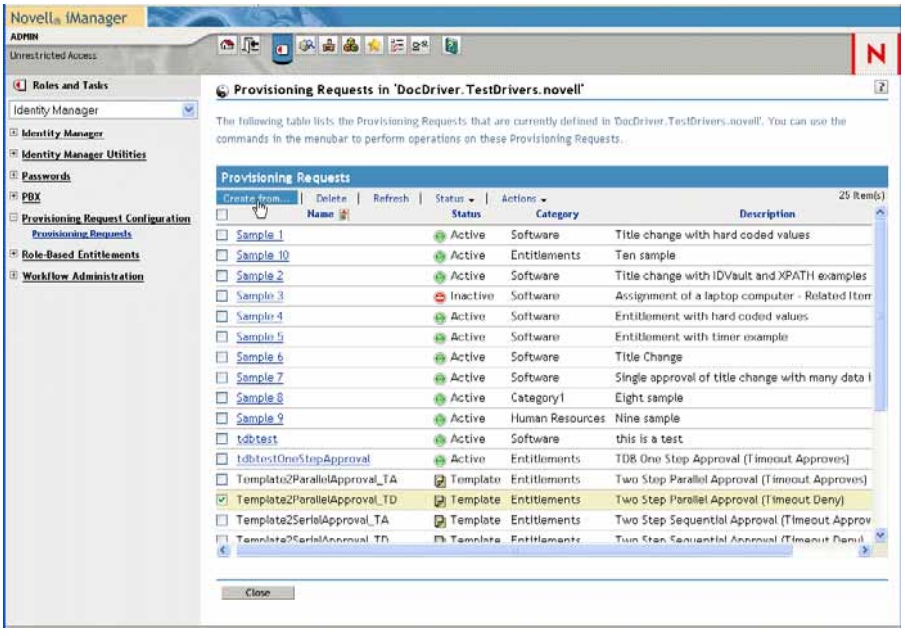
NOTE: If the request definitions were configured to use localized text, the names and descriptions for these definitions show text that is suitable for the current locale.

Changing the driver Once you've selected a driver, the driver selection remains in effect for the duration of your iManager session, unless you select a new driver. To select a new driver, click the *Actions* command and choose *Select User Application Driver* from the *Actions* menu.

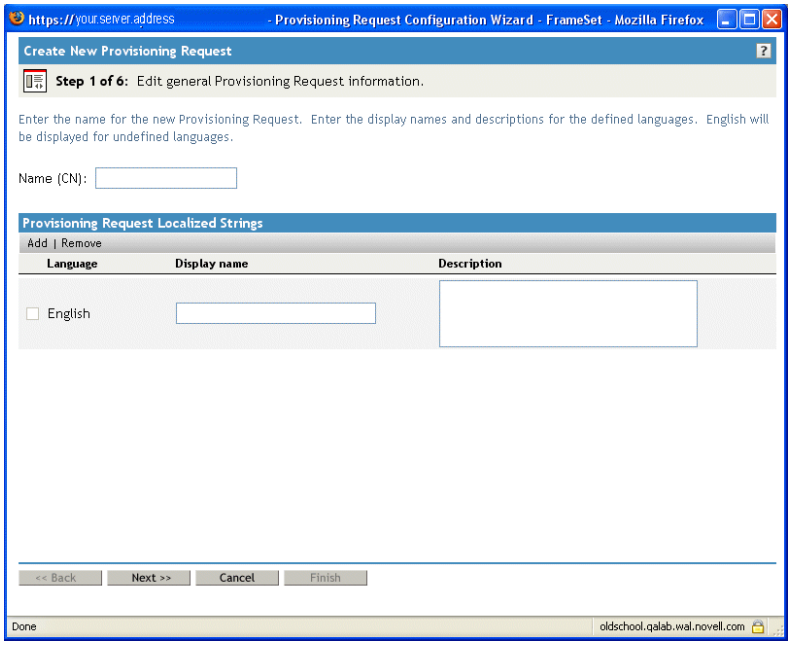
22.3.2 Creating or editing a provisioning request

To create a new provisioning request:

1. Click on the name of the provisioning request you want to use as a template in the Provisioning Request Configuration panel.
2. Click the *Create From* command in the Provisioning Request Configuration panel.



The first page of the Configure New Provisioning Request wizard displays.



3 Type a common name for the new object in the *Name* field.

4 For each language you want to support in your application, type the localized text in the *Display Name* and *Description* fields under *Provisioning Request Localized Strings*. This text will be used to identify the provisioning request throughout the user application.

5 To add a new language to the list, click *Add* and select the desired language.

NOTE: By default, a newly created provisioning request supports only English.

---

6 Click *Next*.

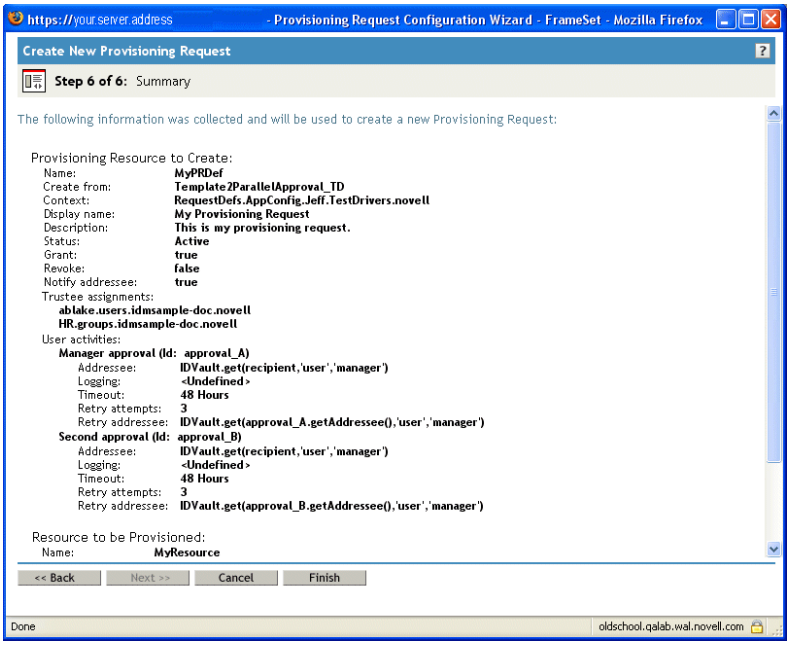
7 Specify the provisioned resource for the request definition, as described in "Specifying the provisioned resource" on page 317.

8 Configure the activities for the workflow associated with the request definition, as described in "Configuring the workflow activities" on page 320.

9 Specify the access rights for the request definition, as described in "Specifying the access rights for the provisioning request" on page 325.

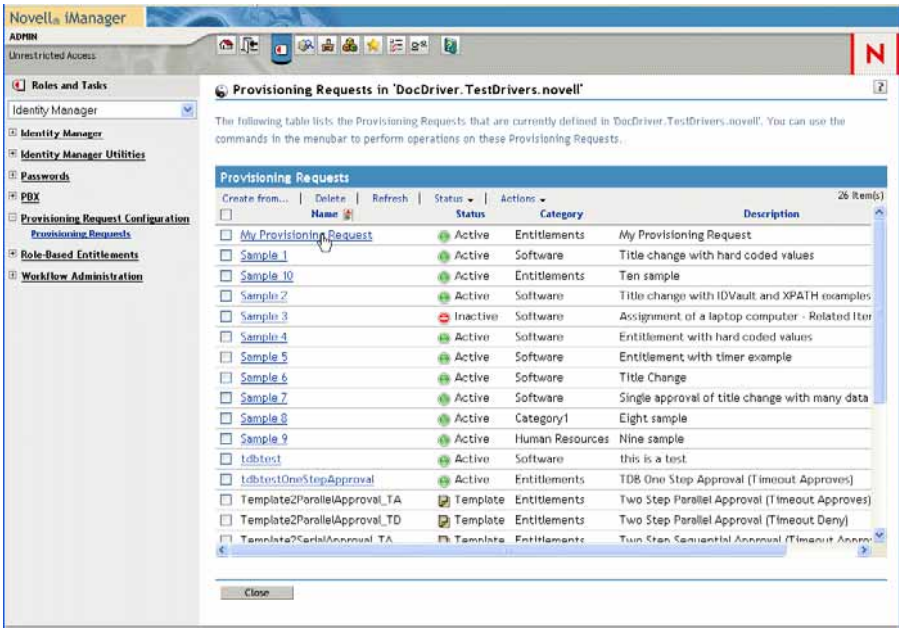
10 Specify the initial status for the request definition, as described in "Specifying the initial status of the provisioning request" on page 325.

11 Review your settings and click *Finish*.



To edit an existing provisioning request:

1 Click on the name of the provisioning request in the Provisioning Request Configuration panel.



You are not permitted to edit a provisioning request that is a template. Request definitions that have a status of Template do not display hypertext links because they are read only.

NOTE: If you have a large number of request definitions, you may want to sort the list by a particular column, such as the Name or Description. To sort by a particular column, simply click on the column heading.

---

2 For each language you want to support in your application, click the check box beside the language in the list under *Provisioning Request Localized Strings*, and type the localized text in

the *Display Name* and *Description* fields. This text will be used to identify the provisioning request throughout the user application.

3 To add a new language to the list, click *Add* and select the desired language.

NOTE: By default, a newly created provisioning request supports only English.

---

4 Click *Next*.

5 Specify the provisioned resource for the request definition, as described in "Specifying the provisioned resource" on page 317.

6 Configure the activities for the workflow associated with the request definition, as described in "Configuring the workflow activities" on page 320.

7 Specify the access rights for the request definition, as described in "Specifying the access rights for the provisioning request" on page 325.

8 Specify the initial status for the request definition, as described in "Specifying the initial status of the provisioning request" on page 325.

9 Review your settings and click *Finish*.

Specifying the provisioned resource

This section provides instructions for specifying a provisioned resource that is based on an entitlement. It does not provide conceptual information about entitlements or instructions for creating and using entitlements.

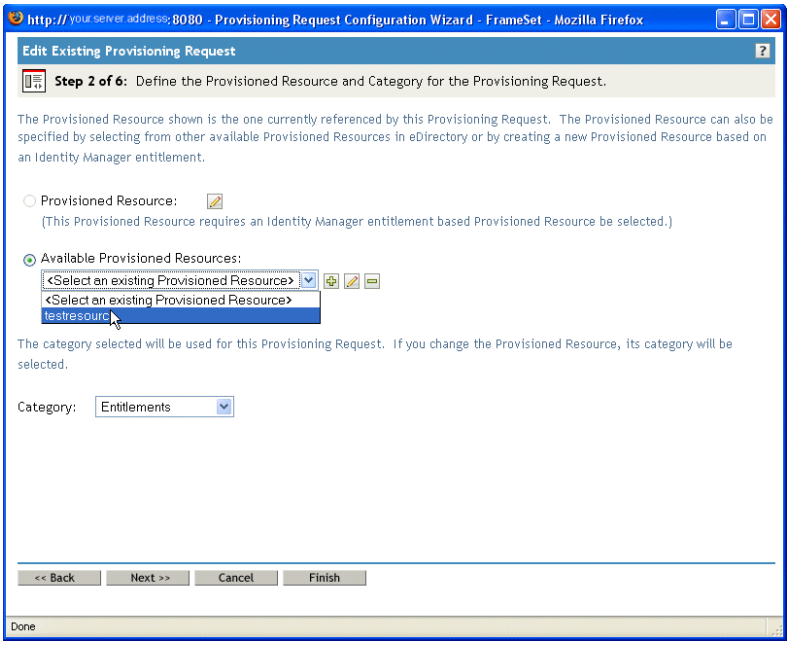
For complete details on entitlements, see the *<z-DocTitleInVariable>Novell Identity Manager: Administration Guide*.

To specify the provisioned resource:

1 To use the target that is currently associated with the request definition, select the *Provisioned resource* radio button.

The Provisioned resource radio button is selected by default if you're editing a request definition that refers to a valid resource. If you're defining a new provisioning request, this radio button is not selected.

2 To bind the request definition to another resource that was previously defined within the currently selected driver, select the *Available provisioned resources* radio button, and pick a target from the dropdown list.



NOTE: If the request definition was bound to a resource that is not an entitlement, you are not permitted to change the resource.

3 Select a category for the provisioned resource definition in the *Category* dropdown list.

The category defaults to the category for the currently selected provisioned resource. Whenever you change the provisioned resource, the category for the request definition is changed as well to match the category for the resource. If you want to assign a different category to the request definition, select that category in the Category dropdown list.

4 To create a new resource based on an Identity Manager entitlement, click the + button.

To edit an existing resource, click the pen button.

To define the characteristics of the resource, follow these steps:

4a Specify the name for the resource in the *Name (CN)* field.

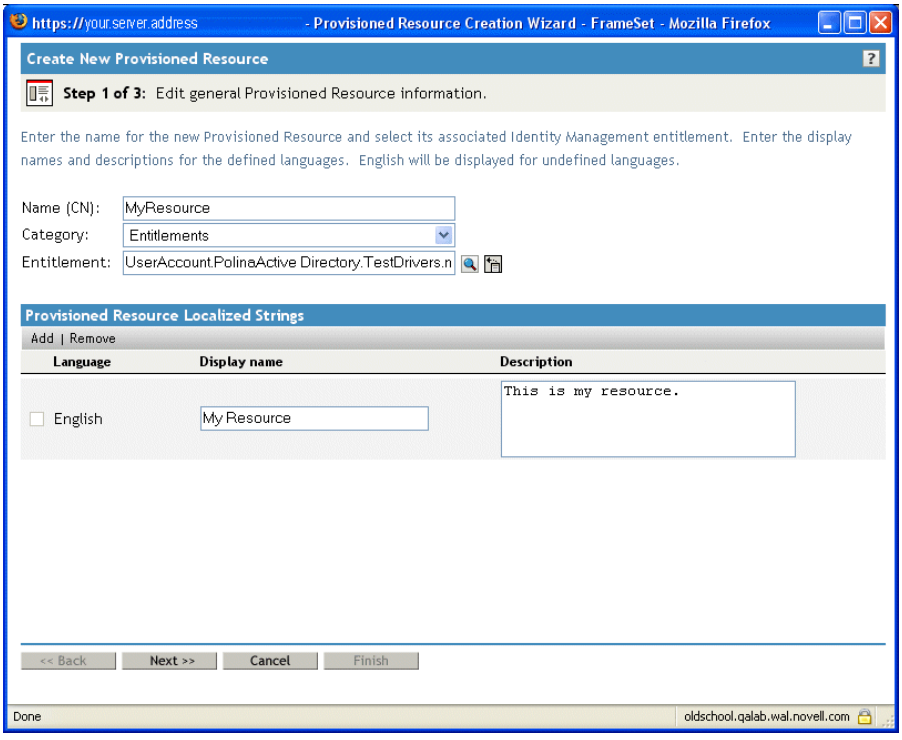
4b Select a category for the resource in the *Category* dropdown.

4c Specify the entitlement in the *Entitlement* field.

4d For each language you want to support in your application, click the check box beside the language in the list under *Provisioned Resource Localized Strings*, and type the localized text in the *Display Name* and *Description* fields. This text will be used to identify the provisioning resource throughout the user application.

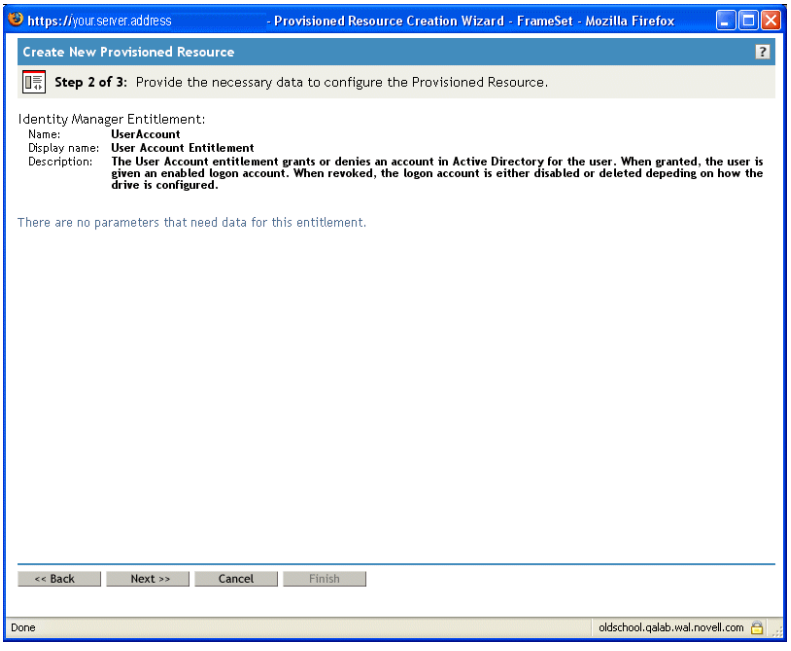
4e To add a new language to the list, click *Add* and select the desired language.

NOTE: By default, a newly created provisioning resource supports only English.



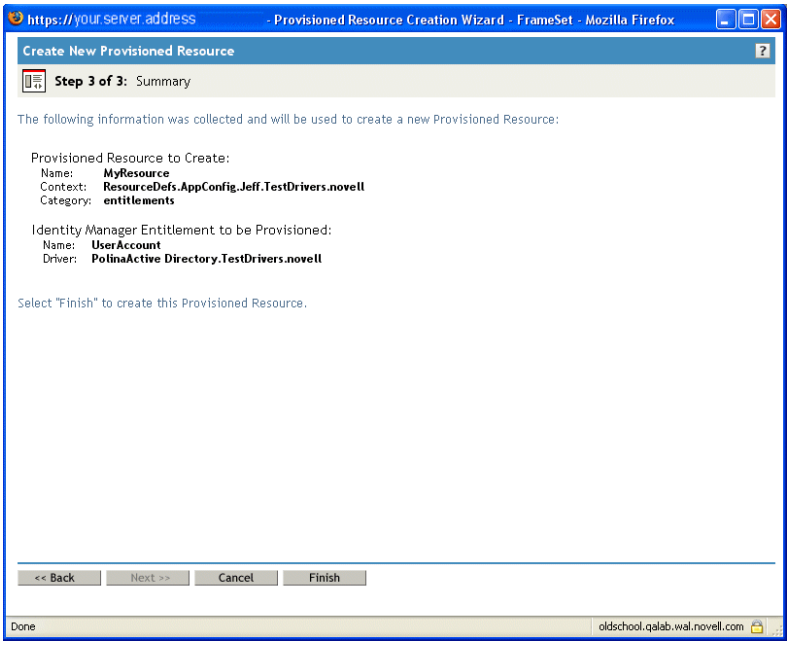
5 Click *Next*.

The Provisioned Resource wizard displays a screen to allow you to provide data for any parameters required for the entitlement.



6 If the entitlement does not require any entitlement parameters, click *Next*.

The Create New Provisioned Resource wizard displays the Summary page, which provides information about the resource you're defining.

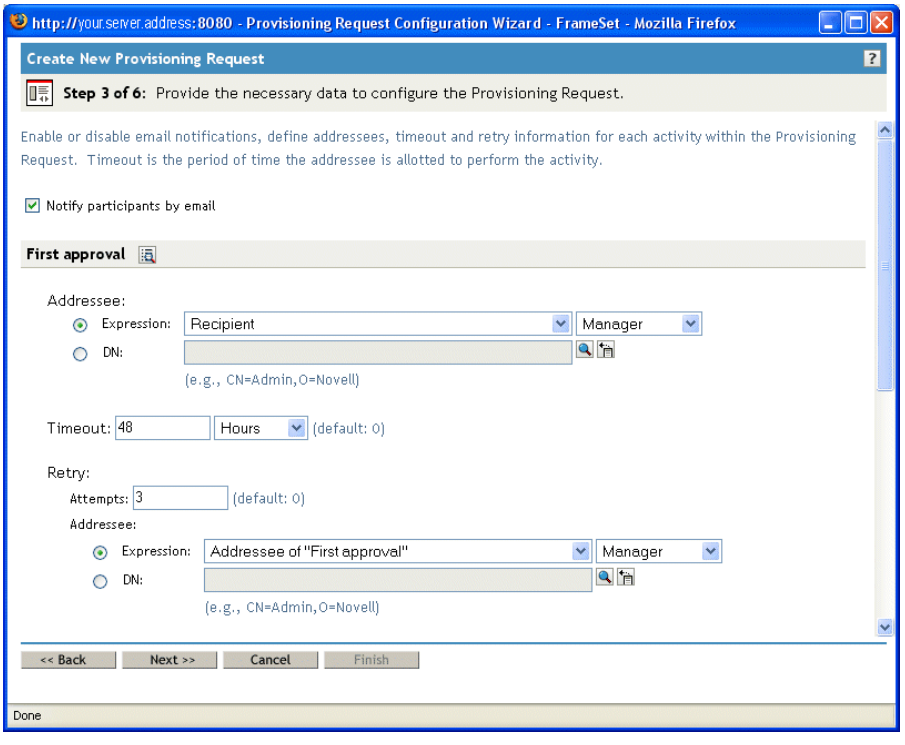


7 Click *Finish*.

Configuring the workflow activities

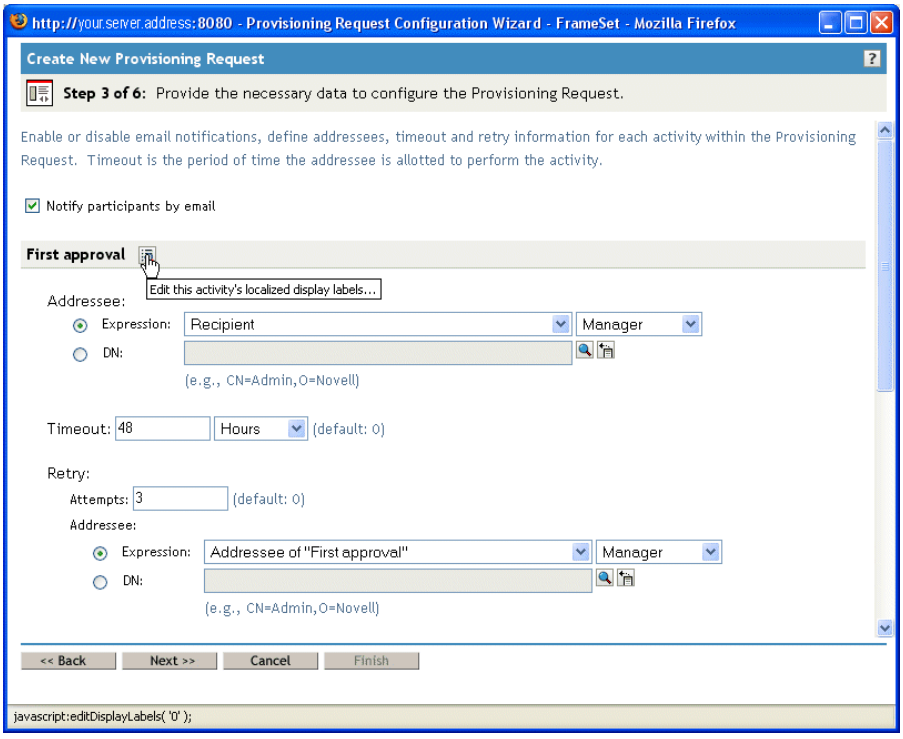
To configure the activities for the associated workflow:

1 Specify whether you want the addressee for each activity to be notified by e-mail by selecting or deselecting the *Notify participants by e-mail* check box.

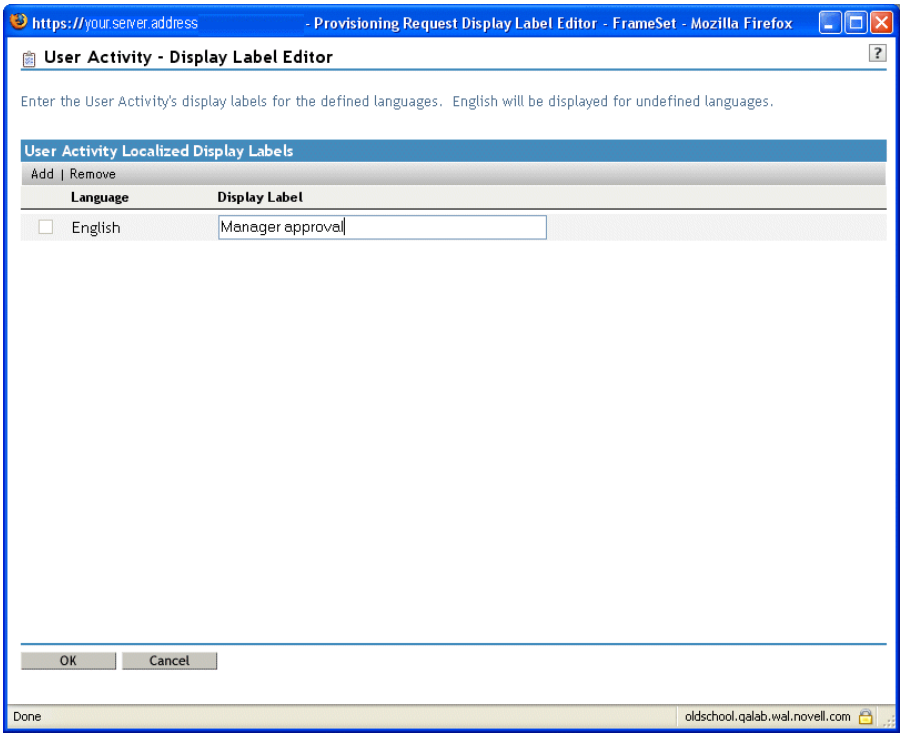


NOTE: If you select the *Notify participants by e-mail* check box, and the addressee has designated a proxy, the proxy will also be notified by e-mail. Delegates are not included in e-mail notifications.

2 For each workflow activity, optionally change the display label by clicking the icon beside the name of the activity (in this case, First Approval).



Type the display label in the *Display Label* field and click *OK*.



NOTE: The default display labels (First approval, Second approval, and so on) suggest that approvals are processed sequentially. For parallel flows, you may want to specify labels that do not imply sequential processing. For example, you might want to assign labels such as One of Three Parallel Approvals, Two of Three Parallel Approvals, and so on.

3 For each workflow activity, also provide the following information:

Field	Description
Addressee Expression	Specifies a dynamic expression that identifies the addressee for the activity. The addressee is determined at runtime, based on how the expression is evaluated. The **first term** of an addressee expression can be any of the following values: • `Initiator` • `Recipient` • `Addressee of` *activity-name* A separate `Addressee of` *activity-name* term is listed in the Expression dropdown for each activity in the workflow (except the activity you are currently configuring). The *activity-name* is the display label you specified for the activity, or the default name, if you did not specify a display label. The **second term** of an addressee expression can be either of the following values: • `Manager` • `<No attribute>` **NOTE:** The `Manager` attribute is available automatically because it has been previously defined on the User entity in the abstraction layer. Other attributes (in addition to Manager) may be available for selection if they meet the following requirements: • Must be defined on the User entity in the abstraction layer • Must be single-valued • Must have a DN data type
Addressee DN	Specifies the distinguished name for a user, group, or task group. **NOTE:** If you want Task Group Managers to be able to search for tasks by task group (in the My Team Tasks action in the user application), you need to specify the task group as the addressee.
Timeout	Specifies the period of time allotted for the addressee to complete the task. The timeout interval applies each time the activity is executed by the addressee. Specify a value in seconds, minutes, hours, or days.

Field	Description
Retry Attempts	Specifies the number of times to retry the activity in the event of a timeout. When an activity times out, the workflow process may try to complete the activity again, depending on the retry count specified for the activity. With each retry, the workflow process may escalate the activity to another user. In this case, the activity is reassigned to another addressee (the user's manager, for example) to give this user an opportunity to finish the work of the activity. In the event that the last retry times out, the activity may be marked as approved or denied, depending on how the workflow was configured.
Retry Addressee Expression	Specifies a dynamic expression that identifies the user who should get this task in the event that the timeout limit has been reached. The retry addressee is determined at runtime, based on how the expression is evaluated. The **first term** of an addressee expression can be any of the following values: • `approval.getAddressee()` • `Initiator` • `Recipient` • `Addressee of` *activity-name* The `approval.getAddressee()` option gets the current addressee. A separate `Addressee of` *activity-name* term is listed in the Expression dropdown for each activity in the workflow (including the activity you are currently configuring). The *activity-name* is the display label you specified for the activity, or the default name, if you did not specify a display label. The **second term** of an addressee expression can be either of the following values: • `Manager` • `<No attribute>` If you select the `approval.getAddressee()` option, and then select `Manager`, each retry will escalate to a new manager at a higher level within the organization. Therefore, you need to be sure to set the retry count to a number that is suitable for your organization. In any case, the retry count should not exceed the number of levels of management above the current addressee.
Retry Addressee DN	Specifies the distinguished name for a user or group that should get this task in the event that the retry limit has been reached.

4 When you finish configuring an activity, you may need to scroll down to see the other activities for the flow.

5 Click *Next*.

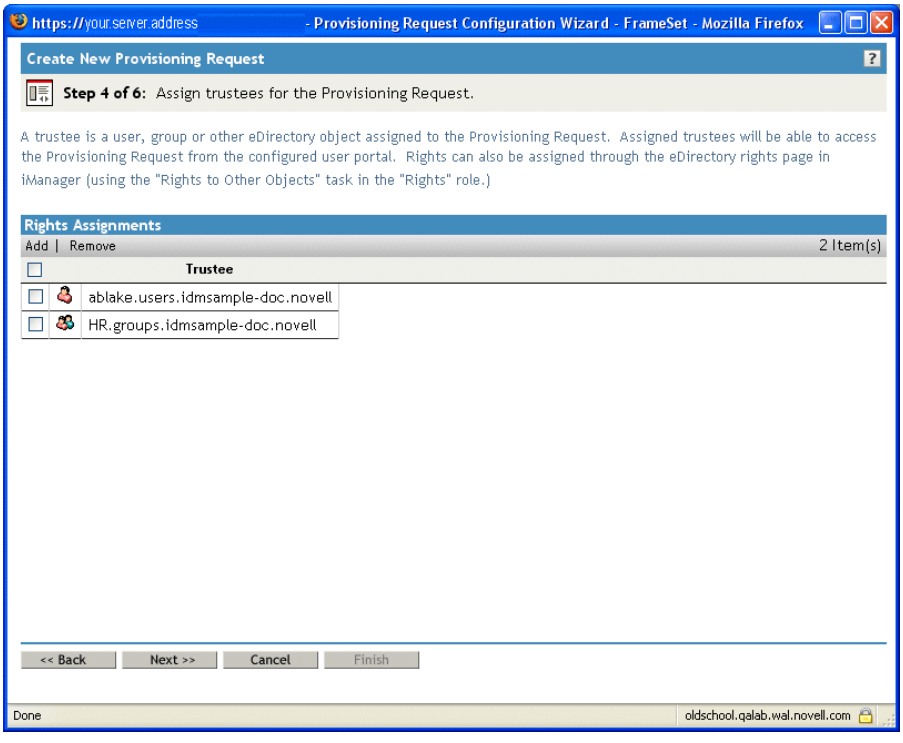
NOTE: The number of activities you can configure varies depending on which workflow template was bound to the request definition. The number and type of entitlement parameters varies depending on the provisioned resource associated with the request.

Specifying the access rights for the provisioning request

To specify the access rights for a provisioning request:

1 To add a user, group, or other eDirectory object to the list of trustees for this request definition, click *Add* and select the object.

Once you've added an object, it is included in the list of trustees.



2 To remove a user, group, or other object, select the item in the *Trustee* list and click *Remove*.

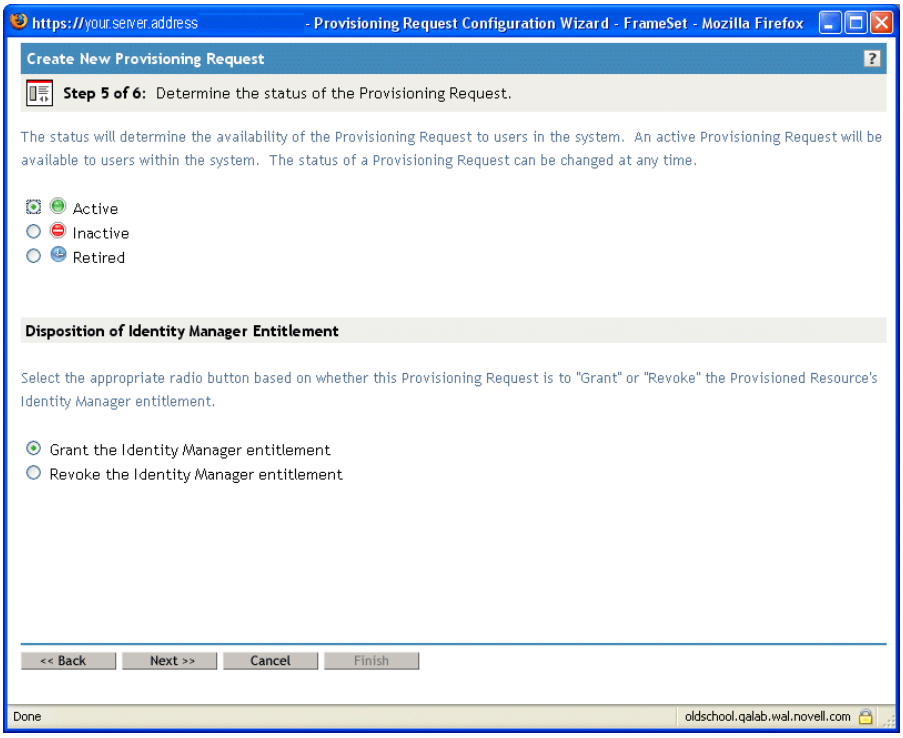
3 Click *Next*.

Specifying the initial status of the provisioning request

To set the initial status of the provisioning request:

1 Click the radio button for the desired status:

Status	Description
Active	Available for use.
Inactive	Temporarily unavailable for use. This is the default.
Retired	Permanently disabled.



2 Click the radio button for the correct action (Grant or Revoke).

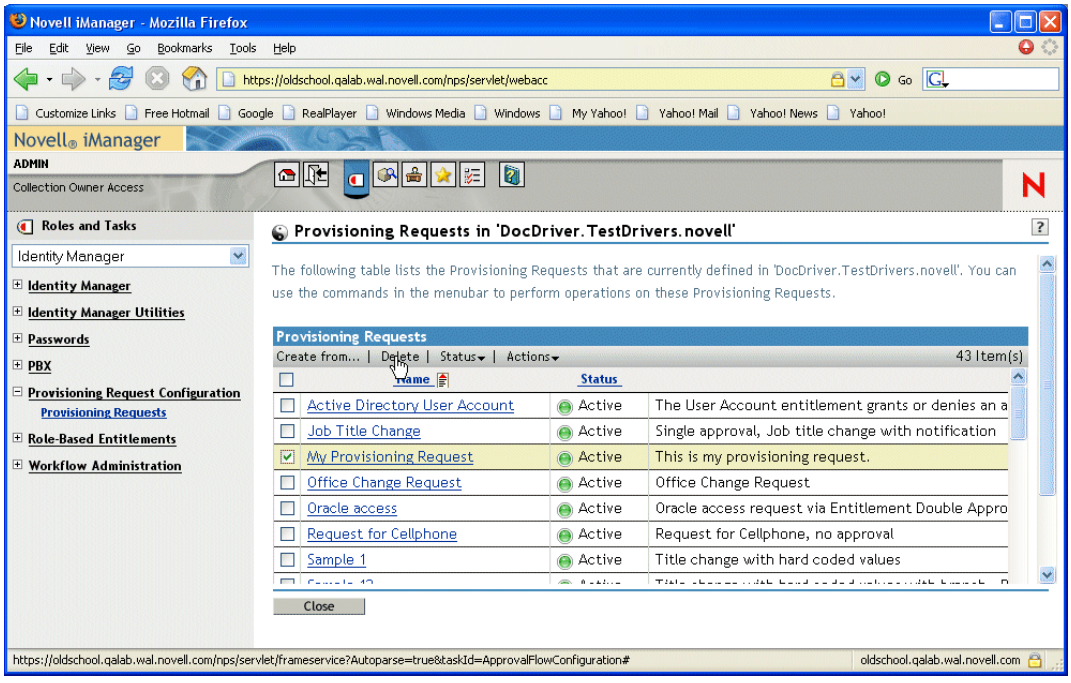
3 Click *Next*.

22.3.3 Deleting a provisioning request

To delete a provisioning request:

1 Select the provisioning request you want to delete by clicking the check box beside the name. You are not permitted to delete a provisioning request that is a template.

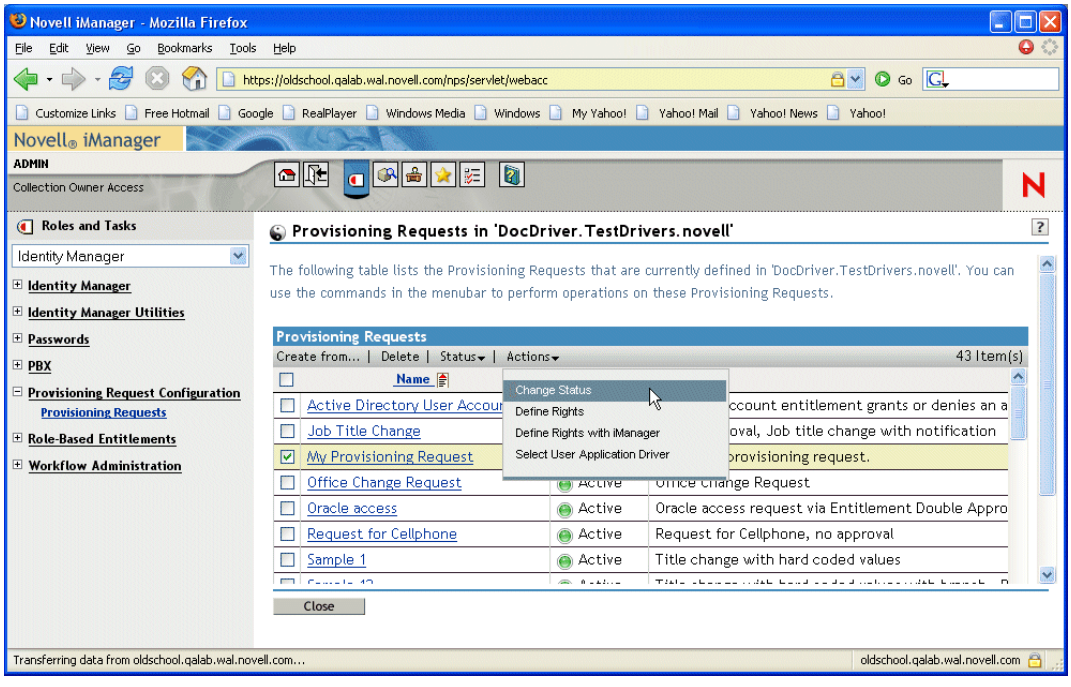
2 Click the *Delete* command in the Provisioning Request Configuration panel.



22.3.4 Changing the status of an existing provisioning request

To change the status of an existing provisioning request:

1. Select the provisioning request for which you want to change status by clicking the check box beside the name.
2. Click the *Change Status* command in the Provisioning Request Configuration panel.



3. Click the status in the Status menu:

Status	Description
Active	Available for use.
Inactive	Temporarily unavailable for use.
Retired	Permanently disabled.

4 Click the radio button for the correct action (Grant or Revoke).

5 Click *Finish*.

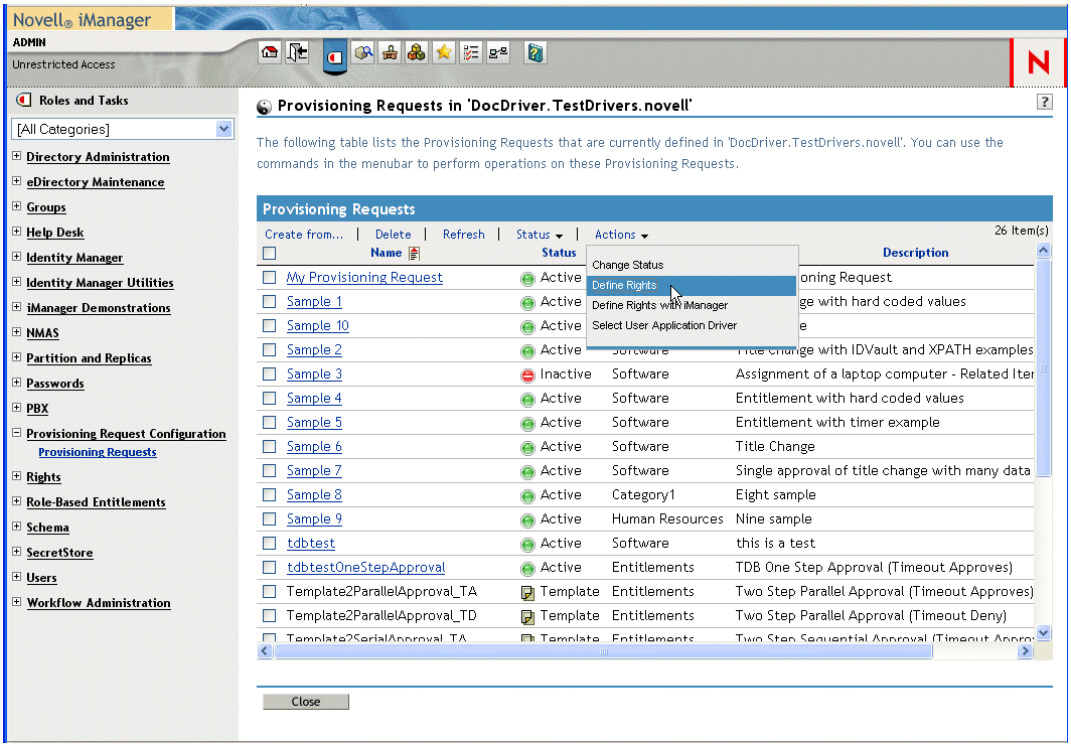
22.3.5 Defining rights on an existing provisioning request

To define rights on an existing provisioning request:

1 Select the provisioning request for which you define rights by clicking the check box beside the name.

2 Click the *Actions* command in the Provisioning Request Configuration panel.

3 Click the *Define Rights* command on the Actions menu.



4 Follow the steps presented under "Specifying the access rights for the provisioning request" on page 325.

To define rights on a provisioning request with iManager:

1 Select the provisioning request for which you want to define rights by clicking the check box beside the name.

2 Click the *Actions* command in the Provisioning Request Configuration panel.

3 Click the *Define Rights with iManager* command on the Actions menu.

Managing Provisioning Workflows

23

This chapter provides instructions for managing provisioning workflows at runtime. It also provides instructions for configuring e-mail notification for provisioning workflows.

Topics include:

23.1 About the Workflow Administration plug-in

The Workflow Administration plug-in to iManager provides a browser-based interface that lets you view the status of workflow processes, reassign activities within a workflow, or terminate a workflow in the event that it is stuck.

You can find the Workflow Administration plug-in in the *Identity Manager category* in iManager. The plug-in includes the *Workflows task* in the *Workflow Administration role*.

The Workflow Administration role also includes the *Email Templates* and *Email Server Options* tasks. These tasks are shortcuts to other tasks listed under the *Passwords* role.

About the Workflows task The Workflows task consists of these panels:

Panel	Description
Workflows	Provides the primary user interface for administering provisioning workflows. The interface lists workflows currently being processed, and lets you perform various actions on these workflows. When you first start the Workflows task, the Workflows panel requires that you select an Identity Manager user application driver. The driver points to a workflow server. You need to pick a driver before you can login to the server and begin workflow administration. Once you’ve selected a driver, you can specify search criteria for selecting the workflows to manage.
Workflow Detail	Provides a read-only user interface for viewing the details about a specific workflow.

23.2 Managing workflows

This section includes procedures for managing provisioning workflows using the Workflow Administration plug-in.

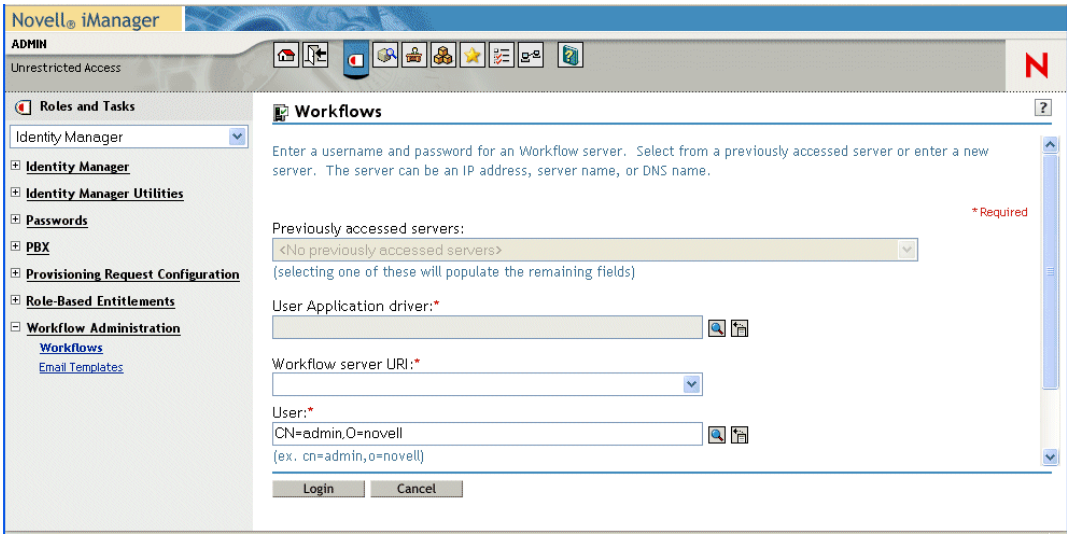
23.2.1 Connecting to a workflow server

Before you can begin managing workflows, you need to connect to a workflow server. If the user application driver is bound to a single workflow server, you can simply specify the name of the driver to use. If the driver is associated with multiple workflow servers, you need to select the target workflow server.

To connect to a workflow server:

1. Select the Identity Manager category in iManager.
2. Open the *Workflow Administration* role.
3. Click on the *Workflows* task.

   iManager displays the Workflows screen.

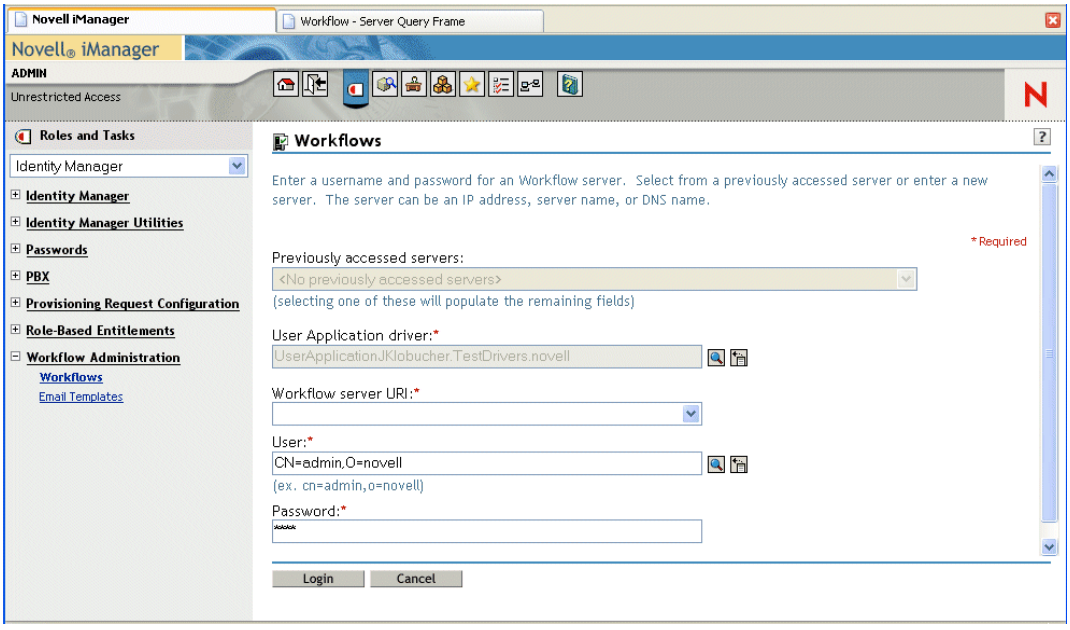


4. If you accessed the target workflow server previously, you can select the server from the *Previously accessed servers* dropdown list.

   iManager fills in the remaining fields on the screen for you.

5. If you have yet not accessed a workflow server, specify the driver name in the *User Application Driver* field and click *OK*.

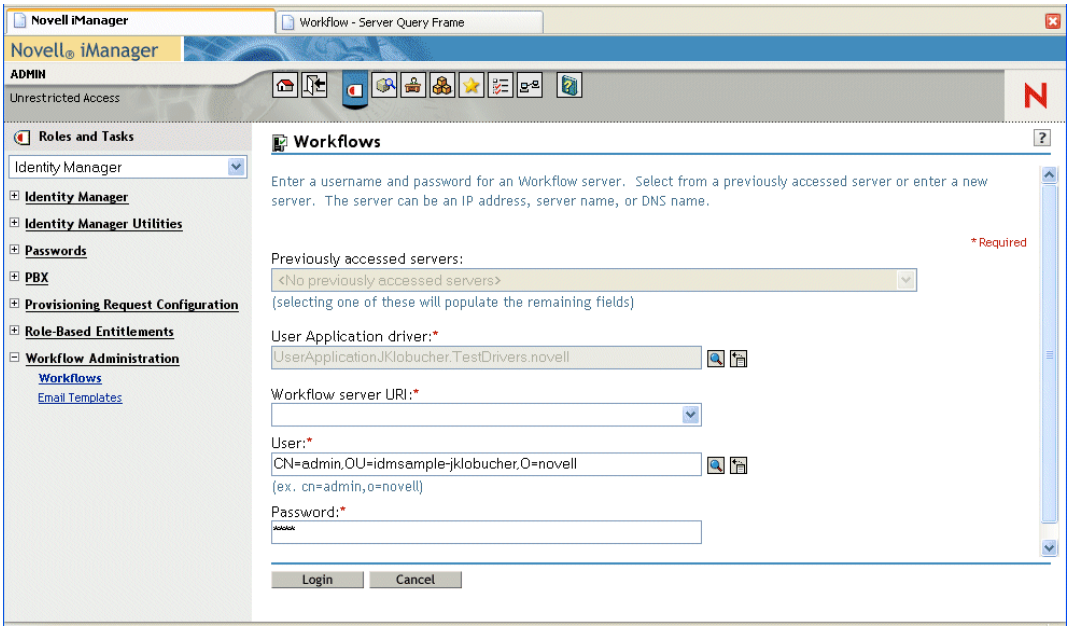
   iManager fills in the remaining fields on the screen for you.



6 If the driver is associated with multiple workflow servers, select the target server in the *Workflow server URI* field.

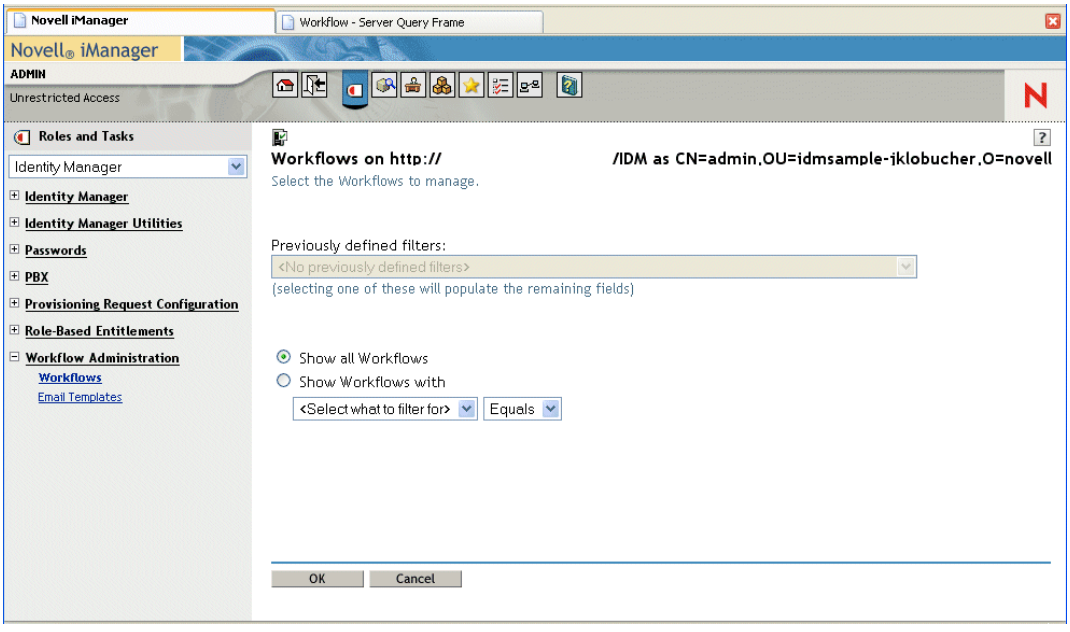
7 Optionally override the user name in the *User* field and the password in the *Password* field.

The user must be the User Application Administrator. By default, the user name is set to the user who is currently logged in to iManager. If this user is not the administrator, you need to change the user name. For example, you might want to modify the user to point to the User Application Administrator for the idmsample test OU, as shown below:



8 Click *Login*.

The Workflow Administration plug-in displays a page that allows you to specify a filter for finding workflows:

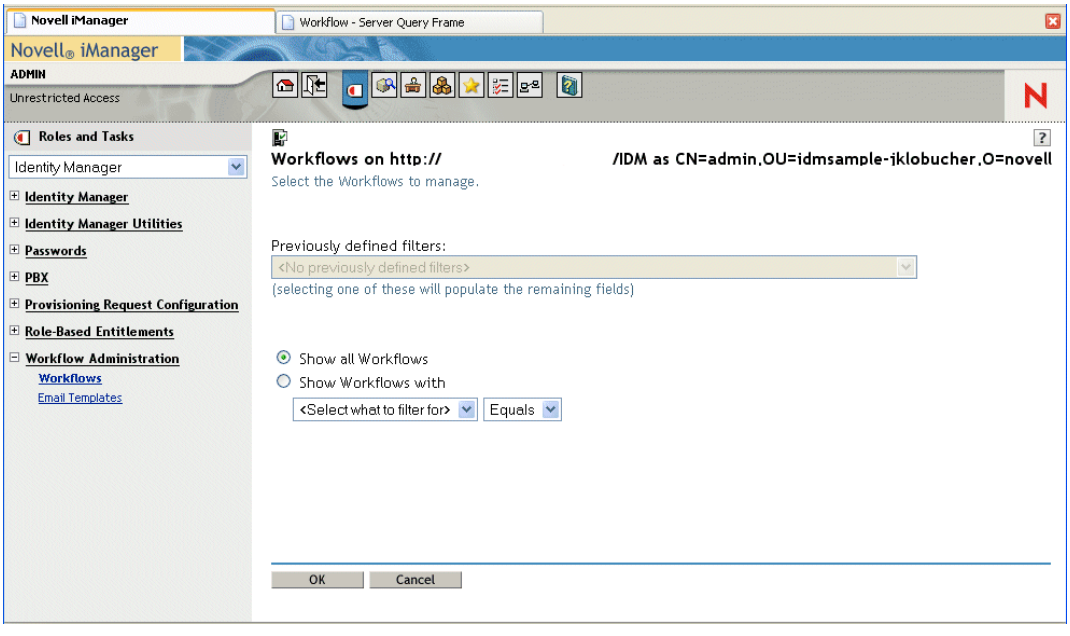


23.2.2 Finding workflows that match search criteria

If the target workflow server is running a large number of workflow processes, you may want to filter the list of workflows you see in iManager. To do this, you can specify search criteria.

To specify search criteria for filtering the list of workflows:

1 Select the *Show Workflows with* radio button.



NOTE: By default, the *Show all Workflows* radio button is selected. Do not change the default if you want to see the complete list of workflows on the server.

2 Select the attribute for which you want to specify criteria.

Attribute	Description
Creation time	Time that the workflow was initiated.
Initiator	User name of the requestor.
Recipient	User name of the recipient.
Process Status	Status of the workflow process as a whole (Completed, Running, or Terminated).
Approval status	Status of the approval process (Approved, Denied, or Retracted).
Entitlement status	Status of the entitlement initiated by the provisioning request (Error, Fatal, Success, Unknown, or Warning).

3 Select an operator:

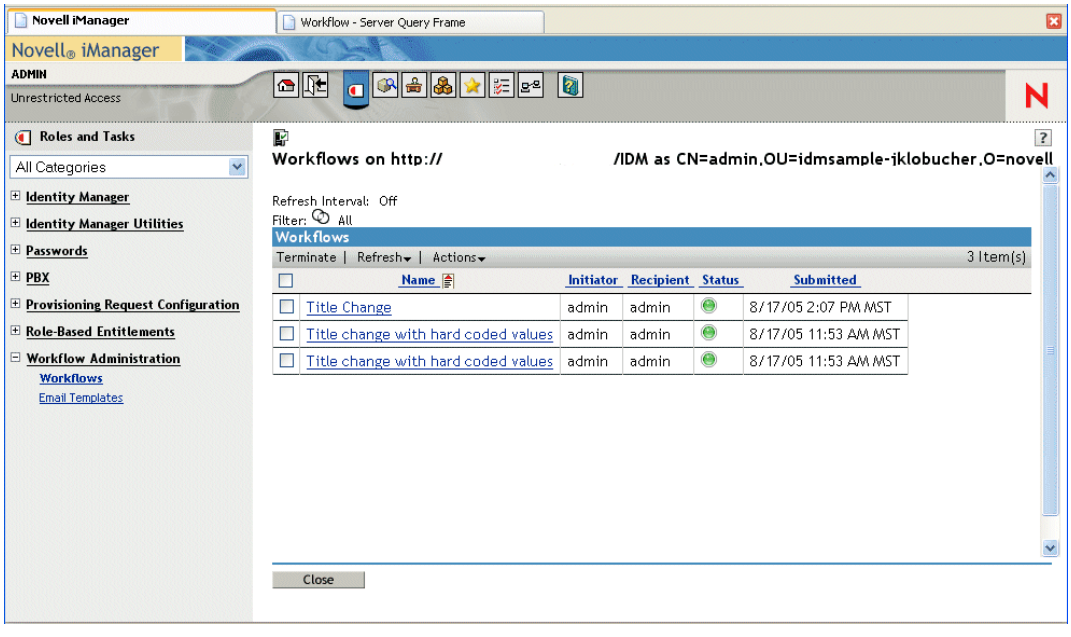
Operator	Comment
Equals	Supported for all attributes.
Before	Only supported for the Creation time attribute.
After	Only supported for the Creation time attribute.
Between	Only supported for the Creation time attribute.

4 Specify a value in the field below the attribute and operator.

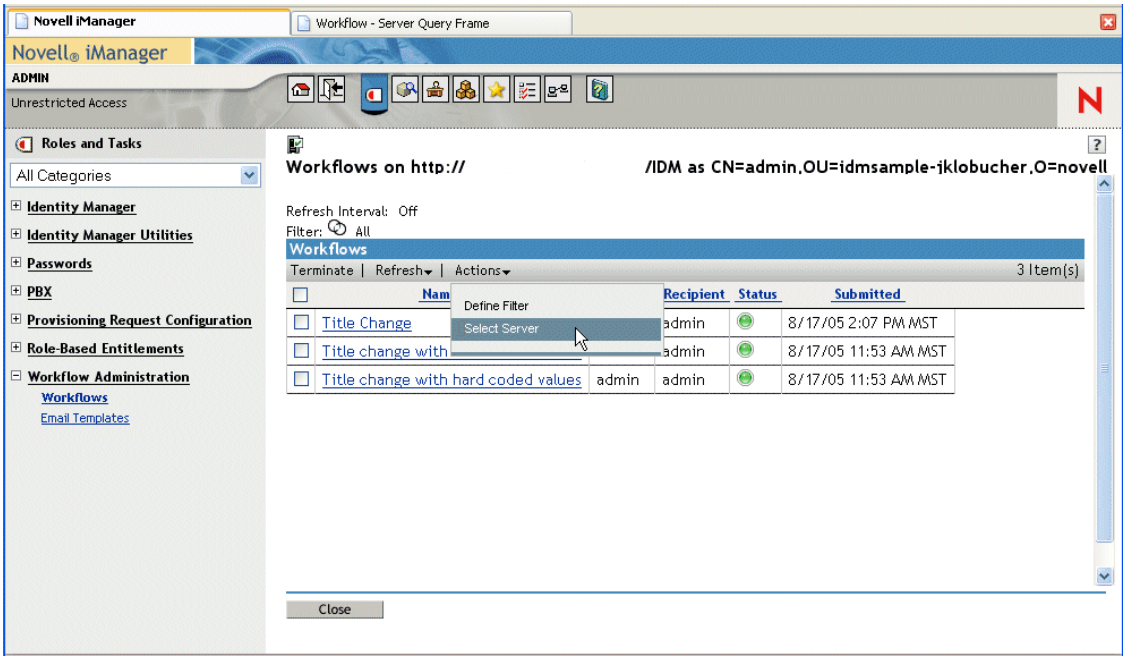
For Creation time, you can use the Date and time control to select the value. For Initiator and Recipient, you can use the Object History or the Object Selector to specify a value. For all other attributes, select the value from the dropdown list.

5 Click *OK*.

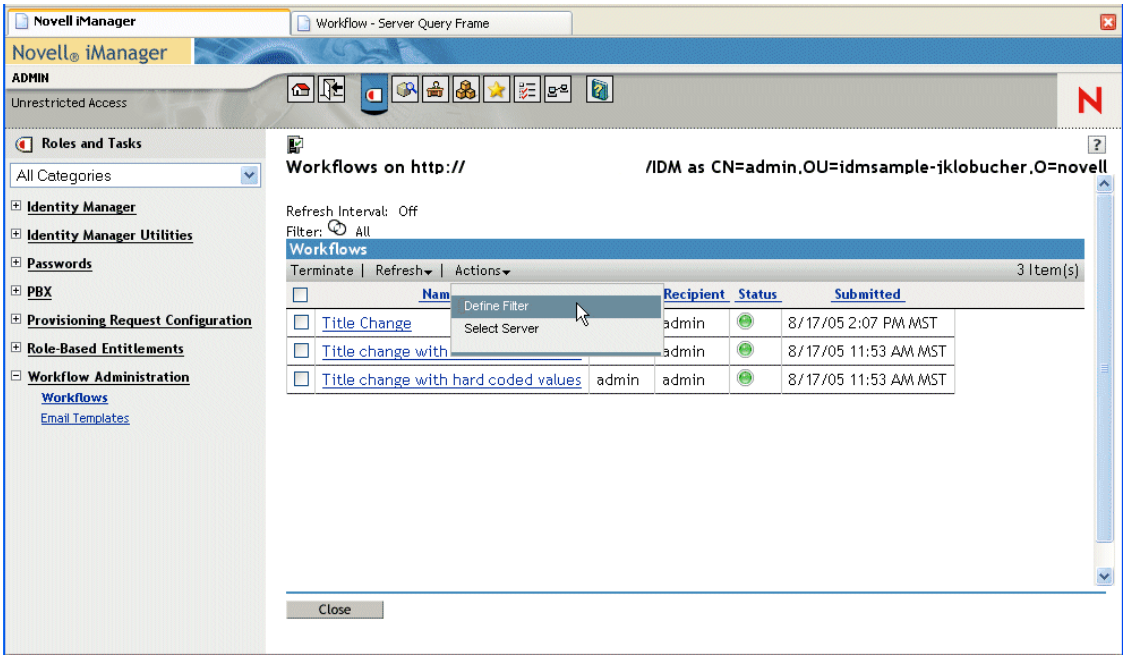
iManager displays the workflows you've selected on the Workflows panel.



Changing the target server and filter Once you've selected a workflow server, this selection remains in effect for the duration of your iManager session, unless you select a new server. To select a new server, click the *Actions* command and choose *Select Server* from the *Actions* menu.



To specify different search criteria, choose *Define Filter* on the *Actions* menu.



23.2.3 Controlling the active workflows display

The Workflows panel lists the workflows that match the search criteria you specified. In addition to filtering the list, you can control the display. For example, you can specify how often to refresh the list and sort the list on a particular column.

Refreshing the list of workflows

When the workflow server is very busy, the list of active workflows can change very frequently. In this case, you will want to refresh the list of active workflows running on the server.

To refresh the list of workflows:

1. Click the *Refresh* command in the Workflows panel.
2. Specify the refresh interval you want to use by selecting one of these options from the Refresh menu:
   - **2a** Refresh Off
   - **2b** Refresh Now
   - **2c** 10 seconds
   - **2d** 30 seconds
   - **2e** 60 seconds
   - **2f** 5 minutes

Sorting the list of workflows

If you have a large number of request definitions, you may want to sort the list by a particular column, such as the Name or Description.

To sort the list of workflows:

1. Click on the heading for the sort column.

23.2.4 Terminating a workflow instance

In the event that you do not want a workflow instance to continue its processing, you can terminate the workflow.

To terminate a workflow process instance:

1. Select the workflow in the Workflows panel by clicking the check box next to the workflow name.
2. Click the *Terminate* command in the Workflows panel.

23.2.5 Viewing details about a workflow instance

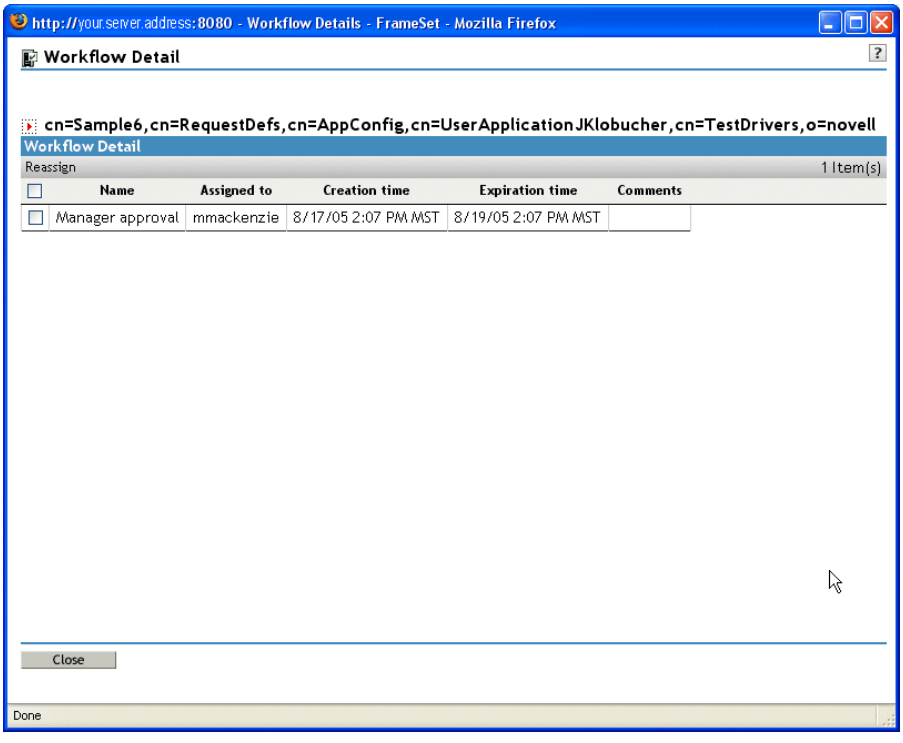
Once you've displayed a set of running workflows on a particular server, you can select a workflow instance to see more details about the running process.

NOTE: If a workflow instance uses a serial processing design pattern, the display will show a single activity as current, since only one user can act on the workitem at any point in time. However, if the workflow handles parallel processing and branching, there may be multiple current activities for a workflow instance.

To view details about a particular workflow instance:

1 Click the name of the workflow instance in the Workflows panel.

iManager displays the Workflow Detail panel.



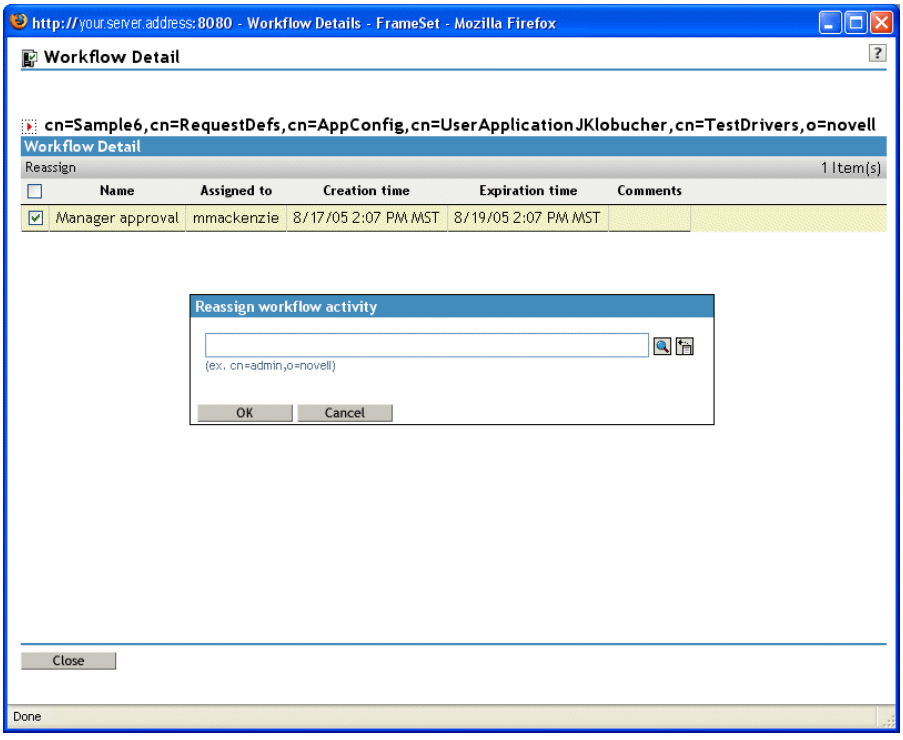
23.2.6 Reassigning a workflow instance

In the event that a workflow instance is stuck, you can reassign the workitem to another user or group.

To reassign a workflow instance:

1 Select the current activity associated with the workflow by clicking the check box next to the name in the Workflow Detail panel.

2 Click the *Reassign* command in the Workflow Detail panel.



3 Select the user or group to which you want to reassign the workitem.

23.3 Configuring the e-mail server

A workflow process often sends e-mail notifications at various points in the course of its execution. For example, an e-mail might be sent when a user assigns a workflow activity to a new addressee.

Before you can take advantage of the e-mail notification capabilities of Identity Manager, you need to configure the SMTP e-mail server. To do this, you need to use the *Email Server Options* task within the *Workflow Administration* role in iManager.

NOTE: This task is a shortcut to the *Email Server Options* task under the *Passwords* role.

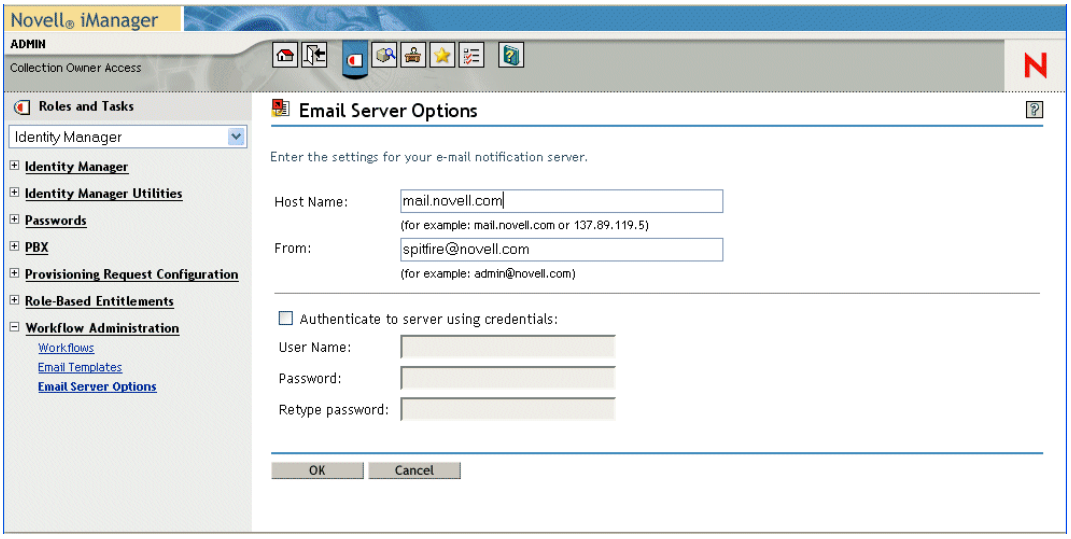
To configure the e-mail server:

1 Select the Identity Manager category in iManager.

2 Open the *Workflow Administration* role.

3 Click on the *Email Server Options* task.

iManager displays the Email Server Options screen.



4 Type the name (or IP address) of the host server in the *Host Name* field.

5 Type the e-mail address for the sender in the *From* field.

When the recipient opens the e-mail, this text is displayed in the From field of the e-mail header. Depending on your mail server settings, the text in this field might need to match a valid sender in the system, to allow the mail server to do reverse lookups or authentication. An example is helpdesk@company.com instead of descriptive text such as The Password Administrator.

6 If your server requires authentication before sending e-mail, select the *Authenticate to server using credentials* check box and specify the user name and password.

7 When you're done, click *OK*.

23.4 Working with the installed e-mail template

Identity Manager ships with an e-mail template that is designed specifically for workflow-based provisioning. This e-mail template is called *New Provisioning Request*. All provisioning request templates that ship with the product are associated with this e-mail template. Therefore, any new request definitions you create will use this e-mail template.

You can edit the New Provisioning Request template to change the content and format of e-mail messages, but you can not create new e-mail templates.

To edit the New Provisioning Request template, you need to use the *Email Templates* task within the *Workflow Administration* role in iManager.

NOTE: This task is a shortcut to the *Edit Email Templates* task under the *Passwords* role.

23.4.1 Default content and format

Here's what the New Provisioning Request template looks like after you install the product:

```
Dear $userFirstName$,
A new provisioning request has been submitted that requires your
```

```
approval.
Request name: $requestTitle$
Submitted by: $initiatorFullName$
Recipient: $recipientFullName$
Please review the details of this request at $PROTOCOL$://
$HOST$:$PORT$/$TASK_DETAILS$ to take the appropriate action.
You can review a list of all requests pending your approval at
$PROTOCOL$://$HOST$:$PORT$/$TASKLIST_CONTEXT$.
```

The template identifies the provisioning request definition that triggered the e-mail message. In addition, it includes a URL that redirects the addressee to the task that requires approval, as well as a URL that displays the complete list of tasks pending for that user.

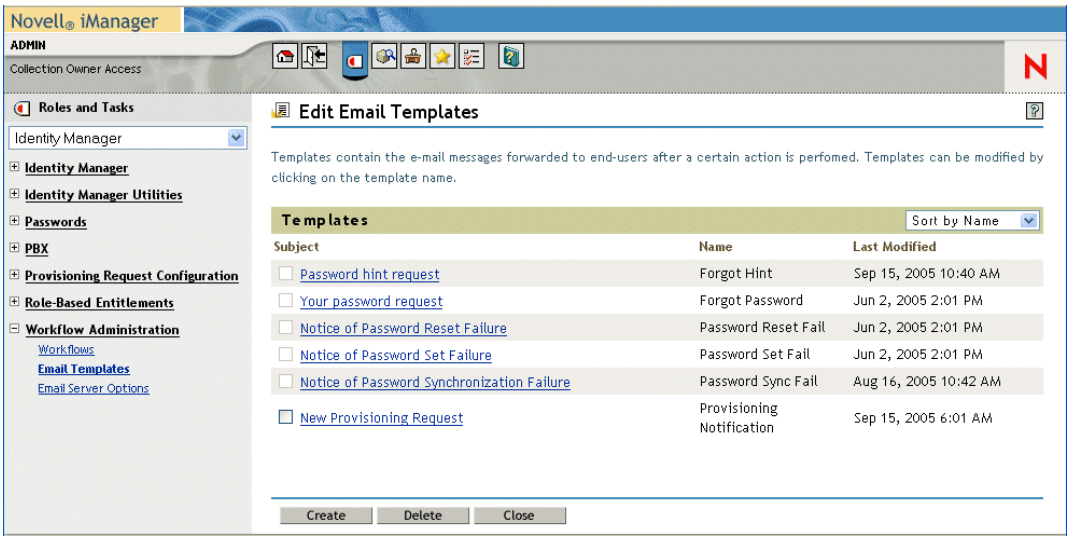
23.4.2 Editing the template

You can change the content or format of the New Provisioning Request template. Note that the template applies to all provisioning requests in the Identity Manager user application, so be sure your edits are suitable for all users and workflow tasks.

To edit the template:

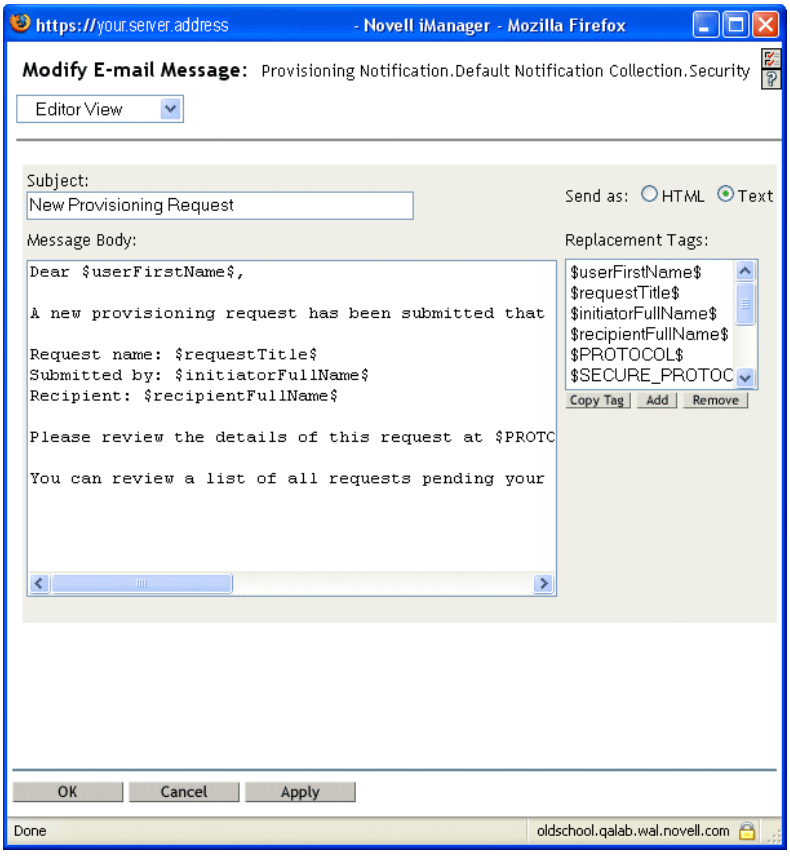
1. Select the Identity Manager category in iManager.
2. Open the *Workflow Administration* role.
3. Click on the *Email Templates* task.

   iManager displays the Edit Email Templates screen.



4. Click on *New Provisioning Request* in the list of templates.

   iManager displays the Modify E-mail Message screen.



5 Make your changes in the *Message Body* box.

6 If necessary, copy one or more of the supplied tags in the Replacement Tags list box to include dynamic text in the message body.

The replacement tags are described briefly below:

Tag	Description
$userFirstName$	The first name of the addressee.
$requestTitle$	The display name of the provisioning request definition.
$initiatorFullName$	The full name of the initiator.
$recipientFullName$	The full name of the recipient.
$PROTOCOL$	The protocol for URLs included in the e-mail message.
$SECURE_PROTOCOL$	The secure protocol for URLs included in the e-mail message.
$HOST$	The host for the JBoss application server that is running the Identity Manager user application.
$PORT$	The port for the Identity Manager user application.
$SECURE_PORT$	The secure port for the Identity Manager user application.
$TASKLIST_CONTEXT$	The page that displays the list of all requests pending for the addressee.

Tag	Description
$TASK_DETAILS$	The page that displays details for the request for which this e-mail message was generated.

7 When you're done, click *OK*.

23.4.3 Modifying default values for the template

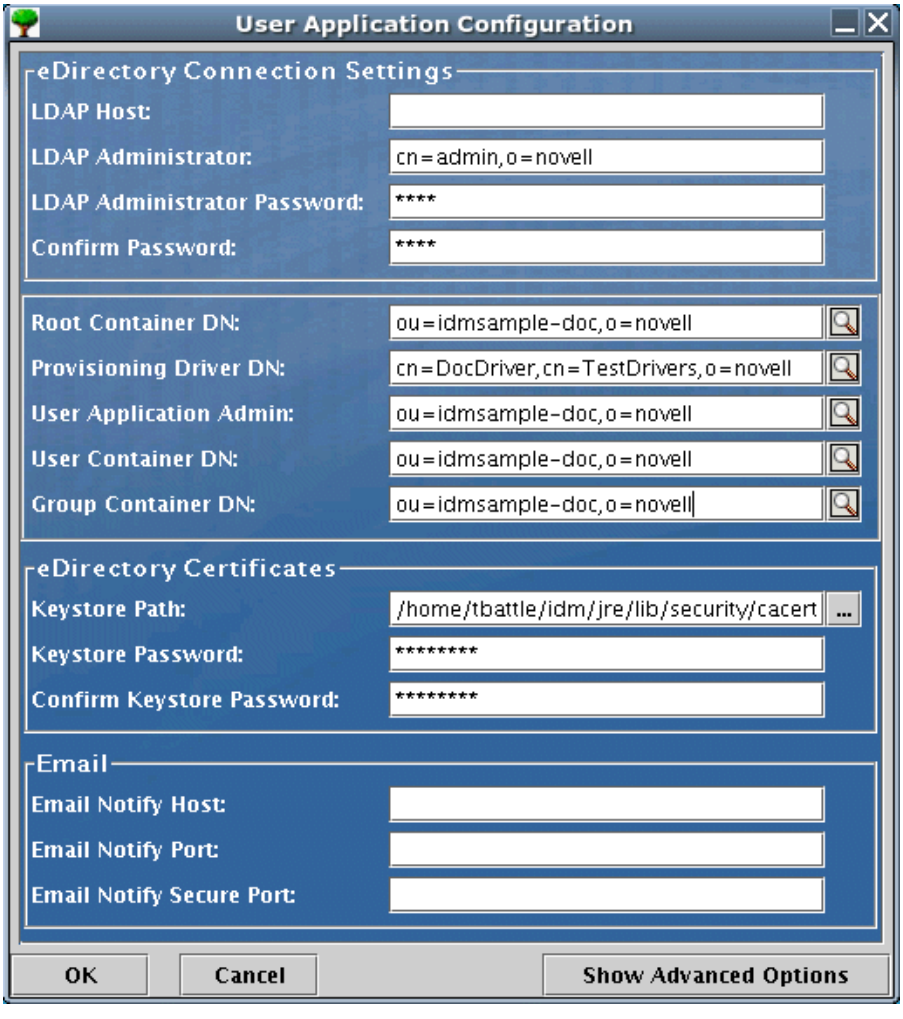
At installation time, you can set default values for several of the replacement tags used in e-mail templates. After you've completed the installation, you can also modify these values by using the User Application Configuration tool.

To modify the installation settings:

1 Run the ldapconfig.sh script in the idm folder.

```
./configupdate.sh
```

NOTE: On Windows, the file to run is `configupdate.bat`.



2 Make changes as necessary to any of the following fields:

Field	Description
Email Notify Host	Used to replace the $HOST$ token in e-mail templates used in approval flows. If left blank, computed by the server. (This is the JBoss host.)
Email Notify Port	Used to replace the $PORT$ token in e-mail templates used in approval flows.
Email Notify Secure Port	Used to replace the $SECURE_PORT$ token in e-mail templates used in approval flows.

3 Click *OK* to confirm your changes.

Appendixes

VI

The following appendixes provide additional reference information and advanced topics for the Identity Manager user application.

Schema Extensions

A

A.1 Attribute schema extensions

ATTRIBUTE NAME	DESCRIPTION
srvprvAOLIMAddress	**AOL IM address**
srvprvActiveDelegatees	The active delegates of a user
srvprvActiveDelegators	The active delegators of a user
srvprvAssetRef	Representation of the aggregate asset properties for a named asset associated to a user via the srvprvAssetRecipientAux class
srvprvAssignExpiration	Time at which a proxy or delegate assignment expires
srvprvAssignFromContainer	Container subjects of a proxy or delegate assignment
srvprvAssignFromGroup	Group subjects of a proxy or delegate assignment
srvprvAssignFromUser	User subjects of a proxy or delegate assignment
srvprvAssignToRelationship	A target relationship of a delegate assignment
srvprvAssignToUser	The User targets of a proxy or delegate assignment
srvprvCategoryKey	Associates a given Provisioning Request Definition to a set of provisioning categories. Values are keys to a srvprvChoice instance
srvprvDefaultTheme	The default theme
srvprvEntitlementRef	Reference to a DirXML-Entitlement
srvprvEntityType	Specifies Directory Abstraction Layer Entity definition type
srvprvFlowStrategy	Specifies the flow invocation strategy to be used for the Provisioning Request Definition
srvprvGrant	Flag which if true specifies that the Provisioning Request Definition supports a Grant operation
srvprvGroupwiseIMAddress	Groupwise IM address
srvprvHeaderFillerFile	Header Filler File Name
srvprvHeaderFillerImage	Header Filler Image
srvprvHeaderFillerLastMod	Header Filler Last Modified
srvprvHeaderLogo2File	Header Logo Secondary Image File Name
srvprvHeaderLogo2Image	Header Logo Secondary Image

ATTRIBUTE NAME	DESCRIPTION
srvprvAOLIMAddress	**AOL IM address**
srvprvHeaderLogo2LastMod	Header Logo Secondary Last Modified
srvprvHeaderLogoFile	Header Logo Primary Image File Name
srvprvHeaderLogoImage	Header Logo Primary Image
srvprvHeaderLogoLastMod	Header Logo Primary Last Modified
srvprvHeaderTextureFile	Header Texture File Name
srvprvHeaderTextureImage	Header Texture Image
srvprvHeaderTextureLastMod	Header Texture Last Modified
srvprvIsTaskManager	Indicates if user is a task group manager
srvprvLocalizedDescrs	Provides set of localized description strings for the provisioning web applications, Designers and iManager
srvprvLocalizedNames	Provides set of localized display name strings for the provisioning web applications, Designers and iManager
srvprvLoginFile	Login File Name
srvprvLoginImage	Login Image
srvprvLoginLastMod	Login Last Modified
srvprvLoginSmallFile	Login Small File Name
srvprvLoginSmallImage	Login Small Image
srvprvLoginSmallLastMod	Login Small Last Modified
srvprvModified	Flag to indicate changes to definitions object instances in the directory model container
srvprvNavBckgrColor	Navigation Background Color
srvprvNavBckgrColorLastMod	Navigation Background Color Last Modified
srvprvNavColor	Navigation Color
srvprvNavColorLastMod	Navigation Color Last Modified
srvprvPreferredLocale	List of saved query/search criteria
srvprvProcessXML	XML document representing a Provisioning process definition including Workflow and Provisioning Action
srvprvRequestDefName	The provisioning request definition name associated with a delegate definition.
srvprvRequestXML	XML document representing the initial request form and its data bindings
srvprvRevoke	Flag which if true specifies that the Provisioning Request Definition supports a Revoke operation
srvprvStatus	Specifies the status of the Provisioning Object Supported values will include

ATTRIBUTE NAME	DESCRIPTION
srvprvAOLIMAddress	**AOL IM address**
srvprvTaskGroups	Groups for which the user is a task manager
srvprvUUID	Unique identifier for portlet
srvprvTaskManager	Task manager of the task group
srvprvYahooIMAddress	Yahoo IM address

A.2 Objectclass schema extensions

OBJECTCLASS NAME	DESCRIPTION
srvprvAppConfig	**Container for application configuration objects of the Provisioning System to which its DirXML-Driver parent connects**
srvprvAppDefs	Container for configuration objects used to initialise the Provisioning run-time environment, such as themes for the Identity Portal
srvprvAssetRecipientAux	Records the provisioning of non-IT assets on a user
srvprvChoice	Enumeration of values which can be assigned to a particular attribute, used in a query, etc for use in the Identity Portlets and other Web Application components
srvprvChoiceDefs	Container for Directory Abstraction Layer Choice definitions, to be exposed by the Identity Portlets and Web Applications
srvprvDelegateeAssignment	Delegatee assignment definition
srvprvDelegateeDefs	Container for delegatee definitions
srvprvDirectoryModel	Container for Directory Abstraction Layer meta-level objects, selected contents of the directory to be exposed by the Identity Portlets and Web Applications
srvprvDirectoryModelConfig	Runtime Directory Abstraction Layer configurarion parameters
srvprvEntity	Defines a view of selected attributes for defined classes in the directory, used by the Identity Portlets and other Web Application components
srvprvEntityAux	Standard ObjectClass
srvprvEntityDefs	Container for Directory Abstraction Layer Entity defintions, to be exposed by the Identity Portlets and Web Applications
srvprvProxyAssignment	Proxy assignment definition
srvprvProxyDefs	Container for proxy definitions
srvprvRelationship	Defines relationships between objects in the directory, for use in the Identity Portlets and other Web Application components

OBJECTCLASS NAME	DESCRIPTION
srvprvAppConfig	**Container for application configuration objects of the Provisioning System to which its DirXML-Driver parent connects**
srvprvRelationshipDefs	Container for Directory Abstraction Layer Relationship definitions, to be exposed by the Identity Portlets and Web Applications
srvprvRequest	Exposes one provisionable item to be granted or revoked, including the workflow process which defines the run-time aspects of the Workflow and Provisioning Target
srvprvRequestDefs	Container for Provisioning Request Definitions, the set of provisionable items to the Web Application run-time
srvprvResource	Defines the set of directory assignments to execute for a provisioning fulfillment operation (either Grant or Revoke)
srvprvResourceDefs	Container for Provisioning Target definitions, including design-time descriptions plus any template or unused targets
srvprvService	Describes how to invoke a specific Web Service from an Workflow This includes specification of input and return values
srvprvServiceDefs	Container for Service Definition objects, which wrap Web Services called by Workflows
srvprvTaskGroupAux	Service provisioning task group
srvprvTheme	Theme Object
srvprvUserAux	Service provisioning user entity
srvprvWebAppConfig	Web Application Config Object
srvprvWorkflow	Defines the network of activities including traversal conditions to be executed in order to obtain approval for a provisioning action
srvprvWorkflowDefs	Container for Workflow objects, including design-time descriptions plus any template or unused flows
srvprvServiceDefs	Container for Service Definition objects, which wrap Web Services called by Workflows
srvprvStatus	Specifies the status of the Provisioning Object Supported values will include
srvprvTaskGroupAux	Service provisioning task group
srvprvTaskGroups	Groups for which the user is a task manager
srvprvTaskManager	Task manager of the task group
srvprvTheme	Theme Object
srvprvUserAux	Service provisioning user entity
srvprvWebAppConfig	Web Application Config Object
srvprvWorkflow	Defines the network of activites including traversal conditions to be executed in order to obtain approval for a provisioning action

OBJECTCLASS NAME	DESCRIPTION
srvprvAppConfig	**Container for application configuration objects of the Provisioning System to which its DirXML-Driver parent connects**
srvprvWorkflowDefs	Container for Workflow objects, including design-time descriptions plus any template or unused flows
srvprvYahooIMAddress	Yahoo IM address

A.3 LDIF representation

The full schema information including syntaxes, containment rules, and other information not shown in the above summary tables, is given below (in LDIF format). This information is subject to change.

```
version: 1
# Copyright (c) 2004-2005 Unpublished Work of Novell, Inc. All Rights
# Reserved.
#
# THIS WORK IS AN UNPUBLISHED WORK AND CONTAINS CONFIDENTIAL,
# PROPRIETARY AND TRADE SECRET INFORMATION OF NOVELL, INC. ACCESS TO
# THIS WORK IS RESTRICTED TO (I) NOVELL, INC. EMPLOYEES WHO HAVE A NEED
# TO KNOW HOW TO PERFORM TASKS WITHIN THE SCOPE OF THEIR ASSIGNMENTS
AND
# (II) ENTITIES OTHER THAN NOVELL, INC. WHO HAVE ENTERED INTO
#  APPROPRIATE LICENSE AGREEMENTS. NO PART OF THIS WORK MAY BE USED,
#  PRACTICED, PERFORMED, COPIED, DISTRIBUTED, REVISED, MODIFIED,
#  TRANSLATED, ABRIDGED, CONDENSED, EXPANDED, COLLECTED, COMPILED,
#  LINKED, RECAST, TRANSFORMED OR ADAPTED WITHOUT THE PRIOR WRITTEN
#  CONSENT OF NOVELL, INC. ANY USE OR EXPLOITATION OF THIS WORK WITHOUT
#  AUTHORIZATION COULD SUBJECT THE PERPETRATOR TO CRIMINAL AND CIVIL
#  LIABILITY.
#
# Base schema extensions for SpitFire
#
# Last Modified: 6/27/05 (ek)
#
# See rfc2252 for information on attribute syntax definitions
#   String = 1.3.6.1.4.1.1466.115.121.1.15
#   Boolean = 1.3.6.1.4.1.1466.115.121.1.7
#   Octet String = 1.3.6.1.4.1.1466.115.121.1.40
#   DN = 1.3.6.1.4.1.1466.115.121.1.12
#   Case Exact String = 1.3.6.1.4.1.1466.115.121.1.26
#   Case Ignore List = 2.16.840.1.113719.1.1.5.1.6
#   Case Ignore String = 1.3.6.1.4.1.1466.115.121.1.15
#   Stream = 1.3.6.1.4.1.1466.115.121.1.5
#   Time = 1.3.6.1.4.1.1466.115.121.1.24
#
# OID registered for EPM:
#   subarc "450" registered at: https://wiki.innerweb.novell.com/
wiki.phtml?title=OID_Registration
#   attribute prefix: 2.16.840.1.113719.1.450.4.{3 digit unique per
```

```
attribute}
#   object class prefix: 2.16.840.1.113719.1.450.6.{3 digit unique
number per class}
#-------------------------------------------------------------------
-----------------------------------------------------
#-- Framework Attributes
#-------------------------------------------------------------------
-----------------------------------------------------
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.127
  NAME 'srvprvUUID'
  DESC 'Standard Attribute'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{64512}
  SINGLE-VALUE
  X-NDS_PUBLIC_READ '1'
  X-NDS_NOT_SCHED_SYNC_IMMEDIATE '1'
 )
dn: cn=schema
changetype: modify
add: objectClasses
objectClasses: (
  2.16.840.1.113719.1.450.6.127
  NAME 'srvprvEntityAux'
  DESC 'Standard ObjectClass'
  AUXILIARY MAY srvprvUUID
  X-NDS_NOT_CONTAINER '1'
 )
#-------------------------------------------------------------------
-----------------------------------------------------
#-- User Attributes
#-------------------------------------------------------------------
-----------------------------------------------------
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.60
  NAME 'srvprvHideUser'
  DESC 'Indicates if a user is hidden during searches'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.61
  NAME 'srvprvHideAttributes'
  DESC 'List of attributes a user is hiding from other users'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 )
```

```
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.62
  NAME 'srvprvQueryList'
  DESC 'List of saved query/search criteria'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.63
  NAME 'srvprvCapabilities1'
  DESC 'Place holder for classifying skills, knowledge, references,
etc.  Classifications are defined in the application.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.64
  NAME 'srvprvCapabilities2'
  DESC 'Place holder for classifying skills, knowledge, references,
etc.  Classifications are defined in the application.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.65
  NAME 'srvprvCapabilities3'
  DESC 'Place holder for classifying skills, knowledge, references,
etc.  Classifications are defined in the application.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.66
  NAME 'srvprvCapabilities4'
  DESC 'Place holder for classifying skills, knowledge, references,
etc.  Classifications are defined in the application.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
```

```
  2.16.840.1.113719.1.450.4.67
  NAME 'srvprvCapabilities5'
  DESC 'Place holder for classifying skills, knowledge, references,
etc.  Classifications are defined in the application.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.68
  NAME 'srvprvIMAddress'
  DESC 'Key-value pair of Instant messenger Addresses i.e.
groupwise~jsmith'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 )
# This is temporary until we convert the application to use the multi-
value IM address (srvprvIMAddress) above
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.69
  NAME 'srvprvGroupwiseIMAddress'
  DESC 'Groupwise IM address'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
 )
# This is temporary until we convert the application to use the multi-
value IM address (srvprvIMAddress) above
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.70
  NAME 'srvprvYahooIMAddress'
  DESC 'Yahoo IM address'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
 )
# This is temporary until we convert the application to use the multi-
value IM address (srvprvIMAddress) above
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.71
  NAME 'srvprvAOLIMAddress'
  DESC 'AOL IM address'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
```

```
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.72
  NAME 'srvprvActiveDelegatees'
  DESC 'The active delegatees of a user'
  SYNTAX  2.16.840.1.113719.1.1.5.1.6
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.73
  NAME 'srvprvActiveDelegators'
  DESC 'The active delegators of a user'
  SYNTAX  2.16.840.1.113719.1.1.5.1.6
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.74
  NAME 'srvprvIsTaskManager'
  DESC 'Indicates if user is a task group manager'
  SYNTAX  1.3.6.1.4.1.1466.115.121.1.7
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.75
  NAME 'srvprvTaskGroups'
  DESC 'Groups for which the user is a task manager'
  SYNTAX  1.3.6.1.4.1.1466.115.121.1.12
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.77
  NAME 'srvprvPreferredLocale'
  DESC 'List of saved query/search criteria'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: objectclasses
objectClasses: (
  2.16.840.1.113719.1.450.6.128
  NAME 'srvprvUserAux'
  DESC 'Service provisioning user entity'
  AUXILIARY MAY ( srvprvHideUser $ srvprvHideAttributes $
srvprvQueryList $
```

```
                  srvprvCapabilities1 $ srvprvCapabilities2 $
srvprvCapabilities3 $ srvprvCapabilities4 $ srvprvCapabilities5 $
                  srvprvIMAddress $ srvprvGroupwiseIMAddress $
srvprvYahooIMAddress $  srvprvAOLIMAddress $ srvprvIsTaskManager $
                  srvprvTaskGroups $ srvprvActiveDelegatees $
srvprvActiveDelegators $ srvprvPreferredLocale)
  X-NDS_NOT_CONTAINER '1'
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.129
  NAME 'srvprvTaskManager'
  DESC 'Task manager of the task group'
  SYNTAX  1.3.6.1.4.1.1466.115.121.1.12
 )
dn: cn=schema
changetype: modify
add: objectclasses
objectClasses: (
  2.16.840.1.113719.1.450.6.130
  NAME 'srvprvTaskGroupAux'
  DESC 'Service provisioning task group'
  AUXILIARY MAY ( srvprvTaskManager )
  X-NDS_NOT_CONTAINER '1'
 )
#-------------------------------------------------------------------------
-----------------------------------------------------
#-- Provisioning Attributes
#-------------------------------------------------------------------------
-----------------------------------------------------
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.100
  NAME 'srvprvCategoryKey'
  DESC 'Associates a given Provisioning Request Definition to a set of
provisioning categories. Values are keys to a srvprvChoice instance.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.101
  NAME 'srvprvGrant'
  DESC 'Flag which if true specifies that the Provisioning Request
Definition supports a Grant operation.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
  SINGLE-VALUE
 )
dn: cn=schema
```

```
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.102
  NAME 'srvprvRevoke'
  DESC 'Flag which if true specifies that the Provisioning Request
Definition supports a Revoke operation.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.103
  NAME 'srvprvFlowStrategy'
  DESC 'Specifies the flow invocation strategy to be used for the
Provisioning Request Definition.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.104
  NAME 'srvprvLocalizedNames'
  DESC 'Provides set of localized display name strings for the
provisioning web applications, Designers and iManager.'
  SYNTAX  1.3.6.1.4.1.1466.115.121.1.26
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.105
  NAME 'srvprvLocalizedDescrs'
  DESC 'Provides set of localized description strings for the
provisioning web applications, Designers and iManager.'
  SYNTAX  1.3.6.1.4.1.1466.115.121.1.26
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.106
  NAME 'srvprvStatus'
  DESC 'Specifies the status of the Provisioning Object. Supported
values will include: Inactive, Active, Template, and Retired.'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
```

```
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.107
  NAME 'srvprvProcessXML'
  DESC 'XML document representing a Provisioning process definition
including Workflow and Provisioning Action.'
  SYNTAX  1.3.6.1.4.1.1466.115.121.1.5
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.108
  NAME 'srvprvEntityType'
  DESC 'Specifies Directory Abstraction Layer Entity definition type:
P-Public definitions or S-System definitions.'
  SYNTAX  1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.109
  NAME 'srvprvRequestXML'
  DESC 'XML document representing the initial request form and its data
bindings'
  SYNTAX  1.3.6.1.4.1.1466.115.121.1.5
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.110
  NAME 'srvprvModified'
  DESC 'Flag to indicate changes to definitions object instances in the
directory model container'
  SYNTAX  1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.111
  NAME 'srvprvEntitlementRef'
  DESC 'Reference to a DirXML-Entitlement'
  SYNTAX  1.3.6.1.4.1.1466.115.121.1.12
  SINGLE-VALUE
 )
#-------------------------------------------------------------------
-----------------------------------------------------
```

```
#-- Provisioning Configuration Containers
#-----------------------------------------------------------------------
-------------------------------------------------------
dn: cn=schema
changetype: modify
add: objectclasses
objectClasses: (
  2.16.840.1.113719.1.450.6.100
  NAME 'srvprvAppConfig'
  DESC 'Container for application configuration objects of the
Provisioning System to which its DirXML-Driver parent connects.'
  SUP top
  STRUCTURAL
  MUST ( cn $ version )
  MAY ( description )
  X-NDS_NAMING ( 'cn' )
  X-NDS_CONTAINMENT ( 'DirXML-Driver' )
 )
dn: cn=schema
changetype: modify
add: objectclasses
objectClasses: (
  2.16.840.1.113719.1.450.6.101
  NAME 'srvprvRequestDefs'
  DESC 'Container for Provisioning Request Definitions, the set of
provisionable items to the Web Application run-time.'
  SUP top
  STRUCTURAL
  MUST ( cn )
  MAY ( description )
  X-NDS_NAMING ( 'cn' )
  X-NDS_CONTAINMENT ( 'srvprvAppConfig' )
 )
dn: cn=schema
changetype: modify
add: objectclasses
objectClasses: (
  2.16.840.1.113719.1.450.6.102
  NAME 'srvprvWorkflowDefs'
  DESC 'Container for Workflow objects, including design-time
descriptions plus any template or unused flows.'
  SUP top
  STRUCTURAL
  MUST ( cn )
  MAY ( description )
  X-NDS_NAMING ( 'cn' )
  X-NDS_CONTAINMENT ( 'srvprvAppConfig' )
 )
dn: cn=schema
changetype: modify
add: objectclasses
objectClasses: (
  2.16.840.1.113719.1.450.6.103
  NAME 'srvprvResourceDefs'
```

```
  DESC 'Container for Provisioning Target definitions, including
design-time descriptions plus any template or unused targets.'
  SUP top
  STRUCTURAL
  MUST ( cn )
  MAY ( description )
  X-NDS_NAMING ( 'cn' )
  X-NDS_CONTAINMENT ( 'srvprvAppConfig' )
 )
dn: cn=schema
changetype: modify
add: objectclasses
objectClasses: (
  2.16.840.1.113719.1.450.6.104
  NAME 'srvprvServiceDefs'
  DESC 'Container for Service Definition objects, which wrap Web
Services called by Workflows.'
  SUP top
  STRUCTURAL
  MUST ( cn )
  MAY ( description )
  X-NDS_NAMING ( 'cn' )
  X-NDS_CONTAINMENT ( 'srvprvAppConfig' )
 )
dn: cn=schema
changetype: modify
add: objectclasses
objectClasses: (
  2.16.840.1.113719.1.450.6.105
  NAME 'srvprvDirectoryModel'
  DESC 'Container for Directory Abstraction Layer meta-level objects,
selected contents of the directory to be exposed by the Identity
Portlets and Web Applications.'
  SUP top
  STRUCTURAL
  MUST ( cn )
  MAY ( description $ srvprvModified )
  X-NDS_NAMING ( 'cn' )
  X-NDS_CONTAINMENT ( 'srvprvAppConfig' )
 )
dn: cn=schema
changetype: modify
add: objectclasses
objectClasses: (
  2.16.840.1.113719.1.450.6.106
  NAME 'srvprvAppDefs'
  DESC 'Container for configuration objects used to initialise the
Provisioning run-time environment, such as themes for the Identity
Portal.'
  SUP top
  STRUCTURAL
  MUST ( cn )
  MAY ( description )
  X-NDS_NAMING ( 'cn' )
```

```
  X-NDS_CONTAINMENT ( 'srvprvAppConfig' )
 )
dn: cn=schema
changetype: modify
add: objectclasses
objectClasses: (
  2.16.840.1.113719.1.450.6.111
  NAME 'srvprvEntityDefs'
  DESC 'Container for Directory Abstraction Layer Entity defintions, to
be exposed by the Identity Portlets and Web Applications.'
  SUP top
  STRUCTURAL
  MUST ( cn )
  MAY ( description )
  X-NDS_NAMING ( 'cn' )
  X-NDS_CONTAINMENT ( 'srvprvDirectoryModel' )
 )
dn: cn=schema
changetype: modify
add: objectclasses
objectClasses: (
  2.16.840.1.113719.1.450.6.112
  NAME 'srvprvRelationshipDefs'
  DESC 'Container for Directory Abstraction Layer Relationship
definitions, to be exposed by the Identity Portlets and Web
Applications.'
  SUP top
  STRUCTURAL
  MUST ( cn )
  MAY ( description )
  X-NDS_NAMING ( 'cn' )
  X-NDS_CONTAINMENT ( 'srvprvDirectoryModel' )
 )
dn: cn=schema
changetype: modify
add: objectclasses
objectClasses: (
  2.16.840.1.113719.1.450.6.113
  NAME 'srvprvChoiceDefs'
  DESC 'Container for Directory Abstraction Layer Choice definitions,
to be exposed by the Identity Portlets and Web Applications.'
  SUP top
  STRUCTURAL
  MUST ( cn )
  MAY ( description )
  X-NDS_NAMING ( 'cn' )
  X-NDS_CONTAINMENT ( 'srvprvDirectoryModel' )
 )
#### Provisioning Configuration Object Classes
dn: cn=schema
changetype: modify
add: objectclasses
objectClasses: (
  2.16.840.1.113719.1.450.6.107
```

```
  NAME 'srvprvRequest'
  DESC 'Exposes one provisionable item to be granted or revoked,
including the workflow process which defines the run-time aspects of
the Workflow and Provisioning Target.'
  SUP top
  STRUCTURAL
  MUST ( cn $ srvprvStatus $ srvprvFlowStrategy $ srvprvGrant $
srvprvRevoke $ srvprvCategoryKey $ srvprvLocalizedNames $
srvprvLocalizedDescrs )
  MAY ( description $ srvprvEntitlementRef $ XmlData $ srvprvRequestXML
$ srvprvProcessXML )
  X-NDS_NOT_CONTAINER '1'
  X-NDS_NAMING ( 'cn' )
  X-NDS_CONTAINMENT ( 'srvprvRequestDefs' )
 )
dn: cn=schema
changetype: modify
add: objectclasses
objectClasses: (
  2.16.840.1.113719.1.450.6.108
  NAME 'srvprvWorkflow'
  DESC 'Defines the network of activites including traversal conditions
to be executed in order to obtain approval for a provisioning action.'
  SUP top
  STRUCTURAL
  MUST ( cn $ srvprvLocalizedNames $ srvprvLocalizedDescrs  )
  MAY ( description $ XmlData )
  X-NDS_NOT_CONTAINER '1'
  X-NDS_NAMING ( 'cn' )
  X-NDS_CONTAINMENT ( 'srvprvWorkflowDefs' )
 )
dn: cn=schema
changetype: modify
add: objectclasses
objectClasses: (
  2.16.840.1.113719.1.450.6.109
  NAME 'srvprvResource'
  DESC 'Defines the set of directory assignments to execute for a
provisioning fulfillment operation (either Grant or Revoke).'
  SUP top
  STRUCTURAL
  MUST ( cn $ srvprvLocalizedNames $ srvprvLocalizedDescrs )
  MAY ( description $ srvprvEntitlementRef $ XmlData )
  X-NDS_NOT_CONTAINER '1'
  X-NDS_NAMING ( 'cn' )
  X-NDS_CONTAINMENT ( 'srvprvResourceDefs' )
 )
dn: cn=schema
changetype: modify
add: objectclasses
objectClasses: (
  2.16.840.1.113719.1.450.6.110
  NAME 'srvprvService'
  DESC 'Describes how to invoke a specific Web Service from an
```

```
Workflow. This includes specification of input and return values.'
  SUP top
  STRUCTURAL
  MUST ( cn  )
  MAY ( description $ XmlData )
  X-NDS_NOT_CONTAINER '1'
  X-NDS_NAMING ( 'cn' )
  X-NDS_CONTAINMENT ( 'srvprvServiceDefs' )
 )
dn: cn=schema
changetype: modify
add: objectclasses
objectClasses: (
  2.16.840.1.113719.1.450.6.114
  NAME 'srvprvEntity'
  DESC 'Defines a view of selected attributes for defined classes in
the directory, used by the Identity Portlets and other Web Application
components.'
  SUP top
  STRUCTURAL
  MUST ( cn $ srvprvEntityType )
  MAY ( description $ XmlData )
  X-NDS_NOT_CONTAINER '1'
  X-NDS_NAMING ( 'cn' )
  X-NDS_CONTAINMENT ( 'srvprvEntityDefs' )
 )
dn: cn=schema
changetype: modify
add: objectclasses
objectClasses: (
  2.16.840.1.113719.1.450.6.115
  NAME 'srvprvRelationship'
  DESC 'Defines relationships between objects in the directory, for use
in the Identity Portlets and other Web Application components.'
  SUP top
  STRUCTURAL
  MUST ( cn )
  MAY ( description $ XmlData )
  X-NDS_NOT_CONTAINER '1'
  X-NDS_NAMING ( 'cn' )
  X-NDS_CONTAINMENT ( 'srvprvRelationshipDefs' )
 )
dn: cn=schema
changetype: modify
add: objectclasses
objectClasses: (
  2.16.840.1.113719.1.450.6.116
  NAME 'srvprvChoice'
  DESC 'Enumeration of values which can be assigned to a particular
attribute, used in a query, etc. for use in the Identity Portlets and
other Web Application components.'
  SUP top
  STRUCTURAL
  MUST ( cn )
```

```
  MAY ( description $ XmlData )
  X-NDS_NOT_CONTAINER '1'
  X-NDS_NAMING ( 'cn' )
  X-NDS_CONTAINMENT ( 'srvprvChoiceDefs' )
 )
dn: cn=schema
changetype: modify
add: objectclasses
objectClasses: (
  2.16.840.113719.1.450.6.117
  NAME 'srvprvDirectoryModelConfig'
  DESC 'Runtime Directory Abstraction Layer configurarion parameters'
  SUP top
  STRUCTURAL
  MUST ( cn )
  MAY ( description $ XmlData )
  X-NDS_NOT_CONTAINER '1'
  X-NDS_NAMING ( 'cn' )
  X-NDS_CONTAINMENT ( 'srvprvDirectoryModel' )
  )
#### User Aux Classes and Attributes
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.80
  NAME 'srvprvAssetRef'
  DESC 'Representation of the aggregate asset properties for a named
asset associated to a user via the srvprvAssetRecipientAux class.'
  SYNTAX 2.16.840.1.113719.1.1.5.1.6
 )
dn: cn=schema
changetype: modify
add: objectclasses
objectClasses: (
  2.16.840.1.113719.1.450.6.80
  NAME 'srvprvAssetRecipientAux'
  DESC 'Records the provisioning of non-IT assets on a user'
  AUXILIARY
  MAY ( srvprvAssetRef )
 )
#------------------------------------------------------------------------
-------------------------------------------------------
#-- Web Application Config Class
#------------------------------------------------------------------------
-------------------------------------------------------
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (2.16.840.1.113719.1.450.4.20 NAME
'srvprvDefaultTheme' DESC 'The default theme'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
dn: cn=schema
changetype: modify
```

```
add: objectclasses
objectClasses: ( 2.16.840.1.113719.1.450.6.21 NAME
'srvprvWebAppConfig'
  DESC 'Web Application Config Object'
  SUP top STRUCTURAL MUST (cn) MAY (description $ srvprvDefaultTheme $
XmlData )
      X-NDS_NOT_CONTAINER '1'
      X-NDS_NAMING 'cn'
      X-NDS_CONTAINMENT ( 'srvprvAppDefs' )
  )
#-------------------------------------------------------------------
-----------------------------------------------------
#-- Theme Branding Structural Class
#-------------------------------------------------------------------
-----------------------------------------------------
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.21
  NAME 'srvprvHeaderLogoImage'
  DESC 'Header Logo Primary Image'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.22
  NAME 'srvprvHeaderLogoFile'
  DESC 'Header Logo Primary Image File Name'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.23
  NAME 'srvprvHeaderLogoLastMod'
  DESC 'Header Logo Primary Last Modified'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.24
  NAME 'srvprvHeaderLogo2Image'
  DESC 'Header Logo Secondary Image'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
  SINGLE-VALUE
```

```
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.25
  NAME 'srvprvHeaderLogo2File'
  DESC 'Header Logo Secondary Image File Name'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 |
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.26
  NAME 'srvprvHeaderLogo2LastMod'
  DESC 'Header Logo Secondary Last Modified'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.27
  NAME 'srvprvHeaderTextureImage'
  DESC 'Header Texture Image'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.28
  NAME 'srvprvHeaderTextureFile'
  DESC 'Header Texture File Name'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.29
  NAME 'srvprvHeaderTextureLastMod'
  DESC 'Header Texture Last Modified'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
```

```
attributeTypes: (
  2.16.840.1.113719.1.450.4.30
  NAME 'srvprvHeaderFillerImage'
  DESC 'Header Filler Image'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.31
  NAME 'srvprvHeaderFillerFile'
  DESC 'Header Filler File Name'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.32
  NAME 'srvprvHeaderFillerLastMod'
  DESC 'Header Filler Last Modified'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.33
  NAME 'srvprvLoginImage'
  DESC 'Login Image'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.34
  NAME 'srvprvLoginFile'
  DESC 'Login File Name'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.35
  NAME 'srvprvLoginLastMod'
  DESC 'Login Last Modified'
```

```
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.36
  NAME 'srvprvLoginSmallImage'
  DESC 'Login Small Image'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.37
  NAME 'srvprvLoginSmallFile'
  DESC 'Login Small File Name'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.38
  NAME 'srvprvLoginSmallLastMod'
  DESC 'Login Small Last Modified'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.39
  NAME 'srvprvNavColor'
  DESC 'Navigation Color'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.40
  NAME 'srvprvNavColorLastMod'
  DESC 'Navigation Color Last Modified'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
 )
dn: cn=schema
```

```
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.41
  NAME 'srvprvNavBckgrColor'
  DESC 'Navigation Background Color'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.42
  NAME 'srvprvNavBckgrColorLastMod'
  DESC 'Navigation Background Color Last Modified'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: objectclasses
objectClasses: (
  2.16.840.1.113719.1.450.6.20
  NAME 'srvprvTheme'
  DESC 'Theme Object'
  SUP top STRUCTURAL MUST (cn) MAY (description $
                                    srvprvHeaderLogoImage $
srvprvHeaderLogoFile $ srvprvHeaderLogoLastMod $
                                    srvprvHeaderLogo2Image $
srvprvHeaderLogo2File $ srvprvHeaderLogo2LastMod $
                                    srvprvHeaderTextureImage $
srvprvHeaderTextureFile $ srvprvHeaderTextureLastMod $
                                    srvprvHeaderFillerImage $
srvprvHeaderFillerFile $ srvprvHeaderFillerLastMod $
                                  srvprvLoginImage $ srvprvLoginFile $
srvprvLoginLastMod $
                                    srvprvLoginSmallImage $
srvprvLoginSmallFile $ srvprvLoginSmallLastMod $
                                 srvprvNavColor $ srvprvNavColorLastMod
$
                                    srvprvNavBckgrColor $
srvprvNavBckgrColorLastMod )
  X-NDS_NOT_CONTAINER '1'
  X-NDS_CONTAINMENT ( 'srvprvAppDefs' )
  X-NDS_NAMING 'cn'
 )
#-------------------------------------------------------------------
-----------------------------------------------------
#-- Attributes, objects, and containers for Proxy, Delegatee and User
availability,
#-------------------------------------------------------------------
-----------------------------------------------------
dn: cn=schema
```

```
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.120
  NAME 'srvprvAssignFromUser'
  DESC 'User subjects of a proxy or delegatee assignment'
  SYNTAX  1.3.6.1.4.1.1466.115.121.1.12
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.121
  NAME 'srvprvAssignFromGroup'
  DESC 'Group subjects of a proxy or delegatee assignment'
  SYNTAX  1.3.6.1.4.1.1466.115.121.1.12
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.122
  NAME 'srvprvAssignFromContainer'
  DESC 'Container subjects of a proxy or delegatee assignment'
  SYNTAX  1.3.6.1.4.1.1466.115.121.1.12
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.123
  NAME 'srvprvAssignToUser'
  DESC 'The User targets of a proxy or delegatee assignment'
  SYNTAX  1.3.6.1.4.1.1466.115.121.1.12
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.124
  NAME 'srvprvAssignToRelationship'
  DESC 'A target relationship of a delegatee assignment'
  SYNTAX  1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.125
  NAME 'srvprvAssignExpiration'
  DESC 'Time at which a proxy or delegatee assignment expires'
  SYNTAX  1.3.6.1.4.1.1466.115.121.1.24
  SINGLE-VALUE
```

```
 )
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: (
  2.16.840.1.113719.1.450.4.126
  NAME 'srvprvRequestDefName'
  DESC 'The provisioning request definition name associated with a
delegatee definition.'
  SYNTAX  1.3.6.1.4.1.1466.115.121.1.12
 )
dn: cn=schema
changetype: modify
add: objectclasses
objectClasses: (
  2.16.840.1.113719.1.450.6.120
  NAME 'srvprvProxyDefs'
  DESC 'Container for proxy definitions.'
  SUP top
  STRUCTURAL
  MUST ( cn )
  MAY ( description )
  X-NDS_NAMING ( 'cn' )
  X-NDS_CONTAINMENT ( 'srvprvAppConfig' )
 )
dn: cn=schema
changetype: modify
add: objectclasses
objectClasses: (
  2.16.840.1.113719.1.450.6.121
  NAME 'srvprvDelegateeDefs'
  DESC 'Container for delegatee definitions.'
  SUP top
  STRUCTURAL
  MUST ( cn )
  MAY ( description )
  X-NDS_NAMING ( 'cn' )
  X-NDS_CONTAINMENT ( 'srvprvAppConfig' )
 )
dn: cn=schema
changetype: modify
add: objectclasses
objectClasses: (
  2.16.840.1.113719.1.450.6.122
  NAME 'srvprvProxyAssignment'
  DESC 'Proxy assignment definition'
  SUP top
  STRUCTURAL
  MUST ( cn $ srvprvAssignToUser )
  MAY ( description $ srvprvAssignFromUser $ srvprvAssignFromGroup $
srvprvAssignFromContainer $ srvprvAssignExpiration )
  X-NDS_NAMING ( 'cn' )
  X-NDS_CONTAINMENT ( 'srvprvProxyDefs' )
 )
```

```
dn: cn=schema
changetype: modify
add: objectclasses
objectClasses: (
  2.16.840.1.113719.1.450.6.123
  NAME 'srvprvDelegateeAssignment'
  DESC 'Delegatee assignment definition'
  SUP top
  STRUCTURAL
  MUST cn
  MAY ( srvprvRequestDefName $ description $ srvprvAssignFromUser $
srvprvAssignFromGroup $ srvprvAssignFromContainer $ srvprvAssignToUser
$ srvprvAssignToRelationship $ srvprvAssignExpiration )
  X-NDS_NAMING ( 'cn' )
  X-NDS_CONTAINMENT ( 'srvprvDelegateeDefs' )
 )
############## DO NOT DELETE THIS LINE #################
#########################################################
```

Configuring the Application Archive

B

This appendix describes advanced settings that can be configured only by editing the WAR file for the user application. Topics include:

B.1 About the user application WAR

The Identity Manager user application is packaged as a J2EE-compliant Web Application Archive (WAR) file. The user application WAR file contains a collection of Java classes and XML files that control the runtime behavior of the application. In general, the WAR should not be modified. However, in rare situations, you may find it necessary to open the WAR file and make minor changes to control the behavior of the application.

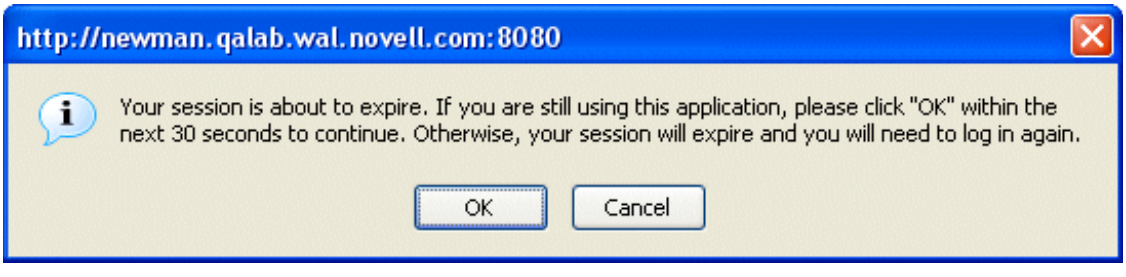
NOTE: The remainder of this appendix presumes familiarity with J2EE concepts and procedures. If you are unsure about how to make changes within a WAR file, consult your J2EE documentation.

B.2 Setting the session timeout

To prevent the server from becoming overloaded with inactive sessions, the Identity Manager user application times out a user session that remains inactive for an extended period of time. The default timeout interval is 10 minutes. You can change the default by editing the *web.xml* file in the WEB-INF folder of the user application WAR file.

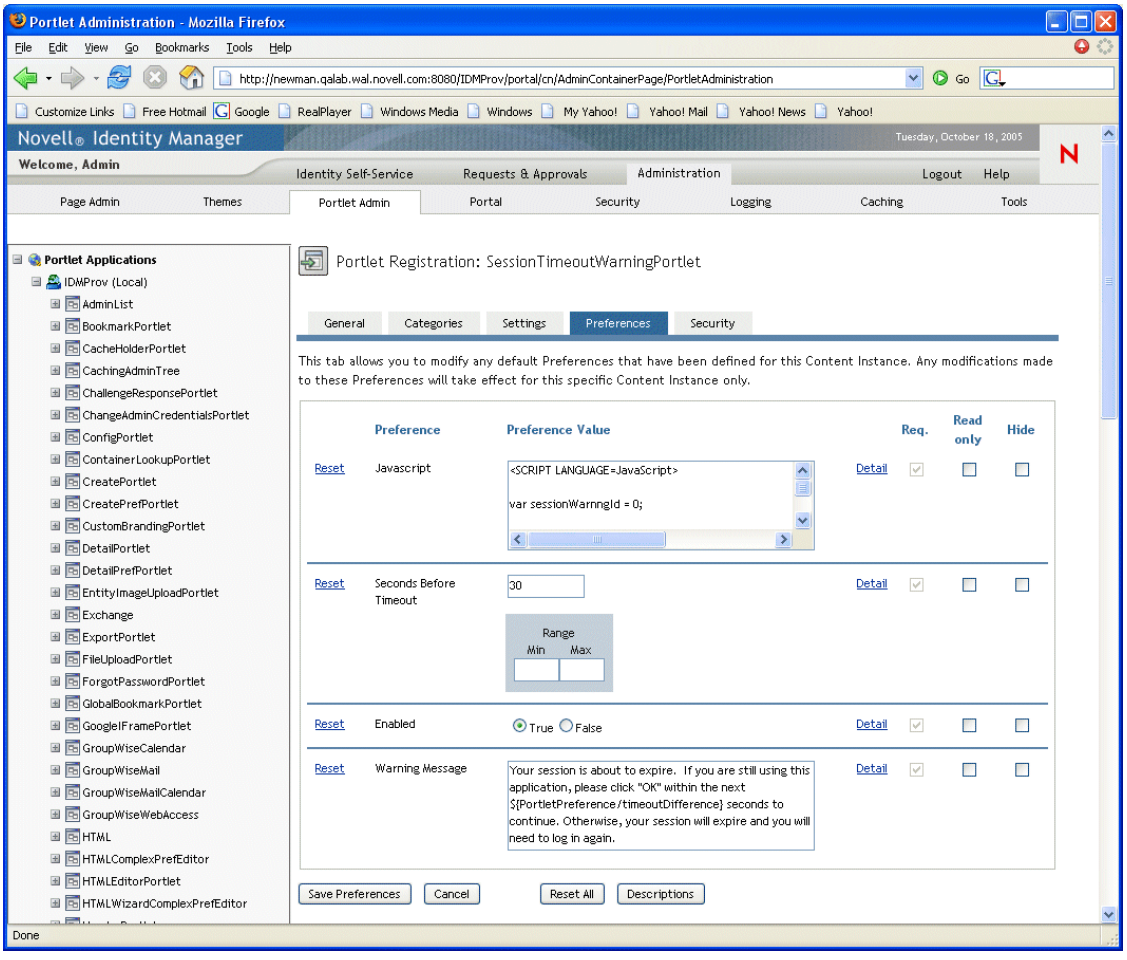
Editing the session timeout interval The web.xml file in the WAR has an element called <*session-timeout*> (which can be found under the <*session-config*> element) that specifies how long a session can be inactive before it times out. To set the session timeout interval, change the value of this element. The value must be specified in minutes.

Controlling the behavior of the alert message By default, the Identity Manager user application displays an alert message whenever a user’s session is about to time out.



If the user does not respond to the message by clicking OK, the session times out. The alert message is enabled by default. You can disable it if you like. In addition, you can specify how much time the user is allowed to have to respond to the alert message.

To control the behavior of the alert message, you need to configure the *SessionTimeoutWarningPortlet*. To do this, you need to edit the portlet preferences on the portlet registration, as shown below:



To specify how much time the user is allowed to have to respond to the alert message, edit the *Seconds Before Timeout* value. To disable the alert message altogether, click *False* next to *Enabled*. When you're done making your changes, click *Save Preferences*.