

Novell Identity Manager

3.5

May 14, 2007

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. See the [Novell International Trade Services Web \(http://www.novell.com/company/policies/trade_services/\)](http://www.novell.com/company/policies/trade_services/) page for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

Copyright © 1997-2007 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

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

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 (<http://www.novell.com/documentation>).

Novell Trademarks

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

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 SSLava™ 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.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR

CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR
DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

Contents

About This Guide	17
Part I Overview	19
1 Introduction to the User Application	21
1.1 About the User Application	21
1.1.1 About Identity Self-Service	22
1.1.2 About Workflow-Based Provisioning	22
1.2 User Application Architecture	23
1.2.1 User Interface	24
1.2.2 Directory Abstraction Layer	24
1.2.3 Workflow Engine	24
1.2.4 Application Server (J2EE-Compliant)	24
1.2.5 Database	25
1.2.6 User Application Driver	25
1.2.7 Designer for Identity Manager	26
1.2.8 iManager	26
1.2.9 Identity Manager Engine	26
1.2.10 Identity Vault	26
1.2.11 Novell Audit	27
1.3 User Application User Types	27
1.3.1 Administrators	27
1.3.2 Designers	29
1.3.3 Users	29
1.4 Design and Configuration Tools	30
1.5 What's Next	32
Part II Configuring the User Application Environment	35
2 Designing the Production Environment	37
2.1 Topology	37
2.1.1 Minimal Design	37
2.1.2 High Availability Design	38
2.1.3 Design Constraints	39
2.2 Security	40
2.2.1 Security Overview	40
2.2.2 Self-Signed Certificates	41
2.2.3 Turning on SSL in JBoss	42
2.2.4 Turning on SOAP Security	42
2.2.5 Mutual Authentication	43
2.2.6 Encryption of Sensitive User Application Data	43
2.3 Digital Signature Configuration	43
2.3.1 Setting Up the User Certificates	43
2.3.2 Configuring JBoss	47
2.3.3 Configuring the User Application	48
2.3.4 Configuring the Provisioning Request Definitions	48
2.4 Enabling Anonymous or Guest Access to the User Application	49
2.4.1 Establishing the Guest Account	50

2.5	Configuring Forgotten Password Self-Service	50
2.5.1	Accessing an External Password Management WAR	52
2.6	Performance Tuning	52
2.6.1	Logging	53
2.6.2	Identity Vault	54
2.6.3	JVM	55
2.6.4	Session Timeout Value	55
2.6.5	Tuning JBoss	56
2.6.6	Using Secure Sockets for User Application Connections to the Identity Vault.	56
2.7	Clustering	58
2.7.1	Clustering JBoss	58
2.7.2	Things to Do Before Installing the User Application	59
2.7.3	Installing the User Application to a JBoss Cluster	61
2.7.4	Things to Do After Installing the User Application	66
2.8	Localizing Text	69
3	Setting Up Logging	71
3.1	About Event Logging	71
3.1.1	About the Log Level Settings	71
3.1.2	Changing the User Application Log Level Settings	71
3.2	Logging to a Novell Audit or Sentinel Server	72
3.2.1	Adding the Identity Manager Application Schema to your Novell Audit Server as a Log Application	73
3.2.2	Enabling Audit Logging	74
3.2.3	Events That Are Logged	74
3.2.4	Log Reports	76
Part III	Administering the User Application	81
4	Using the Administration Tab	83
4.1	About the Administration Tab	83
4.2	Who Can Use the Administration Tab	83
4.3	Accessing the Administration Tab	84
4.4	Administration Actions You Can Perform	86
5	Application Configuration	89
5.1	Portal Configuration Tasks	89
5.1.1	Caching Management	89
5.1.2	Driver Status	99
5.1.3	LDAP Parameters	100
5.1.4	Logging Configuration	102
5.1.5	Portal Settings	107
5.1.6	Theme Administration	107
5.2	Working with the Import and Export Tools	113
5.2.1	Requirements	114
5.2.2	Restrictions	114
5.2.3	Exporting Portal Data	114
5.2.4	Importing Portal Data	116
5.3	Password Management Configuration	120
5.3.1	About Password Management Features	121
5.3.2	Configuring Challenge Response	125
5.3.3	Configuring Forgotten Password	126

5.3.4	Configuring Login	129
5.3.5	Configuring Password Sync Status	131
5.3.6	Configuring Password Hint Change	134
5.3.7	Configuring Change Password	135
6	Page Administration	139
6.1	About Page Administration	139
6.1.1	About Container Pages	139
6.1.2	About Shared Pages	146
6.1.3	An Exception to Page Usage	147
6.2	Creating and Maintaining Container Pages	147
6.2.1	Creating Container Pages	148
6.2.2	Adding Content to a Container Page	150
6.2.3	Deleting Content from a Container Page	152
6.2.4	Modifying the Layout of a Container Page	153
6.2.5	Arranging Content on the Container Page	153
6.2.6	Displaying a Container Page	155
6.3	Creating and Maintaining Shared Pages	155
6.3.1	Creating Shared Pages	156
6.3.2	Adding Content to a Shared Page	158
6.3.3	Deleting Content from a Shared Page	160
6.3.4	Modifying the Layout of a Shared Page	161
6.3.5	Arranging Content on the Shared Page	161
6.3.6	Displaying a Shared Page	163
6.4	Assigning Permissions for Pages	163
6.4.1	Assigning Page View Permission	164
6.4.2	Assigning Shared Page Owners	165
6.4.3	Enabling User Access to the Create User or Group Page	166
6.4.4	Enabling User Access to Individual Administration Pages	167
6.5	Setting Default Pages for Groups	168
6.6	Selecting a Default Shared Page for a Container Page	170
7	Portlet Administration	173
7.1	About Portlet Administration	173
7.2	Administering Portlet Definitions	173
7.2.1	Accessing Portlet Definitions in the Deployed Portlet Application	174
7.2.2	Registering Portlet Definitions	174
7.2.3	Viewing Information About Portlet Definitions	175
7.3	Administering Registered Portlets	177
7.3.1	Accessing Portlet Registrations in the Deployed Portlet Application	178
7.3.2	Viewing Information about Portlet Registrations	179
7.3.3	Assigning Categories to Portlet Registrations	180
7.3.4	Modifying Settings for Portlet Registrations	181
7.3.5	Modifying Preferences for Portlet Registrations	183
7.3.6	Assigning Security Permissions for Portlet Registrations	184
7.3.7	Unregistering a Portlet	186
8	Provisioning Configuration	189
8.1	About Provisioning Configuration	189
8.2	Configuring Delegation, Proxy, and Task Settings	189
8.2.1	Configuring the Delegation and Proxy Service	189
8.2.2	Scheduling Synchronization and Cleanup	191
8.2.3	Configuring Provisioning Interface Display Settings	192

8.3	Configuring the Digital Signature Service	193
8.4	Configuring the Workflow Engine and Cluster Settings	195
8.4.1	Configuring the Workflow Engine	195
8.4.2	Configuring the Workflow Cluster	198
9	Security Configuration	201
9.1	About Security Configuration	201
9.1.1	The User Application Administrator	201
9.1.2	The Provisioning Application Administrator	202
9.2	Assigning the User Application Administrator	202
9.3	Assigning the Provisioning Administrator	203
Part IV	Portlet Reference	205
10	About Portlets	207
10.1	Accessory Portlets	207
10.2	Admin Portlets	207
10.2.1	Shared Page Navigation Portlet	208
10.3	Identity portlets	208
10.4	System Components	210
11	Create Portlet Reference	211
11.1	About the Create portlet	211
11.2	Configuring the Create Portlet	213
11.2.1	Directory Abstraction Layer Setup	213
11.3	Setting Preferences	215
11.4	Configuring the Create Portlet for Self-Registration	216
11.4.1	Guest Access Required Settings	217
12	Detail Portlet Reference	219
12.1	About the Detail portlet	219
12.1.1	Displaying Entity Data	219
12.1.2	Editing Entity Data	223
12.1.3	E-Mailing Entity Data	225
12.1.4	Linking to an organization chart	226
12.1.5	Linking to Details of Other Entities	226
12.1.6	Printing Entity Data	227
12.1.7	Setting Preferred Locale	228
12.2	Prerequisites	228
12.2.1	Configuring the Directory Abstraction Layer	229
12.2.2	Assigning rights to entities	229
12.3	Launching Detail from Other Portlets	229
12.3.1	Launching Detail from the Search List Portlet	229
12.3.2	From the Org Chart Portlet	230
12.4	Using Detail on a Page	230
12.5	Setting Preferences	230
12.5.1	About the Preferences	230
12.6	Setting up Detail for Anonymous Access	233

13 Org Chart Portlet Reference	235
13.1 About Org Chart	235
13.1.1 About Org Chart Relationships	238
13.1.2 About Org Chart Display	239
13.2 Configuring the Org Chart Portlet	240
13.2.1 Directory Abstraction Layer Setup	241
13.2.2 Setting Preferences	241
13.2.3 Dynamically Loading Images	261
13.3 Configuring Org Chart for Guest Access	262
14 Resource Request Portlet	263
14.1 About the Resource Request Portlet	263
14.2 Configuring the Resource Request Portlet	263
14.2.1 Setting Preferences	264
15 Search List Portlet Reference	265
15.1 About Search List	265
15.1.1 About Results List Display Formats	267
15.2 Configuring the Search List portlet	269
15.2.1 Directory Abstraction Layer Setup	270
15.2.2 Setting Search List preferences	271
15.3 Configuring Search List for Anonymous Access	276
Part V Configuring and Managing Provisioning Workflows	279
16 Configuring the User Application Driver to Start Workflows	281
16.1 About the User Application Driver	281
16.2 Setting Up Workflows to Start Automatically	282
16.2.1 About Policies	282
16.2.2 Using the Policy Builder	282
16.2.3 Using the Schema Mapping Policy Editor	286
17 Configuring Provisioning Request Definitions	295
17.1 About the Provisioning Request Configuration Plug-in	295
17.2 Working with the Installed Templates	296
17.3 Configuring a Provisioning Request Definition	298
17.3.1 Selecting the Driver	299
17.3.2 Creating or Editing a Provisioning Request	300
17.3.3 Deleting a Provisioning Request	320
17.3.4 Changing the Status of an Existing Provisioning Request	321
17.3.5 Defining Rights on an Existing Provisioning Request	322
18 Managing Provisioning Workflows	325
18.1 About the Workflow Administration Plug-in	325
18.2 Managing Workflows	325
18.2.1 Connecting to a Workflow Server	326
18.2.2 Finding Workflows that Match Search Criteria	328
18.2.3 Controlling the Active Workflows Display	329
18.2.4 Terminating a Workflow Instance	330

18.2.5	Viewing Details about a Workflow Instance	330
18.2.6	Reassigning a Workflow Instance	330
18.2.7	Managing Workflow Processes in a Cluster.	331
18.3	Configuring the E-Mail Server	333
18.4	Working with E-Mail Templates	334
18.4.1	Default Content and Format.	335
18.4.2	Editing E-mail Templates	343
18.4.3	Modifying Default Values for the Template	345
18.4.4	Adding Localized E-Mail Templates.	346
19	Configuring Provisioning Teams	347
19.1	About the Provisioning Teams Plug-Ins	347
19.1.1	About Teams	347
19.1.2	About Team Request Rights	348
19.1.3	Using a Team to Manage Direct Reports.	349
19.2	Managing Provisioning Teams	349
19.2.1	Selecting the Driver	349
19.2.2	Creating or Editing a Provisioning Team	351
19.2.3	Deleting a Provisioning Team	358
19.3	Managing Provisioning Team Request Rights	358
19.3.1	Selecting the Driver	358
19.3.2	Creating or Editing a Provisioning Team Requests Object	359
19.3.3	Deleting a Provisioning Team Requests Object.	366
19.4	Creating a Team to Manage Direct Reports.	366
Part VI	Appendixes	373
A	Schema Extensions	375
A.1	Attribute schema extensions	375
A.2	Objectclass schema extensions	377
B	Metrics Web Service	381
B.1	About the Metrics Web Service	381
B.1.1	Web Service Semantics.	382
B.1.2	Web Service Endpoint	382
B.1.3	Web Service Methods Grouped by Security Permissions	382
B.1.4	Specifying Filters	385
B.1.5	Generating the Stub Classes	387
B.1.6	Obtaining the Remote Interface	387
B.1.7	Metrics Configuration Settings	389
B.2	Metrics Web Service API	390
B.2.1	Team Manager Methods	391
B.2.2	Provisioning Application Administrator Methods	393
B.2.3	Utility Methods	394
B.3	Metrics Web Service Examples	395
B.3.1	General Examples	395
B.3.2	Other Examples	396
C	Provisioning Web Service	399
C.1	About the Provisioning Web Service	399
C.1.1	Provisioning Web Service Overview	399

C.1.2	Provisioning Web Service Method Categories	400
C.2	Developing Clients for the Provisioning Web Service	400
C.2.1	Web Access to the Provisioning Web Service	400
C.2.2	A Java Client for the Provisioning Web Service	402
C.2.3	Developing a Mono Client	408
C.2.4	Sample Ant File	409
C.2.5	Sample Log4J File	411
C.3	Provisioning Web Service API	411
C.3.1	Processes	411
C.3.2	Provisioning	421
C.3.3	Work Entries	434
C.3.4	Comments	452
C.3.5	Configuration	459
C.3.6	Miscellaneous	463
C.3.7	Cluster	466

About This Guide

This guide describes how to administer the Novell Identity Manager user application. It includes these parts:

- ♦ Part I, “Overview,” on page 19
- ♦ Part II, “Configuring the User Application Environment,” on page 35
- ♦ Part III, “Administering the User Application,” on page 81
- ♦ Part IV, “Portlet Reference,” on page 205
- ♦ Part V, “Configuring and Managing Provisioning Workflows,” on page 279
- ♦ Part VI, “Appendixes,” on page 373

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 guide is for system administrators, architects, and consultants who are responsible for configuring, deploying, and managing the identity self-service features and 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*.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation, or go to www.novell.com/documentation/feedback.html and enter your comments there.

Documentation Updates

For the most recent version of the *Identity Manager User Application: Administration Guide*, visit the [Identity Manager Documentation Web site \(http://www.novell.com/documentation/idm35\)](http://www.novell.com/documentation/idm35).

Documentation Conventions

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

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

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

Overview

These sections introduces you to the Identity Manager User Application, and help you plan for its use in your organization.

- ♦ [Chapter 1, “Introduction to the User Application,” on page 21](#)

Introduction to the User Application

1

This section introduces the Identity Manager User Application. Topics include:

- ♦ [Section 1.1, “About the User Application,” on page 21](#)
- ♦ [Section 1.2, “User Application Architecture,” on page 23](#)
- ♦ [Section 1.3, “User Application User Types,” on page 27](#)
- ♦ [Section 1.4, “Design and Configuration Tools,” on page 30](#)
- ♦ [Section 1.5, “What’s Next,” on page 32](#)

1.1 About the User Application

The Identity Manager User Application is the business user’s view into the information, resources, and capabilities of Identity Manager. The User Application is a browser-based Web application that gives the user the ability to perform a variety of identity self-service tasks. In addition, when used in conjunction with the Provisioning Module and Novell Audit[®], the User Application provides a complete, end-to-end provisioning solution, giving users the ability to initiate and manage provisioning requests and approvals. The Identity Manager User Application is secure, scalable, and easy to manage.

The User Application enables you to address the following business needs:

- ♦ Providing user self-service, allowing a new user to self-register, and providing access to anonymous or guest users.

The User Application provides a set of portlets for managing identity information for employees. You can use the portlets as-is or customize them to deliver the following identity management services:

- ♦ Create a directory object or launch a workflow to create objects.
- ♦ Search identity data for white pages, yellow pages, green pages.
- ♦ View and modify user profiles and attributes.

For more information, see [Part IV, “Portlet Reference,” on page 205](#).

- ♦ Ensuring that access to corporate resources complies with organizational policies and that provisioning occurs within the context of the corporate security policy.

You can grant users access to identity data within the guidelines of corporate security policies.

For more information, see [Section 2.2, “Security,” on page 40](#).

- ♦ Reducing the administrative burden of entering, updating, and deleting user information across all systems in the enterprise.

You can create customized workflows to provide a Web-based interface for users to manipulate distributed identity data triggering workflows as necessary.

For more information, see [Part V, “Configuring and Managing Provisioning Workflows,” on page 279](#).

- ♦ Managing manual and automated provisioning of identities, services, resources, and assets, and supporting complex workflows.

You can implement manual provisioning by creating workflows that route provisioning requests to one or more authorities. For automated provisioning, you can configure the User Application to start workflows automatically in response to events occurring in the Identity Vault.

For more information, see [Part V, “Configuring and Managing Provisioning Workflows,” on page 279](#).

1.1.1 About Identity Self-Service

Identity is the foundation of the User Application. The application uses identity as the basis for authorizing users access to systems, applications, and databases. Each user’s unique identifier—and each user’s roles—comes with specific access rights to identity data. For example, users who are identified as managers can access salary information about their direct reports, but not about other employees in their organization.

The *Identity Self-Service* tab within the application gives users a convenient way to display and work with identity information. It enables your organization to be more responsive by giving users access to the information they need whenever they need it. For example, users might use the *Identity Self-Service* tab to:

- ♦ Manage their own user accounts directly
- ♦ Look up other users and groups in the organization on demand
- ♦ Visualize how those users and groups are related
- ♦ List applications with which they are associated

The User Application Administrator is responsible for setting up the contents of the *Identity Self-Service* tab. What business users can see and do is typically determined by how the application has been configured, by their job requirements and level of authority.

1.1.2 About Workflow-Based Provisioning

A key feature of the Identity Manager User Application is workflow-based provisioning, which enables you to automate the approval and revocation of user access to your organization’s secure resources. Resources can include digital entities such as user accounts, computers, and databases.

The User Application’s *Requests & Approvals* tab gives users a convenient way to make requests for resources. A *provisioning request* is a user or system action intended to grant or revoke resources. Provisioning requests can be initiated directly by the user (through the *Requests & Approvals* tab), or indirectly in response to events occurring in the Identity Vault.

When a provisioning request requires permission from one or more individuals in an organization, the request starts one or more workflows. The workflows coordinate 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. A successful provisioning request results in a *provisioned resource*. Provisioned resources are mapped to Identity Manager entitlements.

By default, the *Requests & Approvals* tab in the User Application does not display any provisioning requests. To configure a provisioning request a designer familiar with your business needs creates a

provisioning request definition, which binds the resource to a workflow. The designer can configure workflows that proceed in a *sequential* fashion, with each approval step being performed in order, or workflows that proceed in a *parallel* fashion. A parallel workflow allows more than one user to act on a workflow task concurrently.

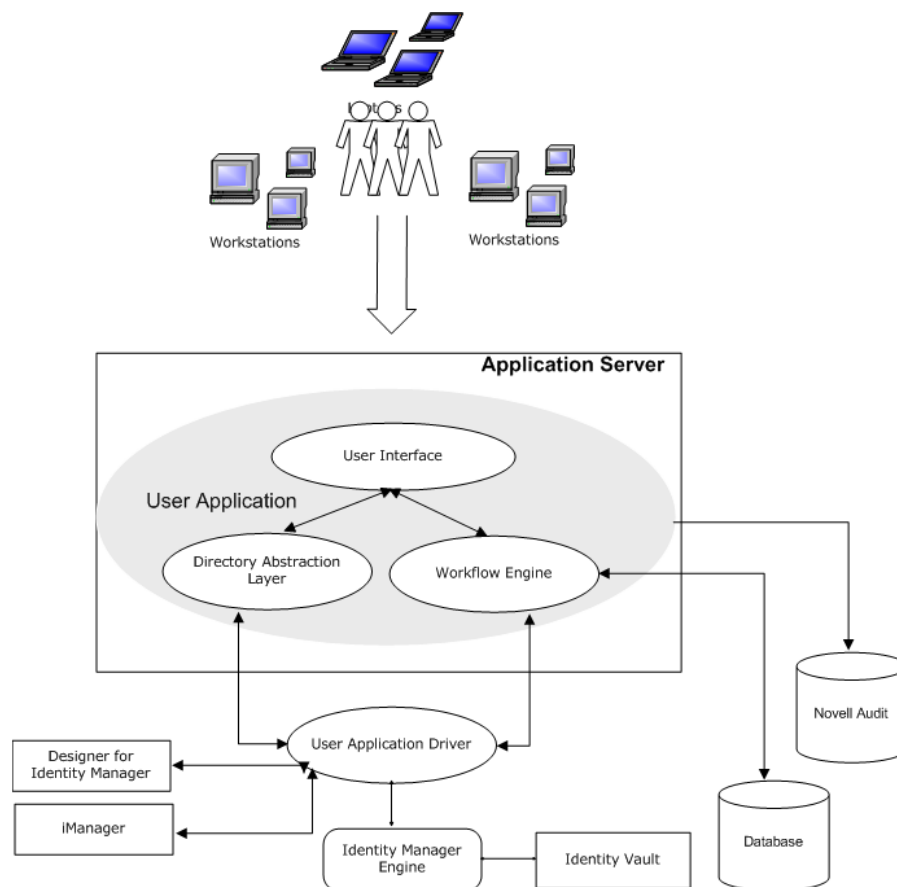
Identity Manager provides a set of Eclipse-based tools for designing the data and the flow of control within the workflows. In addition, Identity Manager provides a set of Web-based tools that provide the ability to configure existing provisioning requests, manage workflows that are in process, and define teams and team rights. For more information, see [Section 1.4, “Design and Configuration Tools,” on page 30](#)

The Provisioning Application Administrator is responsible for managing the workflow-based provisioning features of the User Application. For more information, see [Section 1.3, “User Application User Types,” on page 27](#)

1.2 User Application Architecture

The Identity Manager User Application relies on a number of independent components acting together. The core components are shown in [Figure 1-1](#).

Figure 1-1 User Application Core Components



1.2.1 User Interface

The Identity Manager User Application is a browser-based Java* application. It is comprised of a collection of JSR168-compliant portlets, JavaServer* Pages, and JavaServer Faces that run within a Java Web application on a J2EE*-compliant application server such as JBoss*. The User Application framework provides container services, such as managing window state, portlet preferences, persistence, caching, theming, logging, and acts as a security gatekeeper. The application server, on which the User Application runs, provides various services to the application as a whole, such as scalability through clustering, database access via JDBC*, and support for certificate-based security.

1.2.2 Directory Abstraction Layer

The directory abstraction layer provides a logical view of the Identity Vault data. You define a set of entities and their related attributes based on the Identity Vault objects that you want users to view, modify, or delete in the User Application. The Directory Abstraction layer:

- ◆ Performs all of the User Application's LDAP queries against the Identity Vault. This isolates presentation-layer logic from the Identity Vault, so that all requests for identity data go through the directory abstraction layer.
- ◆ Checks constraints and access control on data requests made via the User Application.
- ◆ Caches runtime configuration and entity-definition data obtained from the Identity Vault. See [Section 5.1.1, "Caching Management," on page 89](#)

You use the directory abstraction layer editor plug-in (available in Designer for Identity Manager) to define the structure of the directory abstraction layer data definitions. To learn more, see the section on the directory abstraction layer editor in the *Identity Manager User Application: Design Guide*.

1.2.3 Workflow Engine

The Workflow Engine (available with the Provisioning Module) is a set of Java executables responsible for managing and executing steps in an administrator-defined workflow and keeping track of state information (which is persisted in a database). When the necessary approvals have been given, the Provisioning System provisions the resource as requested.

During the course of workflow execution, the Workflow Engine can send one or more e-mail messages to notify users of changes in the state of the workflow. In addition, it can send e-mail messages to notify users when updates have been made to proxy, delegate, and availability settings.

You can edit an e-mail template in the Designer for Identity Manager or in iManager and then use this template for e-mail notifications. At runtime, the Workflow Engine retrieves the template from the directory and replaces tags with dynamic text suitable for the notification.

Additional details about the Workflow Engine, including how to configure and manage provisioning workflows, are in [Part V, "Configuring and Managing Provisioning Workflows," on page 279](#).

1.2.4 Application Server (J2EE-Compliant)

The application server provides the runtime framework in which the User Application, directory abstraction layer and Workflow Engine execute. The default is the JBoss application server. The

User Application is packaged as a Java Web Application Archive, or WAR file. The WAR is deployed to the application server.

1.2.5 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 User Application driver in the Identity Vault.)
- ♦ Novell Audit logs

1.2.6 User Application Driver

The User Application driver is an important enabling piece of the User Application. It is responsible for:

- ♦ Storing application-specific environment configuration data.
- ♦ Notifying the directory abstraction layer when important data values change in the Identity Vault. This causes the directory abstraction layer to update its cache.

If the Provisioning Module is installed, the User Application driver can be configured to:

- ♦ Allow events in the Identity Vault to trigger workflows.
- ♦ Communicate the success or failure of a workflow's provisioning activity back to the User Application database, which allows users to view the final status of their requests.
- ♦ Start 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).

Table 1-1 *Artifacts Stored in the User Application Driver*

Artifacts	Description
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 Identity Manager driver) must exist inside a driver set. The driver set is not automatically created by the User Application; you must create one, then create the User Application driver within it.
User Application	The User Application driver object is the container for a variety of artifacts. The User Application driver implements Publisher and Subscriber channel objects and policies. The Publisher channel is not used by the User Application but is available for custom user cases.

Artifacts	Description
<i>App Config Object</i>	<p>The AppConfig object is a container for the following User Application configuration objects.</p> <ul style="list-style-type: none"> ♦ RequestDefs: Container for Provisioning Request Definitions. The definitions stored here (as XML) represent the classes of requests that end users with appropriate rights can instantiate via the User Application. (Provisioning Module only) ♦ WorkflowDefs: :Container for Workflow objects, including design-time descriptions plus any template or unused flows. ♦ ResourceDefs: Container for Provisioned Resource definitions, including design-time descriptions plus any templates or unused targets. ♦ ServiceDefs: Container for Service Definition objects, which wrap Web Services called by workflows. ♦ DirectoryModel: Directory abstraction layer objects that represent different types of content of the Identity Vault that can be exposed in the User Application. ♦ AppDefs: Container for configuration objects that initialize the runtime environment, such as cache configuration information and e-mail notification properties. ♦ ProxyDefs: Container for proxy definitions. ♦ DelegateeDefs: Container for delegate definitions.

1.2.7 Designer for Identity Manager

Designer for Identity Manager provides a set of plug-ins you can use to define the directory abstraction layer objects and provisioning requests and their associated workflows. For more information, see [Section 1.4, “Design and Configuration Tools,” on page 30](#)

1.2.8 iManager

iManager provides a set of plug-ins you can use to configure and manage provisioning requests and their associated workflows. These tools also let you define provisioning teams and team rights. For more information, see [Section 1.4, “Design and Configuration Tools,” on page 30](#).

1.2.9 Identity Manager Engine

The Identity Manager engine provides the runtime framework that monitors events in the Identity Vault and connected systems. It enforces policies and routes data to and from the Identity Vault. The Identity Manager User Application is a connected system. Communication between the Identity Vault, the User Application’s directory abstraction layer, and the Workflow Engine occurs through the User Application driver.

1.2.10 Identity Vault

The Identity Vault is the repository for user data (and other identity data) plus the Identity Manager driver set and the User Application driver. Because the User Application relies on various Identity Vault objects, it’s necessary to extend the eDirectory™ schema to accommodate the custom LDAP

objects and attributes required by the User Application. The schema extension occurs automatically as part of the User Application install. The custom objects and attributes are populated with default values after the User Application driver is installed and activated.

1.2.11 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 [Chapter 3, “Setting Up Logging,” on page 71](#).

1.3 User Application User Types

The Identity Manager User Application users fall into these categories:

- ♦ [Administrators](#)
- ♦ [Users](#)
- ♦ [Designers](#)

1.3.1 Administrators

The User Application defines several types of administrative users. The administrative users defined in [Table 1-2](#) are defined at installation.

Table 1-2 *User Application Administrative Users*

User	Description
LDAP Administrator	<p>A user who has rights to configure the Identity Vault. This is a logical role that can be shared with other administrative user types.</p> <p>The LDAP administrator account is a proxy user for the user application to carry out tasks on the LDAP server that an ordinary logged-in user might not have permission to execute, such as creating a new user, group, or container. It represents credentials (username and password) used to bind to the Identity Vault to perform system LDAP operations, so these are the rights that the user application itself needs to run. The LDAP administrator needs:</p> <ul style="list-style-type: none">♦ Supervisor rights to the User Application Driver and all the objects it contains. You can accomplish this by setting the rights at the driver container level and making them inheritable.♦ Supervisor Entry rights to any of the users that are defined through the directory abstraction layer user entity definition. This should include Write attribute rights to objectClass and any of the attributes associated with the DirXML-EntitlementRecipient, srvprvEntityAux and srvprvUserAux auxiliary classes.♦ Read Rights to the container object cn=DefaultNotificationCollection, cn=Security. This object persists e-mail server settings used for automated provisioning e-mails. It can contain SecretStore credentials for authenticating to the e-mail server itself.

User	Description
User Application Administrator	<p>A user who has the rights to perform administrative tasks for the User Application. This user can:</p> <ul style="list-style-type: none"> ♦ Use the <i>Administration</i> tab of the User Application to manage the User Application. ♦ Use iManager to administer workflow tasks (such as enabling, disabling, or terminating an in-process workflows) ♦ Use iManager or Designer to create new provisioning requests, manage e-mail templates. ♦ Run reports on Novell Audit logging data. <p>This user does not have any special privileges on the <i>Requests & Approvals</i> tab of the user application.</p> <p>This user does not need any special directory rights because it controls application level access via the Administration page. Although a User Application Administrator has the ability to manage themes in the Administration page, the User Application uses the LDAP administrator credentials to modify the theme selections in the Identity Vault.</p> <p>Password self-service: One task of the User Application Administrator is to configure password self-service for the User Application. A feature of password self-service is password synchronization status. To enable the User Application Administrator to view the password synchronization status for other users (for troubleshooting or other reasons), it is recommended that you create a PasswordManagement group and assign one or more users to this group. The members of this group are allowed to view the password synchronization status of other users. If you choose to create this group, it must:</p> <ul style="list-style-type: none"> ♦ Be named PasswordManagement. ♦ Be given the privileges to the Identity Vault. The group must have rights to read the user's eDirectory object attribute for users whose password synchronization status they need to view.
Provisioning Application Administrator	<p>A user who is intended to allow you to delegate provisioning management tasks to a business user without giving him or her full administration rights to the User Application. By default, the Provisioning Administrator cannot access the Administration page, but he or she has full rights to the <i>Request & Approvals</i> tab. For example, when the Provisioning Application Administrator logs in, he or she does not need to select a team because all users are considered to be his or her team members.</p>

iManager Administrators

In addition to the users and their associated tasks above, Identity Manager includes administrators that use iManager to:

- ♦ Create new provisioning requests and workflows.
- ♦ Define teams.
- ♦ Define or manage e-mail templates.
- ♦ Administer workflow tasks (such as enabling, disabling, or terminating in-process workflows).

The user that performs these tasks can be one of the administrators listed above, or a different user that has been given the privileges to perform these tasks.

To create or edit or edit workflow objects in iManager, the user needs the following rights on the RequestDefs.AppConfig container for the specific User Application driver.

- ♦ [Entry Rights] Supervisor or Create.
- ♦ [All Attribute Rights] Supervisor or Write.

To initiate a workflow, the user must have Browse [Entry Rights] on the RequestDefs.AppConfig container for the specific User Application driver or individually per request definition object if you are using a delegated model.

1.3.2 Designers

Designers use the Designer for Identity Manager to customize the User Application for your enterprise. Designer is a tool aimed at information technology professionals such as enterprise IT developers, consultants, sales engineers, architects or system designers, and system administrators who have a strong understanding of directories, databases, and their information environment and who act in the role of a designer or architect of identity-based solutions.

To create or edit or edit workflow objects in Designer, the user needs the following rights on the RequestDefs.AppConfig container for the specific User Application driver.

- ♦ [Entry Rights] Supervisor or Create.
- ♦ [All Attribute Rights] Supervisor or Write.

To initiate a workflow, the user must have Browse [Entry Rights] on the RequestDefs.AppConfig container for the specific User Application driver or individually per request definition object if you are using a delegated model.

1.3.3 Users

The user is the person who views and interacts with the User Application's *Identity Self-Service* and the *Requests & Approval* tab (if the Provisioning Module is installed). A user can be:

- ♦ An *authenticated user* (such as an employee, a manager, or a delegate or proxy for an employee or manager). A *delegate user* is a 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 someone else. 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.
- ♦ An *anonymous or guest user*. The anonymous user can be either the public LDAP guest account or a special account set up in your Identity Vault. The User Application Administrator can enable anonymous access to some features of the *Identity Self-Service* tab (such as a search or create request). In addition, the User Application Administrator can create pages that allow the user to request a resource. See [Table 1-7 on page 33](#) for information on configuring anonymous access.

The user's capabilities within the User Application depend on what features the User Application Administrator has enabled for them. They can be configured to:

- ♦ View hierarchical relationships between User objects by 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.

If the Provisioning Module is installed, the User Application can be configured so that users can:

- ♦ 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 availability.
- ♦ Enter proxy mode in order to claim tasks on behalf of another.
- ♦ View team tasks, request team resources, and so forth (managers only).

1.4 Design and Configuration Tools

The various administrators can use the following tools to design and configure the Identity Manager User Application.

Table 1-3 *Tools for Designing and Configuring the User Application*

Tool	Purpose
Designer for Identity Manager	<p>A powerful, graphical toolset for configuring and deploying Identity Manager. The following plug-ins are designed to help you configure the User Application:</p> <ul style="list-style-type: none"> ♦ Directory Abstraction Layer editor: Lets you define the Identity Vault objects needed for your User Application. ♦ Provisioning Request Definition editor: Lets you create workflows for provisioning request definitions. Also allows you to customize the forms by which users make and approve requests and e-mail templates. ♦ Provisioning view: Lets you import, export, deploy, and migrate directory abstraction layer and provisioning requests to the User Application driver. <p>For more information, see the <i>Identity Manager User Application: Design Guide</i>.</p>

Tool	Purpose
iManager (for Provisioning Module only)	<p>A Web-based administration console. The following plug-ins are designed to help you configure and administer the User Application:</p> <ul style="list-style-type: none"> ◆ Provisioning Request Configuration plug-in: Lets you bind the provisioning request definition to a provisioned resource, specify the runtime characteristics of the associated workflow and enable its use. ◆ Workflow Administration plug-in: 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 stopped and cannot be restarted. ◆ Provisioning Team plug-in: Lets you define the characteristics of a team. A team identifies a group of users and determines who can manage provisioning requests and approval tasks associated with this team. The team definition consists of a list of team managers, team members, and team options. ◆ Provisioning Team Request plug-in: Lets you specify the request rights for a team. The team requests objects specify a list of requests that fall within the domain of a team, as well as the rights given to the team managers. The request rights specify actions that team managers can perform on the provisioning requests and tasks. <p>For more information, see Part V, “Configuring and Managing Provisioning Workflows,” on page 279</p>
User Application Admin tab	<p>A Web-based administration console that allows you to configure, manage, and customize the User Application. It contains the following pages:</p> <ul style="list-style-type: none"> ◆ Application Configuration: Lets you configure caching, LDAP parameters, logging, themes, password module setup ◆ Page Administration: Lets you create new portlets or customize existing Identity Self-Service pages ◆ Portlet Administration: Lets you create new or customize the existing portlets used on the Identity Self-Service pages. ◆ Provisioning: Lets you configure Delegation, Proxy, Tasks, Digital Signature service, and engine and cluster settings. ◆ Security: Lets you define who has Provisioning Administrator and User Application Administrator privileges. <p>For more information, see Part III, “Administering the User Application,” on page 81.</p>

Tool	Purpose
<code>lreport.exe</code> (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 <code>lreport.exe</code> log report tool (Windows* only) is one way to generate the reports. You can also use other methods to create the reports. See Chapter 3, “Setting Up Logging,” on page 71 for details.

1.5 What’s Next

Now that you have learned about the features and architecture of the Identity Manager User Application, you can start to customizing it as needed for your own business needs. Typically, you’ll be:

- ♦ Customizing the user interface and identity self-service features. See [Table 1-5 on page 32](#).
- ♦ Setting up the requests and approval features (if provisioning is installed). See [Table 1-7 on page 33](#).
- ♦ Setting up your production environment. See [Table 1-6 on page 33](#).

Table 1-4 *Customizing the User Interface and Identity Self-Service Features*

To learn about	See
Setting up directory abstraction layer objects	<i>Identity Manager User Application: Design Guide</i>
Customizing the Identity Self-Service pages	Part IV, “Portlet Reference,” on page 205
Adding new pages and setting page security	Chapter 6, “Page Administration,” on page 139
Creating custom instances of the identity portlets	Chapter 7, “Portlet Administration,” on page 173
Changing the User Application’s theme or branding	Section 5.1.6, “Theme Administration,” on page 107
Localizing the User Application user interface	Section 2.8, “Localizing Text,” on page 69
Enabling password self-service	Section 5.3, “Password Management Configuration,” on page 120

Table 1-5 *Setting Up the Requests and Approvals Features*

To learn about	See
Creating provisioning requests	<i>Identity Manager User Application: Design Guide</i> and Chapter 17, “Configuring Provisioning Request Definitions,” on page 295
Customizing request and approval forms	<i>Identity Manager User Application: Design Guide</i>
Defining teams	Chapter 19, “Configuring Provisioning Teams,” on page 347

To learn about	See
Defining e-mail templates	<i>Identity Manager User Application: Design Guide</i> and Section 18.4, “Working with E-Mail Templates,” on page 334

Table 1-6 *Setting Up the User Application Production Environment*

To learn about	See
Your production environment topology	Section 2.1, “Topology,” on page 37
Setting up security	Section 2.2, “Security,” on page 40
Setting up digital signature support	Section 2.3, “Digital Signature Configuration,” on page 43
Performance tuning strategies	Section 2.6, “Performance Tuning,” on page 52
Setting up a cluster	Section 2.7, “Clustering,” on page 58
Setting up logging	Chapter 3, “Setting Up Logging,” on page 71

Table 1-7 *User Application Configuration for Guest Access*

To learn about	See
Guest or anonymous accounts	Section 2.4, “Enabling Anonymous or Guest Access to the User Application,” on page 49
Allowing anonymous users to self-register	Section 11.4, “Configuring the Create Portlet for Self-Registration,” on page 216
Allowing anonymous access to the directory search	Section 15.3, “Configuring Search List for Anonymous Access,” on page 276
Allowing anonymous access to the My profile or Organizational charts	Section 12.6, “Setting up Detail for Anonymous Access,” on page 233 and Section 13.3, “Configuring Org Chart for Guest Access,” on page 262
Allowing anonymous access to a workflow	Chapter 14, “Resource Request Portlet,” on page 263

Configuring the User Application Environment



These sections describes how to configure various aspects of the Identity Manager User Application environment to meet the needs of your organization.

- ♦ [Chapter 2, “Designing the Production Environment,” on page 37](#)
- ♦ [Chapter 3, “Setting Up Logging,” on page 71](#)

Designing the Production Environment

2

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

This section is organized as follows:

- ♦ [Section 2.1, “Topology,” on page 37](#)
- ♦ [Section 2.2, “Security,” on page 40](#)
- ♦ [Section 2.3, “Digital Signature Configuration,” on page 43](#)
- ♦ [Section 2.4, “Enabling Anonymous or Guest Access to the User Application,” on page 49](#)
- ♦ [Section 2.5, “Configuring Forgotten Password Self-Service,” on page 50](#)
- ♦ [Section 2.6, “Performance Tuning,” on page 52](#)
- ♦ [Section 2.7, “Clustering,” on page 58](#)
- ♦ [Section 2.8, “Localizing Text,” on page 69](#)

2.1 Topology

Each major subsystem can have many instances and many ways of connecting. Not every possible layout is supported. This section includes three subsections that describe the possibilities and why some configurations are preferred over others.

- ♦ [Section 2.1.1, “Minimal Design,” on page 37](#)
- ♦ [Section 2.1.2, “High Availability Design,” on page 38](#)
- ♦ [Section 2.1.3, “Design Constraints,” on page 39](#)

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 machine. But you would not do that in the real world, for a variety of reasons including security, maintainability, and performance. 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 application is responsible for capturing event information (and possibly a good deal of other information) from the User Application environment at runtime. It might also be doing double duty as a persistence store for other applications in your company. For a variety of reasons, you probably do 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. Consider putting the Identity Vault on a dedicated machine. You probably do not want another high-traffic system, such as JBoss with a deployment of the User Application, running on the same machine as the Identity Vault.

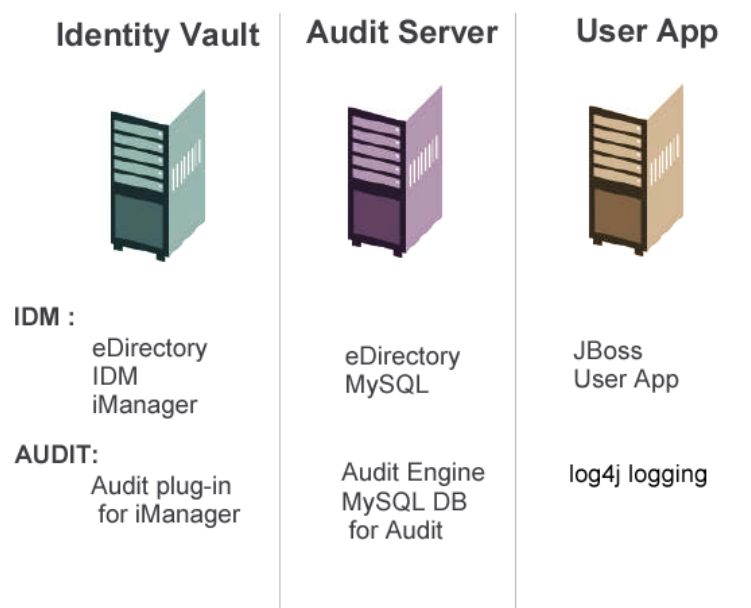
Database: If this instance of MySQL (or other supported database) is also your Novell® Audit database, it is probably on a dedicated machine. The User Application uses this component in the following ways:

- ♦ 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 should probably run this piece on a dedicated machine.

These considerations suggest the following minimal three-machine configuration:

Figure 2-1 *Minimal Three Machine Configuration*



2.1.2 High Availability Design

Clustering for high availability and capacity is discussed in [Section 2.7, “Clustering,” on page 58](#). 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 section “High Availability” in the *Identity Manager Administration Guide*. A comprehensive procedure for setting up such a system using SUSE® Linux is 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 so that each node runs one User Application instance. The instances are all coequals (peers).
- ♦ Automatic failover is supported. An interrupted workflow can resume after the loss of a cluster node.

See [Section 2.7, “Clustering,” on page 58](#) for more information.

2.1.3 Design Constraints

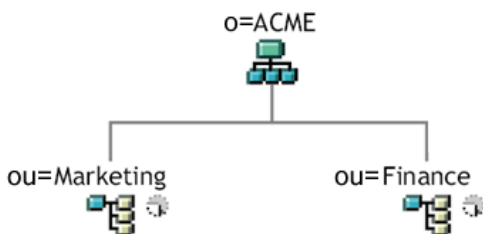
The two most important architectural constraints are:

- ♦ No User Application instance can service (search, query, add users to, and so forth) more than one user container. Also, a user container association with an application 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:

Figure 2-2 Sample Organizational Structure



During installation of the User Application, you are asked to specify the top-level user container that your installation looks 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) are 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

This section includes the following topics:

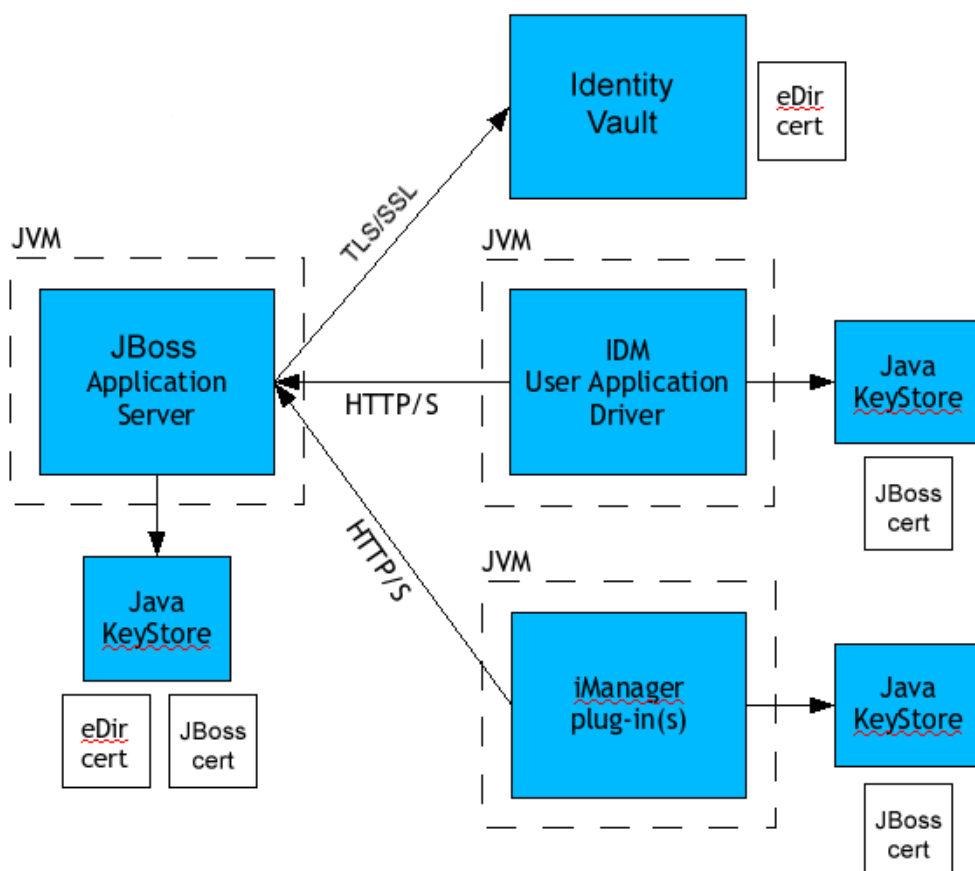
- ♦ [Section 2.2.1, “Security Overview,” on page 40](#)
- ♦ [Section 2.2.2, “Self-Signed Certificates,” on page 41](#)
- ♦ [Section 2.2.3, “Turning on SSL in JBoss,” on page 42](#)
- ♦ [Section 2.2.4, “Turning on SOAP Security,” on page 42](#)
- ♦ [Section 2.2.5, “Mutual Authentication,” on page 43](#)
- ♦ [Section 2.2.6, “Encryption of Sensitive User Application Data,” on page 43](#)

2.2.1 Security Overview

Moving from pre-production to production usually involves hardening the security aspects of the system. In sandbox testing, you might use regular HTTP to connect the User Application driver to JBoss, or you might use a self-signed certificate (as a temporary measure) for driver/app-server communication. In production, on the other hand, you 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 following diagram.

Figure 2-3 Identity Manager User Application Environment



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 might 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 to use a certificate issued by a trusted certificate authority (CA) such as Verisign.

2.2.2 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, use the following 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
```

2.2.3 Turning on SSL in JBoss

The User Application uses HTML forms for authentication. As a result, user credentials are exposed during login. We strongly recommend that you enable SSL in JBoss to protect sensitive information. To enable SSL in JBoss:

- 1 Find the `jbossweb-tomcat55.sar` file under `[IDM]/jboss/server/IDM/deploy/`.
- 2 In `jbossweb-tomcat55.sar`, find `server.xml` and open that file in a text editor.
- 3 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" />
```
- 4 Install a trusted certificate by following the instructions given at [SSL Setup \(http://www.jboss.org/wiki/Wiki.jsp?page=SSLSetup\)](http://www.jboss.org/wiki/Wiki.jsp?page=SSLSetup).

2.2.4 Turning on SOAP Security

- 1 In `IDM.war`, find the `web.xml` file and open it in a text editor.
- 2 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>
```
- 3 Save the file and the archive, then restart JBoss.

2.2.5 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®. For example, if your organization has a need to allow users to log in through a user certificate rather than a password, you can do this by adding iChain to your environment.

See your Novell representative for more information.

2.2.6 Encryption of Sensitive User Application Data

Any sensitive information associated with the User Application that is stored persistently is encrypted by using the symmetric algorithm AES-128. The master key itself is protected by password-based cryptography using PBESWithSHA1AndDESede. The password is never persisted or stored out of memory, and thus cannot be stolen.

Information that is encrypted includes (but is not limited to):

- ♦ LDAP administrator user password
- ♦ LDAP guest user password
- ♦ DSS trusted CA keystore password
- ♦ DSS signature key keystore password
- ♦ DSS signature key entry password
- ♦ Novell Audit signature key

2.3 Digital Signature Configuration

This section provides instructions on configuring your environment to take advantage of the digital signature support provided with the Identity Manager User Application.

NOTE: If you want to use the Novell Certificate Server™ (Novell PKI infrastructure) for digital signing features, you need to use eDirectory 8.8 or later. The digital signature functionality requires PKI 3.1, which ships with eDirectory 8.8.

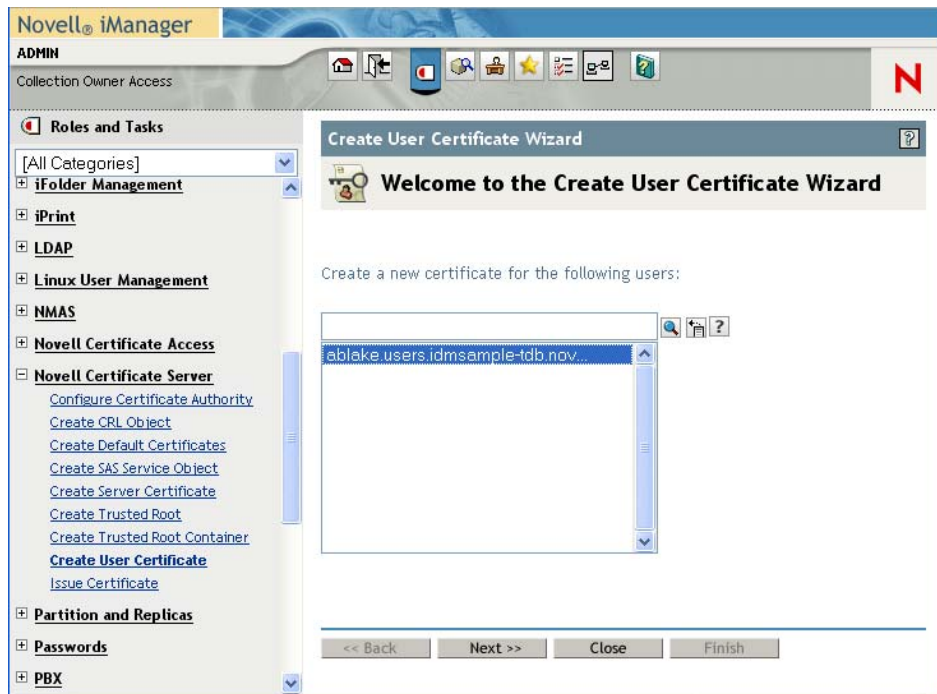
This section includes the following topics:

- ♦ [Section 2.3.1, “Setting Up the User Certificates,” on page 43](#)
- ♦ [Section 2.3.2, “Configuring JBoss,” on page 47](#)
- ♦ [Section 2.3.3, “Configuring the User Application,” on page 48](#)
- ♦ [Section 2.3.4, “Configuring the Provisioning Request Definitions,” on page 48](#)

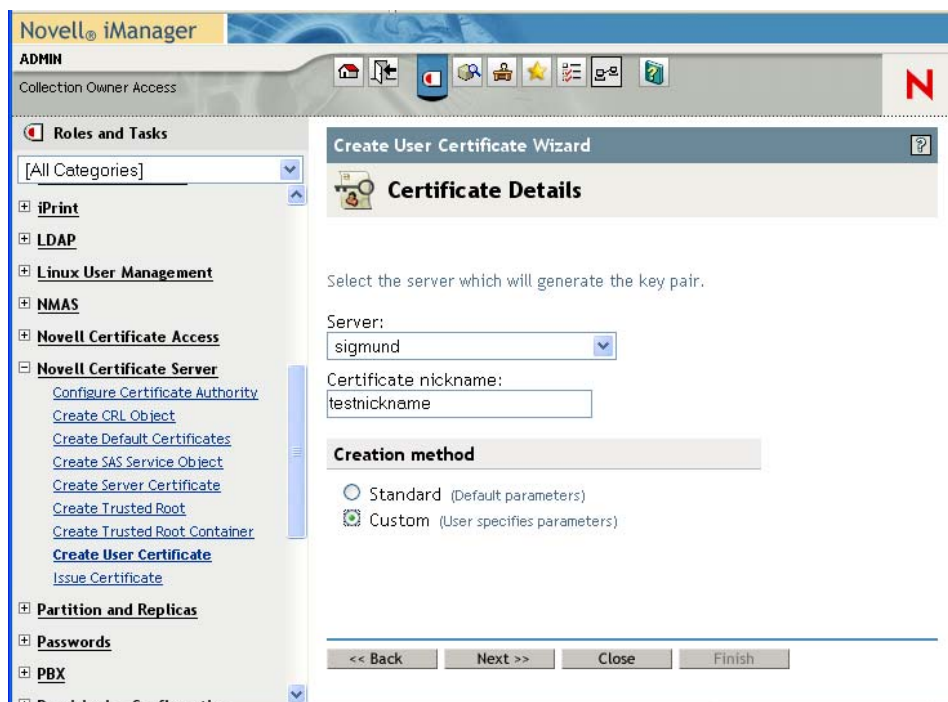
2.3.1 Setting Up the User Certificates

- 1 Create the user certificates using iManager.
 - 1a Log in as an administrator.
 - 1b Under *Novell Certificate Server*, select *Create User Certificate*.

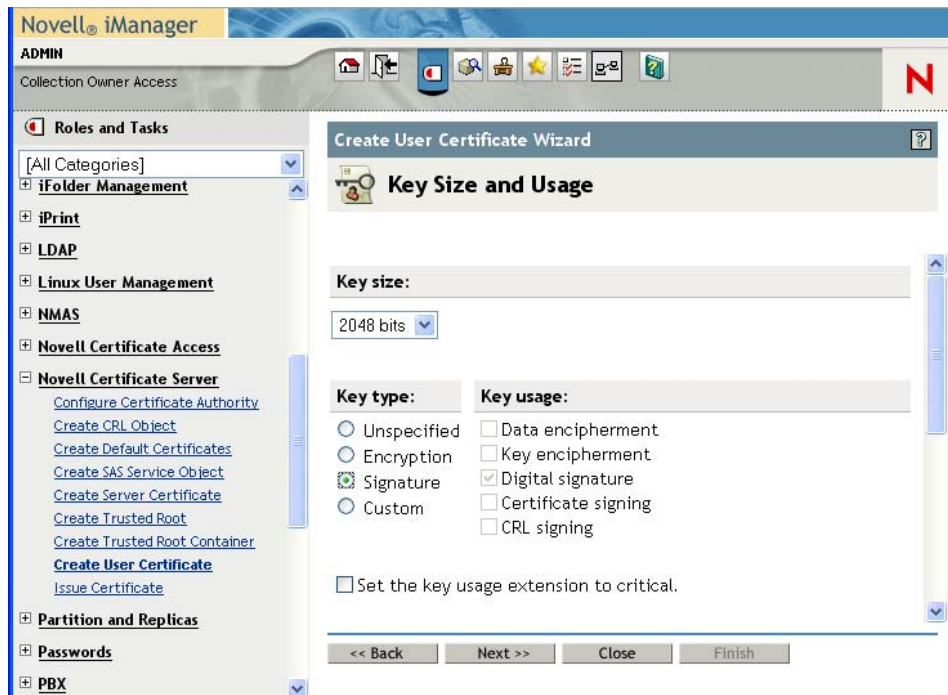
- 1c Select the users for whom you want to create certificates and click *Next*.
You can use the Object Selector or Object History to pick the users.



- 1d Select the server and specify the certificate nickname. Specify *Custom* as the creation method and click *Next*.



- 1e** Specify a key size of 1024 or 2048 bits, depending on which size suits your requirements. Set the key type to *Signature*. Leave other settings as is and click *Next*.



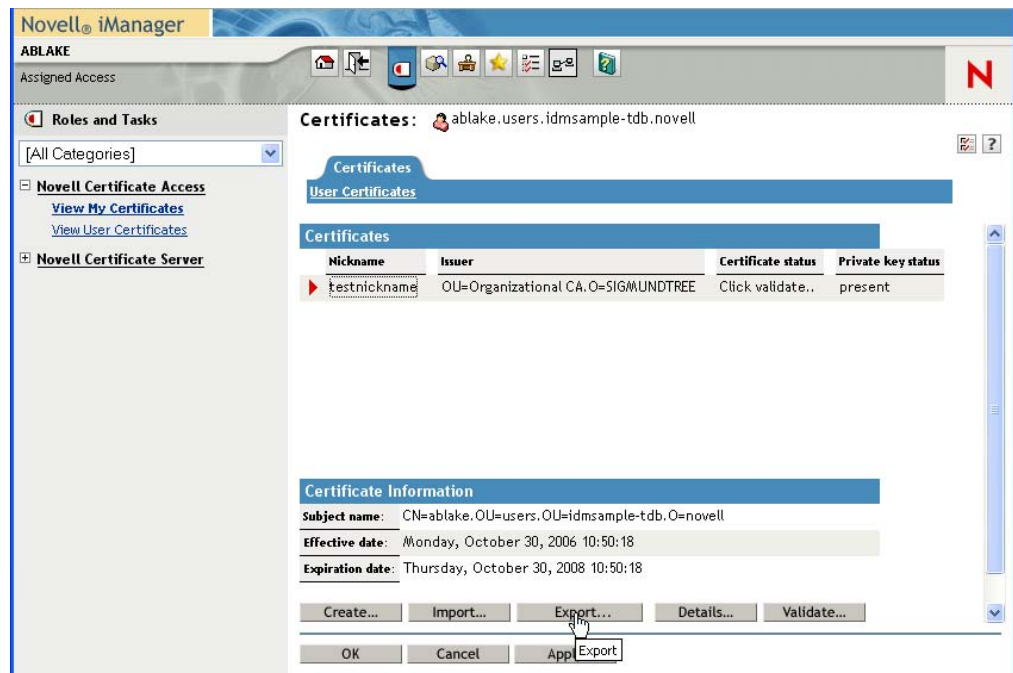
- 1f** If you're using the default configuration, leave the certificate parameters as is and click *Next*.

To enable certificate revocation list (CRL) support, select Custom and check the CRL signing checkbox.

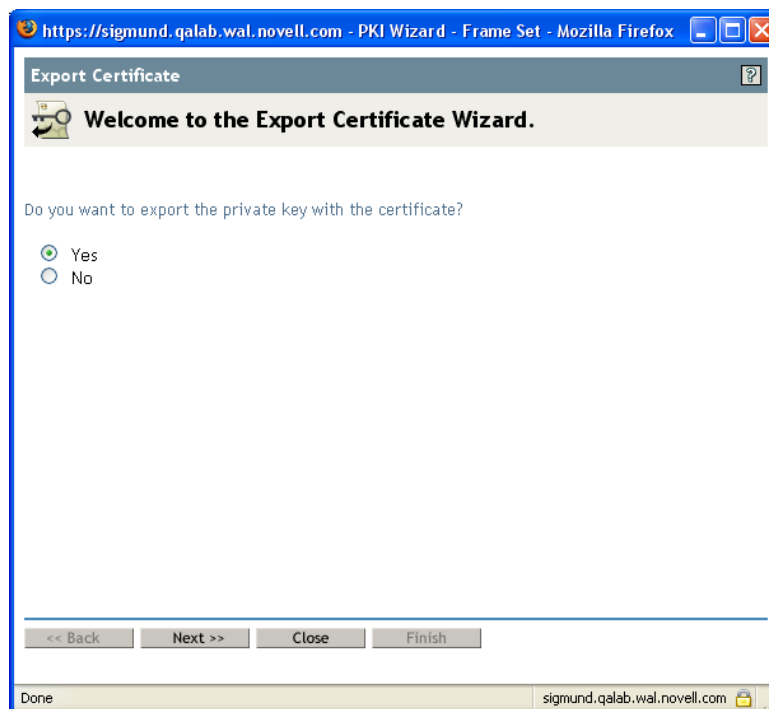
For complete details on CRL configuration, see the Novell Certificate Server documentation.

- 1g** Click *Finish*.
- 1h** Log out.
- 2** Export the user certificate as a PFX file that contains the private key.
- 2a** Log in as the user for whom you want to export a certificate.
- 2b** Under *Novell Certificate Access*, select *View My Certificates*.

2c Select a certificate and click the *Export* button.



2d In the Export Certificate Wizard, click *Yes* to indicate that you want to export the private key with the certificate. Then click *Next*.



2e Enter a password to protect the private key and click *Next*.

2f Select *Export the certificate into the browser* if you do not have a card reader. Otherwise, click on the link that says *Save exported certificate to a file*.

You can also import to the browser later. Therefore, you might want to click on *Save exported certificate to a file* to import to a different browser.

2g Click *Save to Disk* to save the file rather than opening it.

2h Click *Close*.

3 If you're using a smart card, install the smart card reader driver.

4 Install the software needed to transfer certificate information to the smart card. For example, to get the smart card middleware software provided by cryptovision (cv act sc/interface), or download an evaluation copy of their product or documentation, go to: <http://www.cryptovision.com/idmdigsig.html>.

NOTE: You need to install version 3.3 or higher of the cryptovision middleware software. To transfer certificate information to the smart card, you need the administration software. The cryptovision software is not supported on Linux*.

5 Import the key pair (certificate) to the smart card:

If you are planning to use browser certificate support, rather than the smart card, you can skip steps 3 through 5 above. Certificates can be imported into a browser using iManager or the browser certificate management user interface. The cryptovision applet supports Internet Explorer and Firefox* running on Windows only.

2.3.2 Configuring JBoss

To configure the JBoss server, follow these steps:

1 Copy the following JARs to the `JBOSS_HOME/server/idm-server/lib` directory:

- ♦ `dom.jar`
- ♦ `xmldigsig.jar`
- ♦ `xmlsec.jar`

You can download `dom.jar`, `xmldsig.jar`, and `xmlsec.jar` from <http://java.sun.com>. These JARs are included with the Web Services Developer Pack.

For cryptovision, you also need `SafXVerifier.jar`. For details on downloading `SafXVerifier.jar`, see <http://www.cryptovision.com/idmdigsig.html>.

2 Copy `xmlsigner.war` to the `JBOSS_HOME/server/idm-server/deploy` directory.

For details on downloading `xmlsigner.war`, see <http://www.cryptovision.com/idmdigsig.html>.

3 Export the trusted root and all intermediate certificates (using iManager) and import them into the key store specified in your system's local configuration using the `keytool` command:

```
keytool -import -trustcacerts -file certFile
```

The `certFile` is a fully qualified path to the certificate file.

If you're using the Novell Certificate Server, you do not need to export the trusted root.

4 Start the User Application Configuration utility by running the `configupdate` script (`configupdate.bat` on Windows).

5 Click *Show Advanced Options*.

- 6 Under *Trusted Key Store*, type the path to the certificate file in the *Trusted Store Path*. Also, type your password in the *Keystore Password* field. The default password is `changeit`.
The Trusted Key Store contains all trusted signers' certificates used to validate digital signatures.

NOTE: If you're using the Novell Certificate Server, you can simply paste the complete string (for example, `C:\Program Files\Java\jdk1.5.0_08\jre\lib\security\cacerts`) from the *Keystore Path* field under *eDirectory Certificates* to the *Trusted Store Path* under *Trusted Key Store*. You can also paste the *Keystore Password* to the *Trusted Store Password* field.

- 7 If you are using OCSP, under *Miscellaneous*, type the URI for OCSP in the *OCSP URI* field. This value is used to update the status of trusted certificates online. The URI points to the access point for the Online Certificate Status Protocol server.

2.3.3 Configuring the User Application

To configure digital signature support for the User Application, you need to use the *Digital Signature Service* page on the *Administration* tab within the User Application. For details, see [Section 8.3, "Configuring the Digital Signature Service," on page 193](#).

2.3.4 Configuring the Provisioning Request Definitions

You can use Designer for Identity Manager or iManager to configure digital signature support for your provisioning request definitions. The basic requirements for digital signature support are the same whether you perform your configuration steps in Designer or iManager.

To configure a provisioning request definition to support digital signatures, you need to:

- 1 Indicate whether a digital signature is required to initiate the provisioning request.
- 2 Indicate whether a digital signature is required for each approval step within the workflow. Because each approval step might have more than one outgoing link, you need to specify whether a digital signature is required for each link.

After you have indicated whether a digital signature is required to initiate a request or perform an approval step, you need to also specify the following for each request or approval step where a digital signature is required:

Table 2-1 *Digital Signature Settings*

Setting	Description
Digital Signature Type	<p>Specifies whether the digital signature uses data or form as its type:</p> <ul style="list-style-type: none">♦ Data: Specifies that the XML signature serves as the user agreement. When Data is selected, the XML data is written to the audit log. The user can preview XML data before submitting a signature.♦ Form: Specifies that a PDF document that includes the digital signature declaration be generated. This document serves as the user agreement. The user can preview the generated PDF document before submitting a request or approval. When Form is selected, the PDF document (encapsulated in XML) is written to the audit log.
Digital Signature Declaration	<p>Specifies a digital signature confirmation string that confirms the user's signature.</p>

For details on configuring provisioning request definitions in Designer, see the *Identity Manager User Application: Design Guide*. For details on configuring provisioning request definitions in iManager, see [Chapter 17, “Configuring Provisioning Request Definitions,” on page 295](#).

2.4 Enabling Anonymous or Guest Access to the User Application

To enable anonymous or guest user to access the Identity Self-Service features of the User Application, follow the steps outlined in [Table 2-2](#).

Table 2-2 *Setting Up Anonymous Access*

Task	For more information
Determine the guest account you want to use for the anonymous access.	See “Establishing the Guest Account” on page 50 .
Assign the proper Identity Vault rights to the guest user.	Define rights based on the features you want expose to non-authenticated Web application users. In the User Application, you can expose identity portlets such as the search, detail, or chart and create portlet. You can also allow users to initiate a workflow. In these cases the guest user account is used to bind to eDirectory and perform the underlying LDAP operation.
To perform Identity Self-Service tasks, create new pages and portlets specifically for guest access.	See Part IV, “Portlet Reference,” on page 205 .

Task	For more information
To perform a resource request, use the resource request portlet.	See Chapter 14, “Resource Request Portlet,” on page 263.

2.4.1 Establishing the Guest Account

There are two ways to support anonymous or guest access to the User Application. You can:

- ♦ Setup a dedicated user account. Set up the permissions that are needed for the activities of that anonymous user. Remember that if this user is inside the user container, this guest account is returned during searches of the tree. To prevent this, consider putting the guest user outside the user container.
- ♦ Use the public LDAP guest account that corresponds to the [Public] object in eDirectory. The default access for [Public] is Browse rights to the entire tree. You must set up whatever permissions are necessary for this user to perform the guest tasks you provide. If you do not want all anonymous users to perform some of these tasks, this might not be the correct option for your installation.

The User Application allows you to specify only one type of anonymous user, and you are required to specify that user during installation. The installation options are:

- ♦ **Use Public Anonymous Account:** This uses the LDAP guest account.
- ♦ **LDAP Guest:** This is the dedicated user account.

You can modify your installation choice by running the configupdate utility after the installation is complete.

2.5 Configuring Forgotten Password Self-Service

The User Application provides forgotten password self-service that includes prompting for challenge responses, displaying a password hint, or allowing a password change, as needed. By default, forgotten password features are available to users inside your corporate firewall through the deployed User Application WAR, but you can also set up your User Application so that the forgotten password self-service features are accessed from a separate (or external) password management WAR that you can deploy on a separate system located inside or outside your corporate firewall. By default, the installation program generates and installs a WAR called `IDMPwdMgt.WAR` in the installation root directory. `IDMPwdMgt.WAR` contains only the password self-service software along with one theme. The theme is the default User Application theme called *Blue Gloss*. [Table 2-3](#) describes how to configure the User Application to use the external password management WAR.

Table 2-3 Steps for Enabling an External Password Management WAR

Task	Description
Install the User Application and specify that you want to <i>Use External Password WAR</i> , then specify values for the <i>Forgot Password Link</i> and the <i>Forgot Password Return Link</i> .	<p>When you specify this option, the install program puts the <code>IDMPwdMgt.WAR</code> in the installation <code>/bin</code> directory. It renames <code>IDMPwdMgt.WAR</code> based on the value you specify in the <i>Forgot Password Link</i> as described in Identity Manager Install Guide.</p> <p>For the <i>Forgot Password Link</i> configuration parameter, you'll specify the location for the external password WAR. Include the application server host and its secure port, for example <code>https://externalpwdhost:8443/IDMPwdMgt/jsp/pwdmgt/ForgotPassword.jsf</code>.</p> <p>For the <i>Forgot Password Return Link</i>, you'll supply the path that the external Password Management WAR uses to call back the User Application, (it uses a Web Service), for example <code>https://idmhost:8443/IDM</code>.</p> <p>You can update the configuration after installation using the <code>configupdate</code> tool.</p> <p>If you want to change the link locations, you can do so in the <i>User Application Administration</i> tab.</p>
Deploy the separate WAR to a JBoss application server	<p>The JBoss application server must be configured to support SSL. See Section 2.2.3, "Turning on SSL in JBoss," on page 42. In addition:</p> <ul style="list-style-type: none"> ♦ If the external password management WAR is deployed outside the firewall, make sure that the firewall's SSL port is open to allow communication between both application server hosts. ♦ The application server that hosts the external password management WAR must have the server certificate of the application server hosting the core User Application. Use the <code>keytool import</code> command to import the server certificate to the keystore (<code>cacerts</code>) of the JRE used by the application server hosting the external password WAR. The <code>keytool</code> command has this syntax: <pre>keytool -import -file certname.cer -keystore cacerts -storepass changeit -alias uacerts</pre>
Do you want to customize the theme for the external WAR?	For more information, see "Customizing the Theme for External Password WAR" on page 113 .

The external WAR location is saved to the

```
configuration.AppDefs.AppConfig.driver.driverset as
```

```
<property>  
<key>com.novell.pwdmgmt.login.PREF_FORGOT_PSWD_LINK_KEY</key>  
<value>https://externalpwdhost:8443/IDMPwdMgt/jsps/pwdmgmt/  
ForgotPassword.jsf</value>
```

The return location is saved to the

```
configuration.AppDefs.AppConfig.driver.driverset as  
  
<property>  
<key>com.novell.pwdmgmt.login.PREF_FORGOT_PSWD_RETURN_LINK_KEY</  
key>  
<value>https://IDMhost:8443/IDMProv</value>  
</property>
```

The return location is saved to the `userAppURL` property in `External WAR/WEB-INF/faces-managed-beans.xml`, for example

```
<property-name>userAppURL</property-name>  
<property-class>java.lang.String</property-class>  
<value>https://IDMhost:8443/IDMProv</value>
```

2.5.1 Accessing an External Password Management WAR

Users can go to the *Forgot Password* page in the external password WAR directly from a browser like this:

```
https://externalpwdhost:8443/IDMPwdMgt/jsps/pwdmgmt/  
ForgotPassword.jsf.
```

When accessed directly, the external password WAR checks the `WEB-INF\faces-managed-beans.xml` for this entry:

```
<property-name>userAppURL</property-name>  
<property-class>java.lang.String</property-class>  
<value>https://151.155.254.69:8443/IDM</value>
```

The external WAR uses the `userAppURL` entry to call the Web Service that handles the forgot password functionality in the User Application WAR.

Users can access the *Forgot Password* page by clicking the *Forgot Password?* link in the User Application's *Login* page. The User Application redirects the user to the external password management WAR based on the value specified for the *Forgot Password link*. The external password management WAR uses the *Forgot Password Return Link* value to call back to the User Application.

2.6 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.

See the following sections for information:

- ♦ [Section 2.6.1, “Logging,” on page 53](#)
- ♦ [Section 2.6.2, “Identity Vault,” on page 54](#)
- ♦ [Section 2.6.3, “JVM,” on page 55](#)
- ♦ [Section 2.6.4, “Session Timeout Value,” on page 55](#)
- ♦ [Section 2.6.5, “Tuning JBoss,” on page 56](#)
- ♦ [Section 2.6.6, “Using Secure Sockets for User Application Connections to the Identity Vault,” on page 56](#)

2.6.1 Logging

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

NOTE: The kinds of events you can log, and how to enable or disable logging, are covered in [Chapter 3, “Setting Up Logging,” on page 71](#).

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

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

Assigning a value to `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 set that threshold to ERROR, WARN, or FATAL unless 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*.

2.6.2 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 object class types and new attributes pertaining to the User Application. User-application-specific attributes are by default not indexed. For better performance, you might find it useful to index some of those attributes (and perhaps a few traditional LDAP attributes as well), particularly if your user container contains over 5,000 objects.

The general idea is to index only those attributes that you know are regularly queried, which could be different attributes in different production environments. The only way to know which attributes are heavily used is to collect predicate statistics at runtime. The collection process itself degrades performance, however.

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

- ♦ Use ConsoleOne® 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 ConsoleOne. You can create and manage indexes in iManager with eDirectory *Maintenance > Indexes*. For example, if you know that users of your org chart are likely to 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 “Tuning eDirectory” in *Novell’s Guide to Troubleshooting eDirectory* by Peter Kuo and Jim Henderson (QUE Books, ISBN 0-7897-3146-0).

Also read about performance tuning in “Maintaining Novell eDirectory” in the *eDirectory Administration Guide*.

2.6.3 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 maximum 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 might be needed to determine the optimal setting for your particular environment.

NOTE: JBoss and Tomcat performance tuning tips are at <http://wiki.jboss.org/wiki/Wiki.jsp?page=JBossASTuningSliming> (<http://wiki.jboss.org/wiki/Wiki.jsp?page=JBossASTuningSliming>)

2.6.4 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 box 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 runs. 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 improves performance and scalability of the Web application.

Consider the following when adjusting the session timeout:

- ♦ Longer session time-outs can cause the JBoss server to run out of memory if many users 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) can cause system performance degradation.
- ♦ If the server starts experiencing out-of-memory errors, and the JVM heap and garbage collection tuning parameters have already been optimally tuned for the server and usage environments, consider lowering the session timeout.

To adjust the session timeout value, 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, save the file and the archive, and restart the server.

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

2.6.5 Tuning JBoss

By default, the JBoss deployment scanner runs every five seconds. For a production server, this is typically not necessary and might impact performance. You should consider changing the scan period so that the deployment scanner runs less frequently, or turn the deployment scanner off entirely. For information about configuring the deployment scanner, see

[ConfiguringTheDeploymentScannerInConfjbossSystem](http://wiki.jboss.org/wiki/Wiki.jsp?page=ConfiguringTheDeploymentScannerInConfjbossSystem) (<http://wiki.jboss.org/wiki/Wiki.jsp?page=ConfiguringTheDeploymentScannerInConfjbossSystem.xml>).

For more information about tuning JBoss for production environments, see [JBossASTuningSliming](http://wiki.jboss.org/wiki/Wiki.jsp?page=JBossASTuningSliming) (<http://wiki.jboss.org/wiki/Wiki.jsp?page=JBossASTuningSliming>).

2.6.6 Using Secure Sockets for User Application Connections to the Identity Vault

By default, secure sockets are used for communication between the User Application server and the Identity Vault. However, in some environments, not all communication needs to be secured. For example, if the User Application and Identity Vault servers are on an isolated network, and the only ports available to the outside are the HTTP ports, it might be acceptable for some communication between the two servers to be accomplished using non-secure sockets. Some aspects of the application will *always* use a secure connection (for example, a user changing a password) even though the setting might indicate that secure connections are not required. Turning off secure connections, especially for user connections, can greatly increase performance and scalability. If, in a particular environment, there are many concurrent logins, and communication between the User Application server and the Identity Vault server have been secured using the network setup, then

turning off the secure connection for user connections greatly increase the number of concurrent logins that can be processed. We recommend that this option be used only when there is actual evidence of scaling or performance problems in the environment, and adding additional eDirectory servers is not an option.

Additionally, secure connections can be turned off for administrative connections. These connections are used for general queries on the Identity Vault server that do not require user credentials. These connections are pooled and used round-robin. The bind over a secure connection is only done once at application startup (or possibly again later on if the connection becomes unresponsive) and so does not represent the scalability issues that can arise with the user connections. However, the time it takes to encrypt and decrypt the data at both ends does add overhead. We recommend that the default setting be used, unless there is a need to gain extra performance.

Secure communications for administrative and user connections must be disabled in both the User Application and in iManager. To disable secure communications for administrative and user connections, see the following topics:

- ♦ “Disabling Secure Communications Using the User Application Configuration Tool” on page 57
- ♦ “Disabling Secure Communications Using iManager” on page 57

Disabling Secure Communications Using the User Application Configuration Tool

To disable the secure administrative and user connections in the User Application:

- 1 Run the `configupdate` script, located in the User Application directory, as follows:
 - ♦ Linux: Type the following to run `configupdate.sh`:
`./configupdate.sh`
 - ♦ Windows: Run `configupdate.bat`

The User Application configuration utility starts.

- 2 Deselect *Secure Admin Connection* and *Secure User Connection*.



- 3 Click *OK*.

Disabling Secure Communications Using iManager

To disable the requirement for secure LDAP (LDAPS) connections for administrative and user connections to eDirectory using iManager or ConsoleOne:

- 1 Log into your eDirectory tree.
- 2 Navigate to the *LDAP* group object and display its properties.
- 3 Click *General*.
- 4 Deselect *Require TLS for Simple Binds with Password*.

NOTE: In a multi-server eDirectory tree, disabling TLS on the LDAP group removes the TLS requirement from all servers. If you want mixed TLS requirements for each individual server in your tree, you must enable the TLS requirement on each server.

2.7 Clustering

This section includes the following topics:

- ♦ [Section 2.7.1, “Clustering JBoss,” on page 58](#)
- ♦ [Section 2.7.2, “Things to Do Before Installing the User Application,” on page 59](#)
- ♦ [Section 2.7.3, “Installing the User Application to a JBoss Cluster,” on page 61](#)
- ♦ [Section 2.7.4, “Things to Do After Installing the User Application,” on page 66](#)

2.7.1 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. In addition, the Identity Manager User Application supports HTTP session replication and session failover. This means that if a session is in process on a node and that node fails, the session can be resumed on another server in the cluster without intervention.

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 [Load Balancing Web Applications \(http://www.onjava.com/pub/a/onjava/2001/09/26/load.html\)](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, include load balancing in your cluster configuration.

If you need additional information about JBoss setup, see the following sources:

- ♦ For Apache SSL setup, see the appropriate section on the [JBoss Web site \(http://www.jboss.org/wiki/Wiki.jsp?page=Tomcat\)](http://www.jboss.org/wiki/Wiki.jsp?page=Tomcat).
- ♦ For information on IIS SSL setup, see the [JBoss Forum for Installation, Configuration, and Deployment \(http://jboss.org/index.html?module=bb&op=viewtopic&p=3816794#3816794\)](http://jboss.org/index.html?module=bb&op=viewtopic&p=3816794#3816794).

JGroups Cluster Groups

JBoss clusters are based upon a communications module named JGroups. JGroups is installed with JBoss, but it can also be used without JBoss. JGroups provides communications among groups that 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.

User Application Cluster Group

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.

2.7.2 Things to Do Before Installing the User Application

This section provides information that you should be aware of before you install the User Application, and describes tasks that you should perform before installing the User Application.

This section includes the following topics:

- ♦ [“About Multiple Jboss Clusters on the Same Network” on page 59](#)
- ♦ [“Synchronizing JBoss Server Clocks” on page 60](#)
- ♦ [“Avoiding Multiple Logins from the Same Computer in a Cluster” on page 60](#)
- ♦ [“About the User Application Database” on page 60](#)

About Multiple Jboss Clusters on the Same Network

If you have more than one JBoss cluster running on a network, you must separate the clusters to prevent performance problems and anomalous behavior. You accomplish this by ensuring that each cluster uses a different partition name, multicast address, and multicast port. Even if you are not running multiple JBoss clusters on the same network, it's a good idea to specify a unique partition name for the cluster, rather than using the default partition. You can find instructions about running more than one cluster on a network by using your browser to view [Two Clusters Same Network](#)

(<http://wiki.jboss.org/wiki/Wiki.jsp?page=TwoClustersSameNetwork>). The following are important points:

- ♦ The cluster must have a unique cluster partition name and multicast address.

You can specify the cluster partition name and multicast address by editing the JBoss startup script (`start-jboss.bat` or `start-jboss.sh` for Windows or Linux, respectively) supplied with the User Application. You need to modify the JBoss startup scripts for your servers to start JBoss with a `-D` flag and set the `jboss.partition.name` and `jboss.partition.udpGroup` system properties (see “[Configuring the Workflow Engine](#)” on page 62).

- ♦ The cluster must use a unique multicast port.

You specify the port to use by editing the `mcast_port` attribute in the JBoss server `deploy\cluster-services.xml` file.

Synchronizing JBoss Server Clocks

You must synchronize the clocks of the servers in a User Application cluster. If server clocks are not synchronized, sessions might time out early, causing HTTP session failover to not work properly. There are many time synchronization methods available. The method that you use depends on the needs of your organization. One common approach is to use the Network Time Protocol (NTP). For a discussion of using the xNTP protocol for time synchronization, see [Time Synchronization using Extended Network Time Protocol \(xntp\)](http://www.novell.com/coolsolutions/trench/15650.html) (<http://www.novell.com/coolsolutions/trench/15650.html>).

Avoiding Multiple Logins from the Same Computer in a Cluster

We do not recommend using multiple logins across browser tabs or browser sessions on the same host. Some browsers share cookies across tabs and processes, so using multiple logins might cause problems with HTTP session failover (in addition to risking unexpected authentication functionality if multiple users share a computer).

About the User Application Database

When you install the User Application using the User Application installation program, you designate an existing version of a supported database to use (for example, MySQL, Oracle or Microsoft SQL Server). The database is used to store User Application data and User Application configuration information.

When the User Application is installed in a cluster environment, 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.

When you install the User Application into a JBoss cluster by using the User Application installation program, the data source is installed for you. The installation program creates a data source file named `IDM-ds.xml`, and places this file in the deploy directory (for example, `server/IDM/deploy`). The installation program also places the appropriate JDBC driver for the database specified during installation in the `lib` directory (for example, `/server/IDM/lib`). For more information about setting up the User Application database for a cluster, see “[Specifying the User Application Database](#)” on page 61.

NOTE: By default, MySQL sets the maximum number of connections to 100. This number might be too small to handle the workflow request load in a cluster. If the number is too small, you might see the following exception:

```
(java.sql.SQLException: Data source rejected establishment of
connection, message from server: "Too many connections.")
```

To increase the maximum number of connections, set the `max_connections` variable in `my.cnf` to a number greater than 100.

2.7.3 Installing the User Application to a JBoss Cluster

To install the User Application to a cluster you use the User Application installation program to install the User Application to each node in the cluster (see “Installing the User Application in the *Identity Manager 3.5 Installation Guide*). This section provides notes that are specific to installing the User Application to a cluster.

This section includes the following topics:

- “About the Server Configuration” on page 61
- “Specifying the User Application Database” on page 61
- “Selecting the Cluster (all) Option” on page 62
- “Configuring the Workflow Engine” on page 62
- “Using the Same Master Key for Each User Application in the Cluster” on page 63
- “Starting the User Application Cluster Group” on page 65

About the Server Configuration

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 (this is the default; the installation program allows you to change the name), and installs the User Application into the this configuration.

Specifying the User Application Database

All nodes in the JBoss cluster must access the same database instance. When you use the User Application installation program, you are prompted to specify the database name, host and port:

Figure 2-4 Specifying the Database Host and Port

The screenshot shows a window titled "Database Name & Privileged User". Inside, there is a text box with the instruction "Please provide the following:". Below this are four input fields: "Database name (or sid)" with the value "IDM35", "Database user" with the value "root", "Database user password" with masked characters "*****", and "Database user password (confirm)" also with masked characters "*****".

Make sure that you specify the same database parameters each time you install the User Application to a cluster node.

Selecting the Cluster (all) Option

When you use the User Application installation program, you are prompted to specify the IDM configuration:

Figure 2-5 Specifying the Cluster (all) Option and Engine ID

The screenshot shows a window titled "IDM Configuration". It contains a text box with instructions: "Choose 'default' for a single instance, or 'all' if you plan to employ clustering. We will copy one of these servers to 'Server name' and customize it to your needs. The 'Workflow Engine ID' is only valid for cluster installs." Below this, there is a section titled "Single node (default) or cluster (all)?" with two radio buttons: "default" and "all". The "all" option is selected. Below the radio buttons are two input fields: "Server name" with the value "IDM" and "Workflow Engine ID" with the value "Engine1".

Make use that you select the *clustering (all)* option.

Configuring the Workflow Engine

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.

When you install the User Application to the cluster using the User Application installation program (see [“Installing the User Application to a JBoss Cluster” on page 61](#)), you accomplish this by specifying the IP address or host name of the server on which the database for the User Application is installed.

- ◆ Each server in the cluster needs to be started with a unique engine-id.

You can accomplish this 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)
```

You might want to combine the setting of this system property with the setting of other system properties (see [“Setting JBoss system properties in the JBoss startup script” on page 63](#)).

For information about managing running workflows, see [Section , “Managing Workflows in a Cluster,” on page 68](#).

Setting JBoss system properties in the JBoss startup script

Each server in the cluster should be started using the same partition name and partition UDP group (see [“About Multiple Jboss Clusters on the Same Network” on page 59](#)). Each server in the cluster should use a unique engine ID (see [“Configuring the Workflow Engine” on page 62](#)).

You can modify your JBoss startup script (`start-jboss.bat` for Windows, `start-jboss.sh` for Linux) to specify all of these system properties. This script is located in the directory in which your User Application files are stored. For example, to start a server using the partition name “Example_Partition”, the UDP group “228.3.2.1” and the Engine ID “Engine1” you would add the following to the `start-jboss` script:

```
start run.bat -c IDM -Djboss.partition.name=Example_Partition -
Djboss.partition.udpGroup=228.3.2.1 -Dcom.novell.afw.wf.engine-
id=Engine1
```

Using the Same Master Key for Each User Application in the Cluster

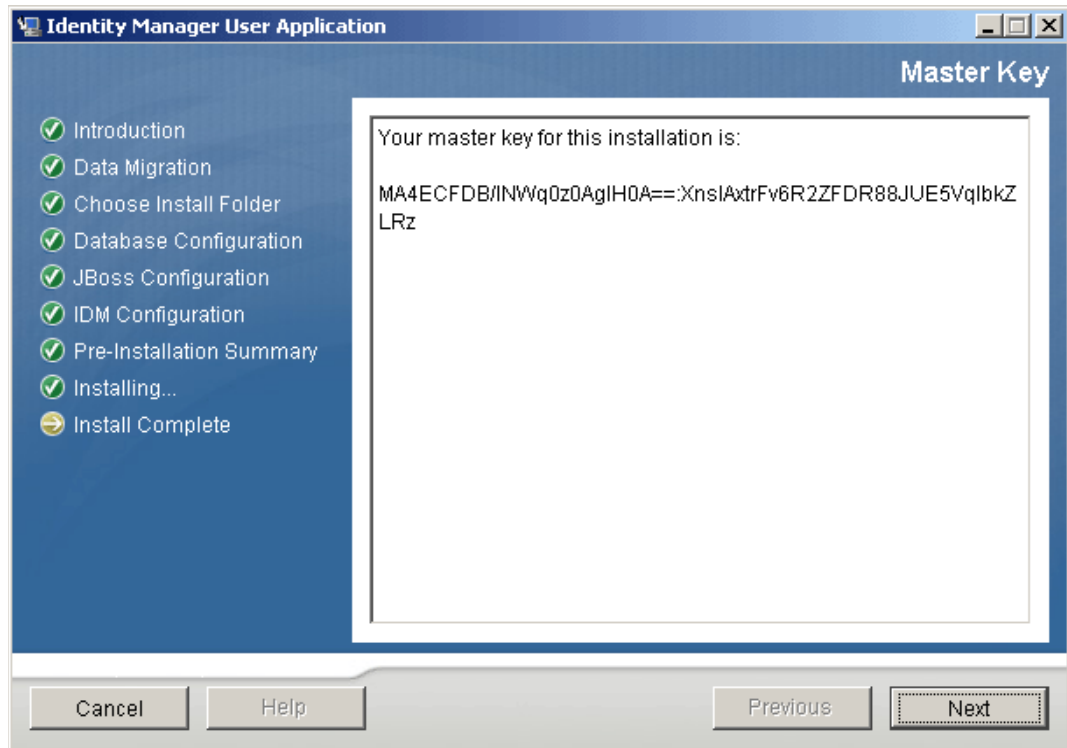
The Identity Manager User Application encrypts sensitive data (see [Section 2.2.6, “Encryption of Sensitive User Application Data,” on page 43](#)). A master key is used to access encrypted data. All User Applications in a cluster must use the same master key. Follow these steps to ensure that all User Applications in a cluster use the same master key.

- 1 Using the User Application installation program, install the User Application to the first node in the cluster.

For information about using the User Application installation program, see “Installing the User Application in the *Identity Manager 3.5 Installation Guide*.”

When you use the User Application installation program to install the first User Application in a cluster, at the end of the installation you are presented with a new master key for the User Application:

Figure 2-6 Master Key

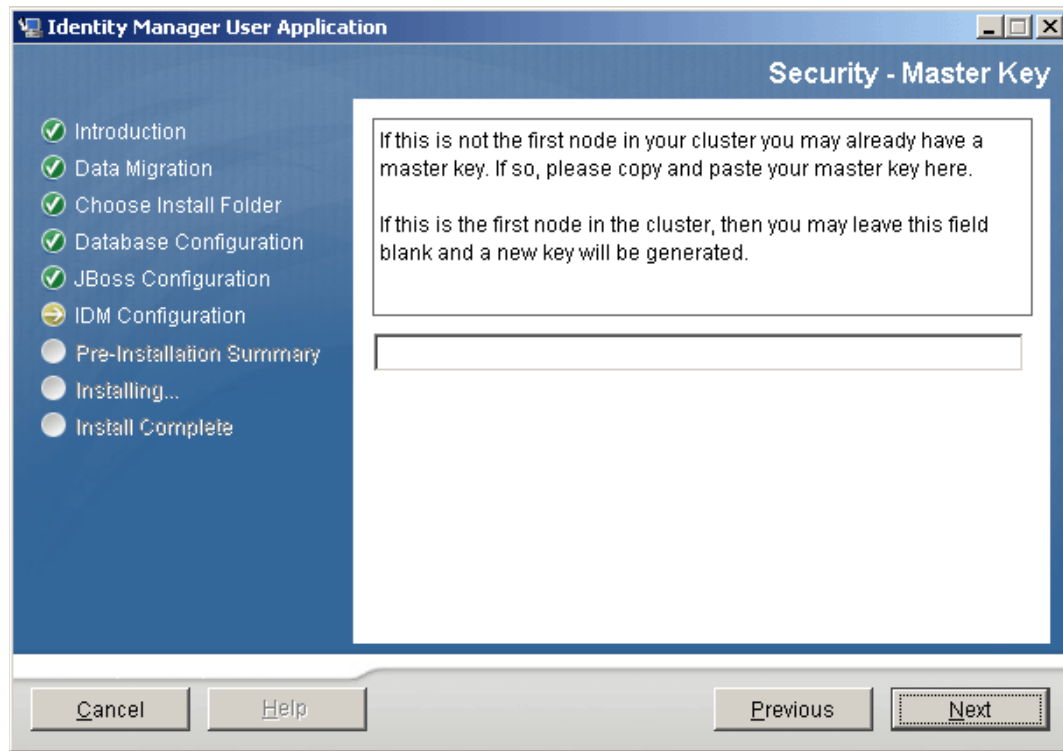


Follow the on-screen instructions to save the master key to a text file.

- 2 Using the User Application installation program, install the User Application to the other nodes in the cluster.

When you install the User Application to the other nodes in the cluster, the installation program provides a page that you use to import the master key:

Figure 2-7 Pasting Master Key in User Application Installation Program

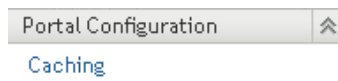


- 3 Import the master key that you saved to a text file in [Step 1 on page 63](#).

Starting the User Application Cluster Group

After the User Applications in your cluster have been installed, you must enable the cluster in the User Application cluster configuration.

- 1 Start the first User Application in the cluster.
- 2 Log in as the User Application administrator.
Don't start any other servers yet.
- 3 Click *Administration*.
The User Application displays the Application Configuration portal.
- 4 Click *Caching*.



The *Caching Management* page is displayed.

- 5 Select *True* for the *Cluster Enabled* property.
- 6 Click *Save*.
- 7 Restart the server.

- 8 If you are using local settings (see “[Specifying the User Application Cluster Group Caching Configuration](#)” on page 67), repeat this procedure for each server in the cluster.

2.7.4 Things to Do After Installing the User Application

This section describes User Application cluster configuration actions that you perform after installing the User Application.

This section includes the following topics:

- “[Configuring the User Application Driver for Clustering](#)” on page 66
- “[Specifying the User Application Cluster Group Caching Configuration](#)” on page 67
- “[Configuring Logging in a Cluster](#)” on page 67
- “[Managing Workflows in a Cluster](#)” on page 68

Configuring the User Application Driver for Clustering

Clustering is the only scenario in which the same User Application driver is used by multiple User Applications. The User Application driver stores various kinds of information (such as workflow configuration and cluster information) that is application-specific. Therefore, a single instance of the User Application driver should be not shared among multiple applications.

The User Application stores application-specific data to control and configure the application environment. This includes JBoss application server cluster information and the workflow engine configuration. The only User Applications that should share a single User Application driver instance are those applications that are part of the same JBoss cluster.

In a cluster, the User Application driver must be configured to use the host name or IP address of the dispatcher or load balancer for the cluster. You create the User Application driver when you install the User Application (see the *Novell Identity Manager Installation Guide*). You configure the User Application driver using iManager.

- 1 Log into the instance of iManager that manages your Identity Vault.
- 2 Click the *Identity Manager* node in the iManager navigation frame.
- 3 Click *Identity Manager Overview*.
- 4 Use the search page to display the Identity Manager Overview for the driver set that contains your User Application driver.
- 5 Click the round status indicator in the upper right corner of the driver icon:



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

- 6 Click *Edit Properties*.
- 7 In the *Driver Parameters* section, change the *Host* parameter to the host name or IP address of the dispatcher.

8 Click *OK*.

Specifying 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 “[Cache Settings for Clusters](#)” on page 97). Changes to the cluster configuration only take effect for a server node when the server node is restarted.

In most cases you should use global settings when configuring a cluster. However, global settings present a problem if you need to use TCP, because the IP address of the server must be specified in the JGroups initialization string for each server. You can use local settings to specify a JGroups initialization string by checking *Enable Local for Cluster Properties*, then typing the JGroups initialization string in the *Local* field. For an example of a working JGroups TCP protocol stack, see [JGroupsStackTCP](http://wiki.jboss.org/wiki/Wiki.jsp?page=JGroupsStackTCP) (<http://wiki.jboss.org/wiki/Wiki.jsp?page=JGroupsStackTCP>).

WARNING: If you specify local settings and enter an incorrect configuration in the JGroups initialization string, the cache cluster function might not start. Unless you know how to configure JGroups correctly and understand the protocol stack, you should not use local settings.

Alternatively, you can add a token (for example, “IDM_HOST_ADDR”) to the global settings for the *Cluster Properties*. You can then edit the *hosts* file on each server in the cluster to specify the IP address for that server.

Configuring Logging in a Cluster

This section includes the following topics:

- ♦ “[JBoss Logging](#)” on page 67
- ♦ “[User Application Logging](#)” on page 68

JBoss Logging

You can configure JBoss for logging in a cluster. 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`).

User Application Logging

The User Application logging configuration (see [Section 5.1.4, “Logging Configuration,” on page 102](#)) is not propagated to all servers in cluster. For example, if you use the Logging administration page on a server in a cluster to set the logging level for `com.novell.afw.portal.aggregation` to Trace, this setting is not propagated to the other servers in the cluster. You must individually configure the level of logging messages for each server in the cluster.

Managing Workflows in a Cluster

The Identity Manager User Application workflow cluster implementation binds process instances to the engine on which they started. This is done by associating a workflow process instance with an engine-id and is maintained in the cluster database. When a workflow engine is started, it resumes process instances that are assigned to its engine-id. This prevents multiple engines in a cluster from resuming the same process instance. If a workflow engine fails, processes that were running on that engine are automatically resumed on another engine in the cluster.

You can manually reassign processes to other engines in the cluster. For example, an administrator could reassign processes back to a failed workflow engine when the workflow engine is brought back online, or redistribute processes to other engines when an engine is permanently removed from the cluster (see [Section 18.2.7, “Managing Workflow Processes in a Cluster,” on page 331](#)).

When the workflow engine starts up it checks to see if its engine ID is already in use by another node in the cluster. When this is the case, the workflow engine checks the cluster database to see if the status of the engine is SHUTDOWN or TIMEDOUT. If it is, the workflow engine starts. If the status is STARTING or RUNNING, the workflow engine logs a warning, then waits for a heartbeat timeout to occur. If the heartbeat timeout occurs, that means that the other workflow engine with the same ID was not shut down properly, so it's safe to start. If the heartbeat timer is updated, that means another workflow engine with the same ID is running in the cluster, so the workflow engine cannot start. You can specify the heartbeat timeout (the maximum elapsed time between heartbeats before a workflow engine is considered timed out) by setting the *Heartbeat Interval* and *Heartbeat Factor* properties in the User Application (see [Section 8.4.2, “Configuring the Workflow Cluster,” on page 198](#)).

2.8 Localizing Text

Identity Manager provides a number of tools for localizing the User Application. This section provides a convenient reference for finding the information that you need to localize in the User Application.

Table 2-4 *Localization Topics*

Localization topic:	Where to find it:
Set preferred locale for User Application	See “Preferred Locale” and “Choosing a Preferred Language” in the <i>Identity Manager User Application: User Guide</i> .
E-mail templates	See Section 18.4.4, “Adding Localized E-Mail Templates,” on page 346.
Challenge questions	See “Security: Best Practices” in the <i>Novell Identity Manager 3.5 Administration Guide</i> .
Password sync status application name	See Table 5-14, “Password Sync Status Application Settings,” on page 134.
Names of container pages	See the <i>Page Name</i> property in Section 6.2.1, “Creating Container Pages,” on page 148.
Names of shared pages	See Section 6.3.1, “Creating Shared Pages,” on page 156.
Portlet preferences	See Section 7.3.5, “Modifying Preferences for Portlet Registrations,” on page 183.
Provisioning request definitions created in iManager	See Section 17.3.2, “Creating or Editing a Provisioning Request,” on page 300.
Provisioning team definitions	Section 19.2.2, “Creating or Editing a Provisioning Team,” on page 351.
General information about localizing display labels in directory abstraction layer objects and provisioning request definitions in Designer	See “Localizing Display Labels” in the <i>Identity Manager 3.5 User Application: Design Guide</i> .
Entity display labels	See “Adding Entities” in the <i>Identity Manager 3.5 User Application: Design Guide</i> .
Display labels for global lists	“Working with Lists” in the <i>Identity Manager 3.5 User Application: Design Guide</i> .
Display labels for relationship properties	See “Relationship Properties” in the <i>Identity Manager 3.5 User Application: Design Guide</i> .
Digital signature declaration strings	See “Creating a Signature Declaration” in the <i>Identity Manager 3.5 User Application: Design Guide</i> .
Workflow activity display names	See “Workflow Activity Reference” in the <i>Identity Manager 3.5 User Application: Design Guide</i> .

Setting Up Logging

3

This section includes the following:

- ♦ [Section 3.1, “About Event Logging,” on page 71](#)
- ♦ [Section 3.2, “Logging to a Novell Audit or Sentinel Server,” on page 72](#)

3.1 About Event Logging

The Identity Manager User Application implements logging by using log4j, an open-source logging package distributed by The Apache Software Foundation. See [Logging Services \(http://logging.apache.org/log4j\)](http://logging.apache.org/log4j) for details. 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, review the [Novell Audit documentation \(http://www.novell.com/documentation/novellaudit20/index.html\)](http://www.novell.com/documentation/novellaudit20/index.html).

3.1.1 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>
  <priority value="INFO"/>
  <appender-ref ref="CONSOLE"/>
  <appender-ref ref="FILE"/>
</root>
```

Change the priority value to:

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

3.1.2 Changing the User Application Log Level Settings

The User Application enables you to change the log level settings of individual loggers.

- 1 Log in to the User Application as the User Application Administrator.
- 2 Select the *Administration* tab.

- 3 Select the *Logging* link.
- 4 Change the *Log Level* of any logger.
- 5 To save the changes for application server restarts, select *Persist the logging changes*.
- 6 Click *Submit*.

The User Application logging configuration is saved in `<installdir>/jboss/server/IDMProv/conf/idmuserapp_logging.xml`.

3.2 Logging to a Novell Audit or Sentinel Server

To log to a Novell Audit or Sentinel server:

Step	What to do	For more information
1	Add the Identity Manager application schema to the Novell Audit server as a log application	Section 3.2.1, “Adding the Identity Manager Application Schema to your Novell Audit Server as a Log Application,” on page 73

Step	What to do	For more information
2	Configure the Novell Audit platform agent on your application server	<p>The Platform Agent is required on any client that reports events to Novell Audit or Sentinel. You configure the platform agent through the <code>logevent</code> 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:</p> <ul style="list-style-type: none"> ♦ Linux: <code>/etc/logevent.conf</code> ♦ Windows: <code>/<WindowsDir>/logevent.cfg</code> (Usually <code>c:\windows</code>) <p>Specify the following four properties:</p> <p>Loghost: The IP address or DNS name of your Novell Audit or Sentinel server. For example: <code>LogHost=xxx.xxx.xxx.xxx</code></p> <p>LogJavaClassPath: The location of the <code>lcache.jar</code> file <code>NauditPA.jar</code>. For example: <code>LogJavaClassPath=/opt/novell/idm/NAuditPA.jar</code></p> <p>LogCacheDir: Specifies where <code>lcache</code> stores cache files. For example: <code>LogCacheDir=/opt/novell/idm/naudit/cache</code></p> <p>LogCachePort: Specifies on which port <code>lcache</code> listens for connections. The default is 288, but in a Linux server, set the port number greater than 1000. For example: <code>LogCachePort=1233</code></p> <p>Specify any other settings needed for your environment.</p> <hr/> <p>IMPORTANT: You must restart the Platform Agent any time you change the configuration.</p> <hr/> <p>For more information about the structure of the <code>logevent</code> configuration file, see the section on configuring platform agents (http://www.novell.com/documentation/novellaudit20/index.html) in the section on the logging system in the <i>Novell Audit Administration Guide</i>.</p>
3	Enable Novell Audit logging	Section 3.2.2, "Enabling Audit Logging," on page 74

3.2.1 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:

- 1 Locate the following file:

dirxml.lsc

This file is located in the Identity Manager User Application installation directory after the install, for example `/opt/novell/idm`.

- 2 Use a Web browser to access an iManager with the NovellAudit plug-in installed, 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, select the appropriate Container Name, and click the *New Log Application* link.
- 6 When the New Log Application dialog box displays, specify the following:

For this setting	Do this
<i>Log Application Name</i>	Type any name that is meaningful for your environment
<i>Import LSC File</i>	Use the <i>Browse</i> button to select the <code>dirxml.lsc</code> file

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.

3.2.2 Enabling Audit Logging

To enable Novell Audit logging in your Identity Manager User Application:

- 1 Log in to the User Application as the User Application Administrator.
- 2 Select the *Administration* tab.
- 3 Select the *Logging* link.
- 4 Select the *Also send logging messages to NovellAudit* check box (near the bottom of the page).
- 5 To save the changes for any subsequent application server restarts, make sure *Persist the logging changes* is selected.
- 6 Click *Submit*.

3.2.3 Events That Are 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:

Table 3-1 *Logged Events*

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
31470	Digital Signature	Digital_Signature_Verification_Request	Info
31471		Digital_Signature_Verification_Failure	Error
31472		Digital_Signature_Verification_Success	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
31534		Workflow_Escalated	Info
31535		Workflow_Reminder_Sent	Info
31536		Digital_Signature	Info
31537		Workflow_ResetPriority	Info
3152A		Workflow_Completed	Info
3152B		Workflow_Timedout	Info
3152C		User_Message	Info
31533		Workflow_Retracted	Info

Event ID	Process	Event	Severity
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
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

3.2.4 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 77](#).
- ♦ Write queries against the logged data by using iManager to select *Auditing and Logging > Queries*.
- ♦ Write your own SQL queries against the logged data.
- ♦ Produce Identity Manager reports in Sentinel (see [“Sentinel Reports” on page 79](#)).

The default Novell Audit table is called NAUDITLOG.

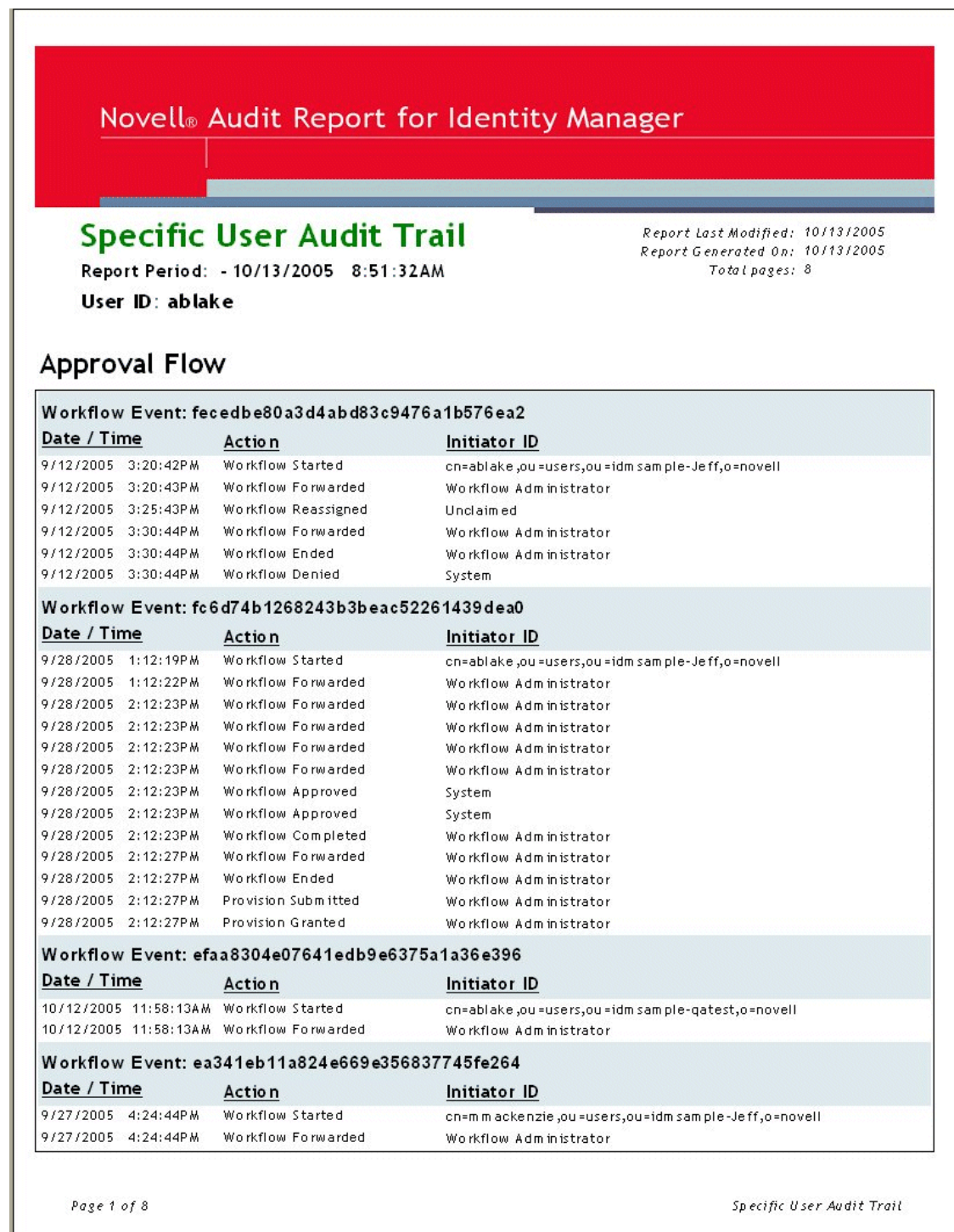
Predefined Log Reports

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

The following graphic shows an example of the Specific User Audit Trail report:

Figure 3-1 Sample Audit Trail Report



The report files are in the following locations:

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/novellaudit20) (<http://www.novell.com/documentation/novellaudit20>).

Sentinel Reports

If you have configured the platform agent to send events to Sentinel, you can produce the following reports about Identity Manager events in Sentinel:

- ♦ `IDM_Administrative_Action_Report.rpt`
- ♦ `IDM_Historical_Approval_Flow_Report.rpt`
- ♦ `IDM_Password-Management.rpt`
- ♦ `IDM_Provisioning_Report_by_Top_10_DHNs.rpt`
- ♦ `IDM_Provisioning_Report_by_Top_10_DIPs.rpt`
- ♦ `IDM_Resource_Provisioning_Report.rpt`
- ♦ `IDM_Specific_User_Audit_Trail_Report.rpt`
- ♦ `IDM_Specific_User_Provisioning_Report.rpt`
- ♦ `IDM_Sync-vs-Reset.rpt`
- ♦ `IDM_User_Provisioning_Report.rpt`
- ♦ `IDM_Workflow_Stats_by_Top_10_DHNs.rpt`
- ♦ `IDM_Workflow_Stats_by_Top_10_DIPs.rpt`

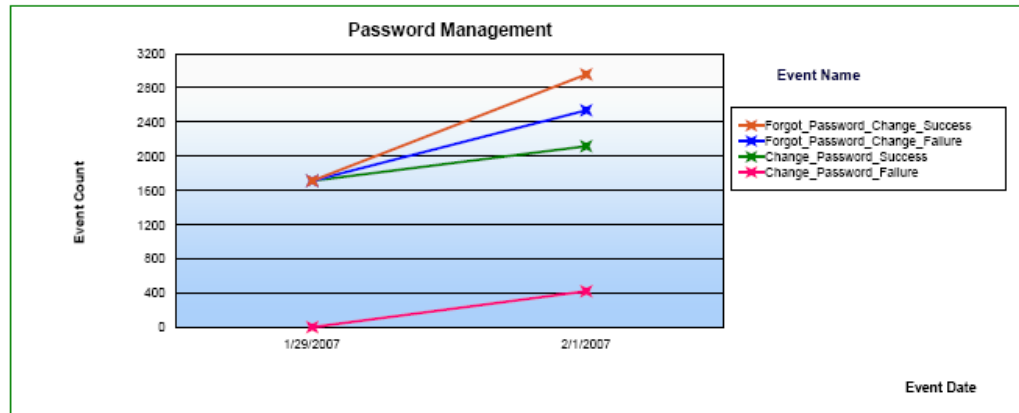
For more information about Sentinel reports, see the *Sentinel User's Guide*. The following is a sample Sentinel report on Password Management:

Figure 3-2 Sample Sentinel Report

Password Management: 01/01/2005 - 03/01/2007

Report Description : This report shows password related events count trend monitored by Sentinel Collectors. The graph below shows Daily event trend based on the total event count for the selected Date Range.

Report Period: 01-01-2005 12:00:00 AM - 03-01-2007 12:00:00 AM

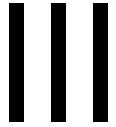


Below Cross Chart Summary indicates the total number of events related to Password Management by date-wise.

Password Management Event Count :

Event Date	Change_Password_Failure	Change_Password_Success	Forgot_Password_Change_Failure	Forgot_Password_Change_Success	Total
1/29/2007	1	1712	0	0	1713
2/1/2007	421	1698	420	420	2959

Administering the User Application



These sections describe how to configure and manage the Identity Manager User Application by using the *Administration* tab of the user interface.

- ♦ [Chapter 4, “Using the Administration Tab,” on page 83](#)
- ♦ [Chapter 5, “Application Configuration,” on page 89](#)
- ♦ [Chapter 6, “Page Administration,” on page 139](#)
- ♦ [Chapter 7, “Portlet Administration,” on page 173](#)
- ♦ [Chapter 8, “Provisioning Configuration,” on page 189](#)
- ♦ [Chapter 9, “Security Configuration,” on page 201](#)

Using the Administration Tab

4

This section 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:

- ♦ [Section 4.1, “About the Administration Tab,” on page 83](#)
- ♦ [Section 4.2, “Who Can Use the Administration Tab,” on page 83](#)
- ♦ [Section 4.3, “Accessing the Administration Tab,” on page 84](#)
- ♦ [Section 4.4, “Administration Actions You Can Perform,” on page 86](#)

4.1 About the Administration Tab

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

For example, choose the *Administration* tab 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

4.2 Who Can Use the Administration Tab

The *Administration* tab is not visible to typical end users of the Identity Manager user interface. There are three 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 9, “Security Configuration,” on page 201](#).

Provisioning Application Administrators: A Provisioning Application Administrator is authorized to perform provisioning-related tasks for the Identity Manager User Application. During installation, a user is specified as Provisioning Application administrator. For details, see [Chapter 8, “Provisioning Configuration,” on page 189](#). The User Application administrator can use the Security page on the *Administration* tab to specify other Provisioning Application administrators. This includes performing the tasks on the Provisioning page of the *Administration* tab. For details, see [Chapter 9, “Security Configuration,” on page 201](#).

Users permitted by User Application Administrator: 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 6, “Page Administration,” on page 139](#)

4.3 Accessing the Administration Tab

When you are a User Application Administrator (or other permitted user), you can access the *Administration* tab of the Identity Manager user interface 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*.

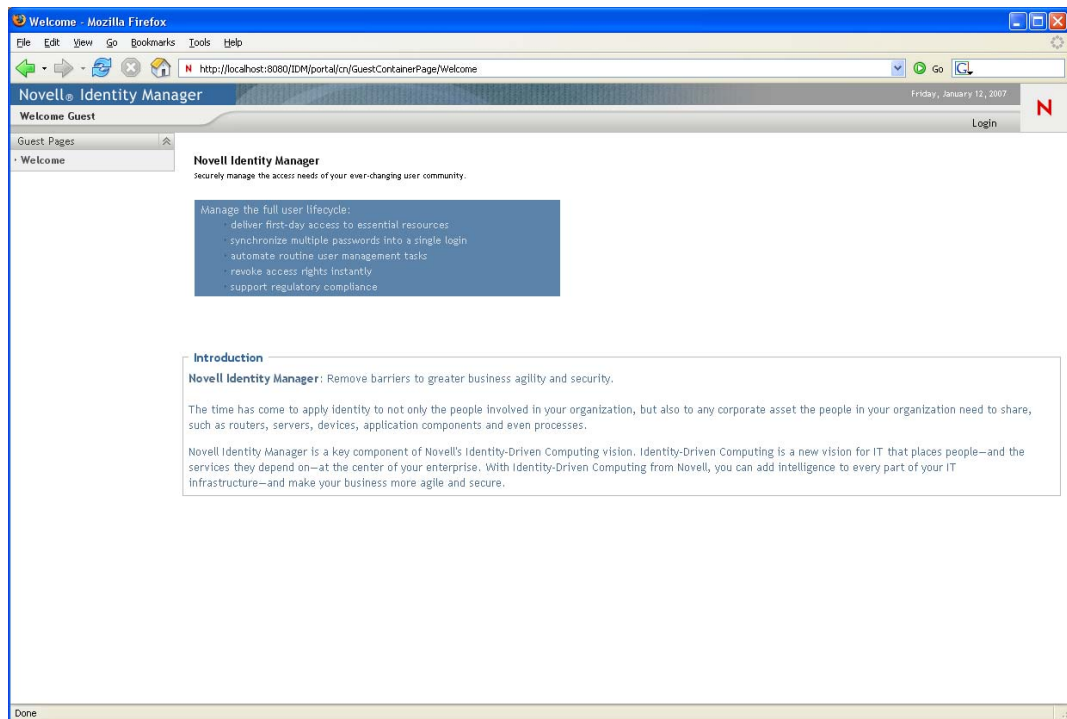
NOTE: To use the Identity Manager user interface, make sure your Web browser has JavaScript* and cookies enabled.

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 Welcome Guest page of the user interface displays:



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

The user interface prompts you for a username and password:

The image shows the Novell Identity Manager login interface. It features a blue header with the text "Novell® Identity Manager". Below the header, there are two input fields labeled "Username:" and "Password:". Under the "Username:" field, there is a link that says "→ Forgot Password?". At the bottom right, there is a "Login..." button. A small red "N" logo is visible in the bottom left corner of the interface.

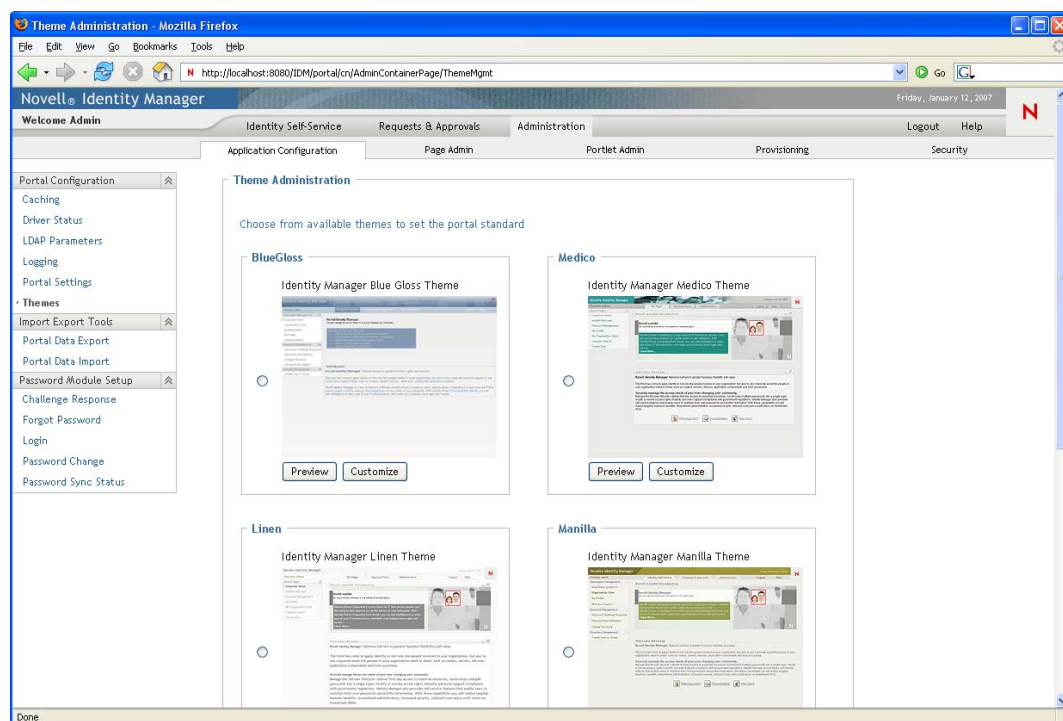
- 3** Specify the username and password of a User Application Administrator (or a user with some *Administration* tab permissions), then click *Login*.

After you log in, you see the appropriate user-interface content for that user.

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 Application Configuration page:



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

4.4 Administration Actions You Can Perform

After you're on the *Administration* page, you can use any available actions to configure and manage the Identity Manager User Application. [Table 4-1](#) contains a summary.

Table 4-1 *Administration Actions Summary*

Action	Description
Application Configuration	Controls User Application configuration of caching, logging, password management, and LDAP connection parameters. Provides read-only information about the driver status and the portal. Provides access to tools that allow you to export or import portal content (pages and portlets used in the Identity Manager User Application). For details, see Chapter 5, "Application Configuration," on page 89.
Page Admin	Controls the pages displayed in the Identity Manager user interface and who has permission to access them For details, see Chapter 6, "Page Administration," on page 139.

Action	Description
Portlet Admin	<p>Controls the portlets available in the Identity Manager user interface and who has permission to access them</p> <p>For details, see Chapter 7, “Portlet Administration,” on page 173.</p>
Provisioning	<p>Controls the configuration of delegation and proxy tasks, digital signature service and engine and cluster settings.</p> <p>For details, see Chapter 8, “Provisioning Configuration,” on page 189.</p>
Security	<p>Specifies who is a User Application Administrator and Provisioning Administrator for the Identity Manager User Application</p> <p>For details, see Chapter 9, “Security Configuration,” on page 201.</p>

Application Configuration

5

This section describes the tasks that you can perform from the Application Configuration page. It includes the following sections:

- ♦ [Section 5.1, “Portal Configuration Tasks,” on page 89](#)
- ♦ [Section 5.2, “Working with the Import and Export Tools,” on page 113](#)
- ♦ [Section 5.3, “Password Management Configuration,” on page 120](#)

5.1 Portal Configuration Tasks

This section includes information about:

- ♦ [Section 5.1.1, “Caching Management,” on page 89](#)
- ♦ [Section 5.1.2, “Driver Status,” on page 99](#)
- ♦ [Section 5.1.3, “LDAP Parameters,” on page 100](#)
- ♦ [Section 5.1.4, “Logging Configuration,” on page 102](#)
- ♦ [Section 5.1.5, “Portal Settings,” on page 107](#)
- ♦ [Section 5.1.6, “Theme Administration,” on page 107](#)

5.1.1 Caching Management

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.

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. However, if you have a specific need, you can manually flush selected caches or all caches.

1 Go to the Caching page:

Caching Management

Flush Cache

Choose a cache from the list and click on Flush Cache button to flush the cache.

Flush all

Flush Cache

Cluster Configuration

All changes to the current cluster configuration will take effect the next time application starts up. Group ID must be a unique name which does not match JBoss reserved cluster name DefaultPartition or TreeCache. Only an experienced administrator should modify the default cluster properties. Please refer to JBoss documentation before making any changes.

	Current	Global	Enable Local	Local
Cluster Enabled:	False	False	<input type="checkbox"/>	
Group ID:	c373e901ab5e8ee996444553544200	c373e901ab5e8ee996444553544200	<input type="checkbox"/>	
Cluster Properties:	view	UOP{mcast_addr=228.8.8.8;mcast_p	<input type="checkbox"/>	

Cache Configuration

All changes to the current cache configuration will take effect the next time application starts up. (* Indicates required)

Settings that apply to entire cache system

	Current	Global	Enable Local	Local
Lock Acquisition Timeout:*	15000	15000	<input type="checkbox"/>	
Wake Up Interval Seconds:*	5	5	<input type="checkbox"/>	
Eviction Policy Class:*	org.jboss.cache.eviction.LRUPolicy	org.jboss.cache.eviction.LRUPolicy	<input type="checkbox"/>	

Settings that apply to all non-customizable Cache Holders

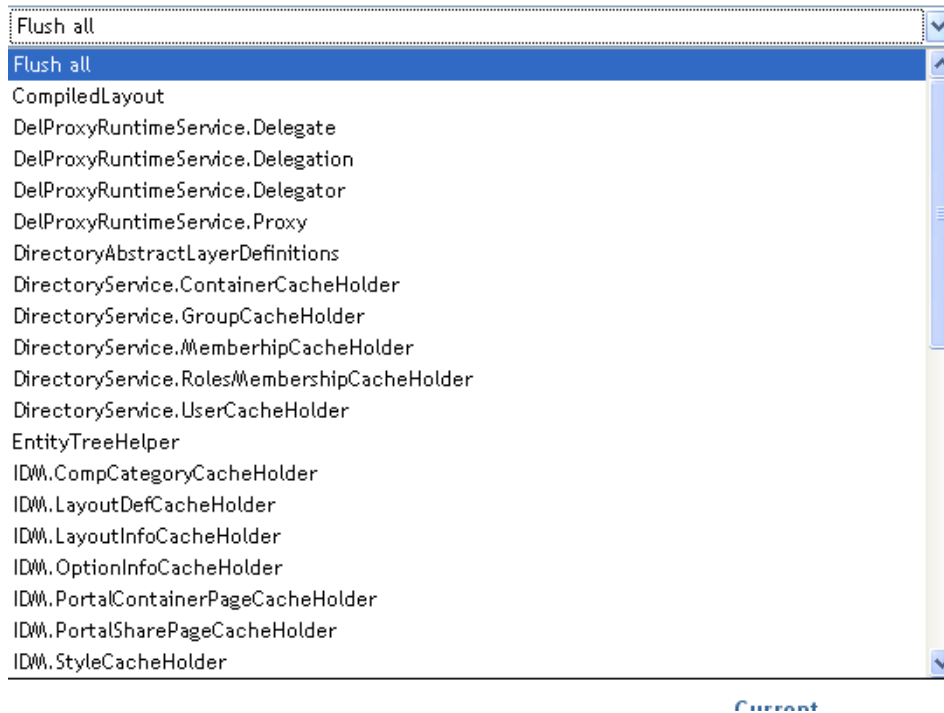
	Current	Global	Enable Local	Local
Max Nodes:*	10000	10000	<input type="checkbox"/>	
Time To Live Seconds:*	0	0	<input type="checkbox"/>	

Settings that apply to customizable Cache Holders. Click the 'Customize Cache Holders' button to change the settings for customizable cache holders.

Customize Cache Holders

Save

- 2 In the *Flush Cache* section of the page, use the drop-down list to select a particular cache to flush (or select *Flush all*):



The list of available caches is dynamic; it changes depending on what data is cached at the moment.

- 3 Click *Flush Cache*.

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 in [“Flushing caches” on page 89](#) to force the latest definitions to be loaded from the Identity Vault.

For more information on the User Application's directory abstraction layer, see the *Identity Manager User Application: Design Guide*.

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 is automatically flushed on every server in the cluster.

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 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).

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/jboss-cache (<http://www.jboss.org/products/jboss-cache>).

How Cache Settings Are Stored

Two levels of settings are available for controlling cache configuration: global, and local. Use these settings to customize the caching behavior of the Identity Manager User Application. **Table 5-1 on page 92** describes the cache configuration settings.

Table 5-1 *Cache Configuration Settings*

Level	Description
Global settings	<p>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.</p> <p>To find the global settings in your Identity Vault, look for the following object under your Identity Manager User Application driver:</p> <pre>configuration.AppDefs.AppConfig</pre> <p>For example:</p> <pre>configuration.AppDefs.AppConfig.MyUserApplicationDriver.MyDriverSet.MyOrg</pre> <p>The XmlData attribute of the configuration object contains the global settings data.</p>

Level	Description
Local setting	<p>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 to reassign a server to a different cluster.</p> <p>To find the local settings on your application server, look for the following file under your JBoss server configuration's <code>conf</code> directory: <code>sys-configuration-xmldata.xml</code>, for example <code>jboss/server/IDM/conf/sys-configuration-xmldata.xml</code>.</p> <p>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.)</p>

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.

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).

Cluster and Cache Configuration

Cluster Configuration

All changes to the current cluster configuration will take effect the next time application starts up. Group ID must be a unique name which does not match JBoss reserved cluster name DefaultPartition or TreeCache. Only an experienced administrator should modify the default cluster properties. Please refer to JBoss documentation before making any changes.

	Current	Global	Enable Local	Local
Cluster Enabled:	False	False	<input type="checkbox"/>	
Group ID:	c373e901aba5e8ee9966444553544200	c373e901aba5e8ee9966444553544200	<input type="checkbox"/>	
Cluster Properties:	view	UDP(mcast_addr=228.8.8.8;mcast_p	<input type="checkbox"/>	

Cache Configuration

All changes to the current cache configuration will take effect the next time application starts up. (* indicates required)

Settings that apply to entire cache system

	Current	Global	Enable Local	Local
Lock Acquisition Timeout:*	15000	15000	<input type="checkbox"/>	
Wake Up Interval Seconds:*	5	5	<input type="checkbox"/>	
Eviction Policy Class:*	org.jboss.cache.eviction.LRUPolicy	org.jboss.cache.eviction.LRUPolic	<input type="checkbox"/>	

Settings that apply to all non-customizable Cache Holders

	Current	Global	Enable Local	Local
Max Nodes:*	10000	10000	<input type="checkbox"/>	
Time To Live Seconds:*	0	0	<input type="checkbox"/>	

Settings that apply to customizable Cache Holders. Click the 'Customize Cache Holders' button to change the settings for customizable cache holders.

Customize Cache Holders

The global settings always have values. The local settings are optional.

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
<i>Lock Acquisition Timeout</i>	Specify the time interval (in milliseconds) that the cache waits for a lock to be acquired on an object. You might 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.
<i>Wake Up Interval Seconds</i>	Specify the time interval (in seconds) that the cache eviction policy waits before waking up to do the following: <ul style="list-style-type: none"> ♦ Process the evicted node events ♦ Clean up the size limit and age-out nodes
<i>Eviction Policy Class</i>	Specify the classname for the cache eviction policy that you want to use. The default is the LRU eviction policy that JBoss Cache provides: <p><code>org.jboss.cache.eviction.LRUPolicy</code></p> <p>If appropriate, you can change this to another eviction policy that JBoss Cache supports.</p> <p>To learn about supported eviction policies, go to www.jboss.org/products/jbosscache (http://www.jboss.org/products/jbosscache).</p>
<i>Max Nodes</i>	Specify the maximum number of nodes allowed in the cache. For no limit, specify: <p>0</p> <p>You can customize this setting for some cache holders. See “Customizable Cache Holders” on page 95.</p>
<i>Time To Live Seconds</i>	Specify the time to idle (in seconds) before the node is swept away. For no limit, specify: <p>0</p> <p>You can customize this setting for some cache holders. See “Customizable Cache Holders” on page 95.</p>
<i>Max Age</i>	Specifies the number of seconds an entry should be allowed to stay in the cache holder since its creation time. For no time limit, specify: <p>0</p> <p>This setting is only available for “Customizable Cache Holders” on page 95.</p>

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 deselected, 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 servers.

Customizable Cache Holders

You can customize the *Max Nodes*, *Time To Live*, and *Max Age* settings for some cache holders. The cache holders are listed in [Table 5-2](#).

Table 5-2 Customizable Cache Holders

Cache Holder Name	Description
DirectoryAbstractionLayerDefinitions	Caches the Directory Abstraction Layer definitions to optimize performance for all data model operations. See "Flushing the Directory Abstraction Layer Cache" on page 91.
DirectoryService.ContainerCacheHolder	Caches containers in the directory layer. Containers are shared by many users and groups, and reading them from the directory layer involves both network communication (with the LDAP server) and object creation. By default, the cache is limited to 50 containers, and the LRUs have a default Time To Live (TTL) of 10 minutes. Depending on the directory topography in your enterprise, you might need to adjust the maximum number of nodes or the TTL if you find the performance is suffering because of queries to the LDAP server for container objects. Making settings too high in combination with a large number of usable containers can cause unneeded memory consumption and net lower performance from the server.
DirectoryService.DelProxyRuntimeServiceDelegate	Caches delegate assignments.
DirectoryService.DelProxyRuntimeService.Delegation	Caches user availability settings.
DirectoryService.DelProxyRuntimeService.Delegator	Caches the delegator entities.
DirectoryService.DelProxyRuntimeService.Proxy	Caches proxy assignments.

Cache Holder Name	Description
DirectoryService.GroupCacheHolder	Caches groups in the directory layer. Groups are often shared by many users, and reading them from the directory layer involves both network communication (with LDAP server) and object creation. By default, the cache is limited to 500 groups, and the LRUs have a default TTL of 10 minutes. Depending on the user/group topography in your enterprise, you might need to adjust the maximum number of nodes or the TTL if you find the performance is suffering because of queries to the LDAP server for groups objects. Settings that are too high, in combination with a large number of usable groups, can cause unneeded memory consumption, and net lower performance from the server.
DirectoryService.MemberhipCacheHolder	Caches the relationship between a user and a set of groups. Querying the set of groups a user belongs to can be a network and CPU intensive operation on the LDAP server, especially if dynamic groups are enabled. For this reason, relationships are cached with an expiration interval so that changes in the criteria for inclusion/exclusion in a group (such as time-based dynamic groups) are reflected. The default Max Age is five minutes. However, if you use dynamic groups which have a requirement for finer grained time control, then you can adjust the Max Age on this cache holder to be just below the minimum time your finest grained time based dynamic group requires. The lower this value is, the more times the user's groups are queried during a session. Setting a value too high keeps the user/group relationships in memory perhaps longer than the user's session needlessly consuming memory.
DirectoryService.RolesMembershipCacheHolder	Caches the application role membership list by role.
DirectoryService.TeamManagerRuntime.Team	Caches the application team instances and team provisioning requests.

Cache Holder Name	Description
DirectoryService.UserCacheHolder	Caches users in the directory layer. Reading users from the directory layer involves both network communication (with LDAP server) and object creation. By default, the cache is limited to 1000 users, and the LRUs have a default TTL of 10 minutes. Depending on the user topography in your enterprise, you might need to adjust the maximum number of nodes or the TTL if you find the performance is suffering because of queries to the LDAP server for user objects. Making settings too high combined with a large number of different users logging in can cause unneeded memory consumption, and net lower performance from the server.
GlobalCacheHolder	The general purpose cache holder. This configuration applies to all caches that are not customizable (that is, all cache holders not listed in this table.)
JUICE	<p>Caches the resource bundles used by the user interface controls and DN display expression lookup results. Changing the setting of the cache holder has a performance impact for the DN display expression lookups because they are frequently used in the User Application.</p> <p>The low value should be at least 300 seconds, but a higher value than 900 seconds is ok. A lower value should be used if the customer is frequently changing the attributes that are used in the DN display expression</p>
RoleManager.RolesCacheHolder	Caches user role memberships listed by user.
Workflow.Model.Process	Caches the provisioning process XML object structure.
Workflow.Model.Request	Caches the provisioning request XML object structure.
Workflow.Provisioning	Caches provisioning request instances that have not completed. The default maximum capacity for the LRU cache is 500. The capacity can be modified by clicking the <i>Administration/Provisioning</i> tab and choosing the Engine and Cluster settings. The Process Cache Maximum Capacity appears on this page. This cache reduces the memory footprint for workflow processing without compromising performance.

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.

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 `Tomcat-Cluster` 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 cluster configuration settings on the *Caching* page determine whether to participate in a cluster and invalidates cache changes in the other nodes in that cluster. If clustering is enabled, the User Application accomplishes this by sending cache entry invalidation messages to each node as changes occur.

Preparing to Use a Cluster

To use caching across a cluster:

- 1 Set up your JGroups cluster. This involves using the User Application installation program to install the Identity Manager User Application to each application server in the cluster (see [Section 2.7, "Clustering," on page 58](#)).
- 2 Enable the use of that cluster in the User Application's cache configuration settings
See ["Configuring Cache Settings for Clusters" on page 98](#).

Configuring Cache Settings for Clusters

After you have a cluster ready to use, you can specify settings for the support of caching across that cluster.

- 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
<i>Cluster Enabled</i>	Select <i>True</i> to invalidate cache changes to the other nodes in the cluster specified by Group ID. If you don't want to participate in a cluster, select <i>False</i> .

Setting	What to do
<i>Group ID</i>	<p>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.</p> <p>Remember that the DefaultPartition and Tomcat-Cluster Group IDs are reserved for use by the JBoss Application Server.</p> <hr/> <p>TIP: To see the Group ID in logging messages, make sure that the level of the caching log (<code>com.sssw.fw.cachemgr</code>) is set to Info or higher.</p> <hr/>
<i>Cluster Properties</i>	<p>Specify the JGroups protocol stack for the cluster specified by Group ID. This setting is for experienced administrators who might need to adjust the cluster properties. Otherwise, you should not change the default protocol stack.</p> <p>To see the current cluster properties, click <i>view</i>.</p> <p>For details on the JGroups protocol stack, go to www.jboss.org/wiki/Wiki.jsp?page=JGroups (http://www.jboss.org/wiki/Wiki.jsp?page=JGroups).</p> <hr/>

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.

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.

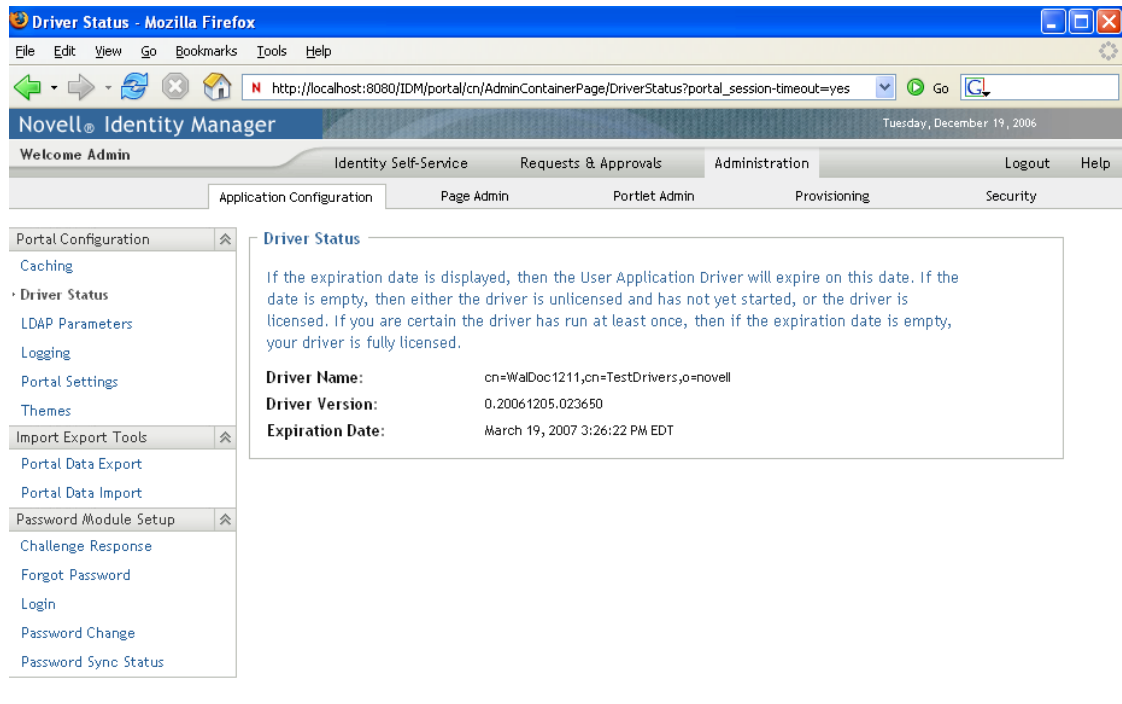
If you need to use the TCP protocol instead of the default UDP protocol, see “[Specifying the User Application Cluster Group Caching Configuration](#)” on page 67.

- 3 Click *Save*.
- 4 When you're ready for your saved settings to take effect, restart the User Application on the applicable application servers.

5.1.2 Driver Status

You can use the Driver Status pane to determine the expiration status of your driver.

Figure 5-1 Sample Driver Status for a “Trial” Driver



An Expiration Date value of *No Expiration* means that the driver has been started and is fully licensed or has not yet been started. If it has not been started, it might also be a trial driver. If there is an expiration date, then the driver is a trial driver and it has been started. The page itself describes what values of UNKNOWN mean.

5.1.3 LDAP Parameters

You can use the LDAP Parameters pane to:

- ♦ Change the credentials used by the Identity Manager User Application when connecting to the Identity Vault (LDAP provider)
- ♦ Change the credentials for the guest account, if your system is configured to use a specific guest account, rather than LDAP anonymous account.
- ♦ View other LDAP properties of the Identity Manager User Application. The values of these settings are determined when you install the User Application.

The user interface displays different fields depending on how you configured the guest account during installation. If you specified a guest account, the user interface includes fields that let you update the credentials for that account. If you have configured your system to use the LDAP Public Anonymous account, the user interface displays this message: The application is configured to use public anonymous account. To use a specific guest account, enable the guest account using the ldap configuration tool.

To administer LDAP connection parameters:

- 1 On the Application Configuration page, select *LDAP Connection Parameters* from the navigation menu on the left.

The LDAP Connection Parameters panel displays:

The screenshot shows the Novell Identity Manager web interface in Mozilla Firefox. The browser address bar shows the URL: `http://localhost:8080/IDM/portal/cn/AdminContainerPage/LDAPConfiguration`. The interface has a top navigation bar with tabs: Welcome Admin, Identity Self-Service, Requests & Approvals, Administration (selected), Logout, and Help. Below this is a sub-navigation bar with tabs: Application Configuration (selected), Page Admin, Portlet Admin, Provisioning, and Security. On the left is a sidebar menu with options: Portal Configuration (expanded), Caching, Driver Status, LDAP Parameters (selected), Logging, Portal Settings, Themes, Import Export Tools (expanded), Portal Data Export, Portal Data Import, Password Module Setup (expanded), Challenge Response, Forgot Password, Login, and Password Sync Status. The main content area is titled 'LDAP Parameters' and contains two sections. The first section, 'Change LDAP Connection Credentials', has a text box for 'Admin Username' containing 'cn=admin,o=novell', and empty text boxes for 'Admin Password' and 'Confirm Admin Password'. Below this is a text box for 'Guest Username' containing 'cn=bbrown,ou=users,ou=idmsample-aih,o=nc', and empty text boxes for 'Guest Password' and 'Confirm Guest Password'. A 'Submit' button is at the bottom of this section. The second section, 'LDAP Properties (Read Only)', displays a table of LDAP configuration details.

LDAP Properties (Read Only)	
Authority:	sigmund.qalab.wal.novell.com
Container Object:	locality
Container Object Attribute:	l
Container Object 2:	country
Container Object 2 Attribute:	c
Container Object 3:	organizationalUnit
Container Object 3 Attribute:	ou
Container Object 4:	organization
Container Object 4 Attribute:	o
Container Object 5:	domain
Container Object 5 Attribute:	dc
Dynamic Group Object:	dynamicGroup
Group Object:	groupOfNames
Group Root Container:	ou=groups,ou=idmsample-aih,o=novell
Group Search Scope:	subtree
Group User Member Attribute:	member
KeyStore Path:	c:\Program Files\jdk1.5.0_08\jre\lib\security\cacerts
Login Attribute:	cn
	true
Naming Attribute:	cn
Object Attribute:	objectClass
	389
Provisioning Driver DN:	cn=WalDoc070201,cn=TestDrivers,o=novell
Root Name:	ou=idmsample-aih,o=novell
	636
Use Dynamic Groups:	false

2 Examine and modify the settings, as appropriate. For details, see: “Settings You Can Change” on page 101.

3 If you make changes that you want to apply, click *Submit*.

Settings You Can Change

On the LDAP Connection Parameters panel, you can modify settings for the credentials for:

- ♦ The Identity Manager User Application whenever it connects to the Identity Vault (LDAP provider).
- ♦ The guest account (if configured).

The initial values for the credentials are specified during installation. These installation values are written to the `sys-configuration-xml` data file in the `jboss/server/configuration-name/conf` folder. If you make changes to these credentials via the Administration page, your changes are saved to the User Application’s database; they are not saved to the `sys-configuration-xml` data file. After values are written to the database, the User Application no longer checks the values written to the `sys-`

configuration.xmldata file. This means that you cannot use the configupdate utility to change the credentials because they are ignored. However, you can use configupdate to change the type of guest user (LDAP Guest or Public Anonymous Account).

Table 5-3 *LDAP Parameters*

Setting	What to do
Admin Username	<p>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.</p> <p>It is typical to specify the Identity Vault's <code>root</code> administrator as the LDAP connection username. The <code>root</code> administrator has full control over the tree, so you need not assign any special trustee rights.</p> <p>For example:</p> <pre>cn=admin,o=myorg</pre> <p>If you specify some other user, you need to assign inheritable trustee rights to the properties [All Attributes Rights] and [Entry Rights] on your User Application driver.</p> <hr/> <p>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.</p>
Admin Password and Confirm Password	<p>Type the password that is currently set for that username in the Identity Vault.</p>
Guest Username	Type the guest user's distinguished name
Confirm Guest Password	Type the password for the guest user.

If TLS is enabled for your LDAP server, you might encounter the following error when you update the Admin username and password: `Unable to authenticate to LDAP Provider`. Disable this error by disabling TLS via iManager.

5.1.4 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; after it reaches a certain size, it rolls over to another file. If you have 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 3, “Setting Up Logging,” on page 71](#).

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.

[Table 5-4 on page 103](#) lists and describes the logs.

Table 5-4 Identity Manager User Application Logs

Log Name	Description
com.novell	Parent of other Identity Manager User Application logs
com.novell.afw.portal.aggregation	Messages related to portal page processing
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)

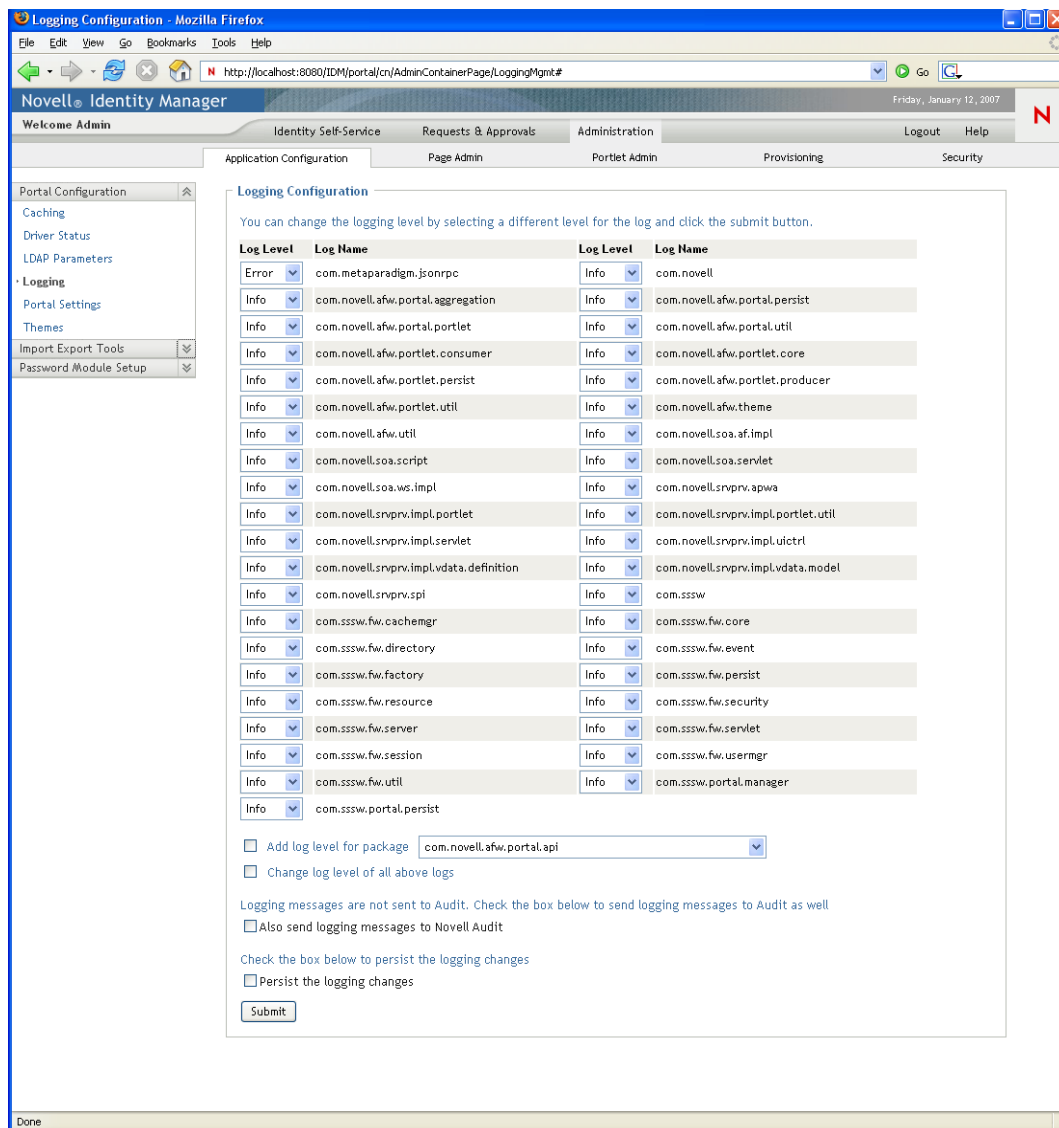
Log Name	Description
<code>com.novell.srvprv.impl.portlet.core</code>	Messages from the core identity portlets and password portlets
<code>com.novell.srvprv.impl.portlet.util</code>	Messages from the identity-related utility portlets
<code>com.novell.srvprv.impl.servlet</code>	Messages from the UI control framework's ajax servlet and ajax services
<code>com.novell.srvprv.impl.uictrl</code>	Messages from the UI control registry API and approval form rendering
<code>com.novell.srvprv.impl.vdata</code>	Messages from the directory abstraction layer
<code>com.novell.srvprv.spi</code>	Messages from the UI control registry API
<code>com.sssw.fw.cachemgr</code>	Messages related to the framework cache subsystem
<code>com.sssw.fw.core</code>	Messages related to the framework core subsystem
<code>com.sssw.fw.directory</code>	Messages related to the framework directory subsystem
<code>com.sssw.fw.event</code>	Messages related to the framework event subsystem
<code>com.sssw.fw.factory</code>	Messages related to the framework factory subsystem
<code>com.sssw.fw.persist</code>	Messages related to the framework persistence subsystem
<code>com.sssw.fw.resource</code>	Messages related to the framework resource subsystem
<code>com.sssw.fw.security</code>	Messages related to the framework security subsystem
<code>com.sssw.fw.server</code>	Messages related to the framework server subsystem
<code>com.sssw.fw.servlet</code>	Messages related to the framework servlet subsystem
<code>com.sssw.fw.session</code>	Messages related to the framework session subsystem
<code>com.sssw.fw.usermgr</code>	Messages related to the framework user subsystem
<code>com.sssw.fw.util</code>	Messages related to the framework utility subsystem
<code>com.sssw.portal.manager</code>	Messages related to the Portal Manager
<code>com.sssw.portal.persist</code>	Messages related to portal persistence

The User Application logs are hierarchical. For example, `com.novell` is the parent of other logs underneath it. Any additional logs inherit its properties.

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.

- 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 drop-down list 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.

Level	Description
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*.

You can change the log level for all of the logs to one setting by selecting *Change log level* of all above logs and using the drop-down list to select the level.

Adding Logs for Other Packages

You can add logs for other packages used by the User Application.

- 1 Go to the Logging page:
- 2 At the bottom of the page, select *Add Log Level for Package*, then use the drop-down list to select the package.
- 3 Choose a log level from the drop-down, then click *Submit*.

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 Select or deselect the following setting, as appropriate: *Also send logging messages to Audit*.
- 3 Click *Submit*.

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.

However, 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/idmuserapp_logging.xml
```

To toggle persistence of settings on or off:

- 1 Go to the Logging page.
- 2 Select or deselect the following setting, as appropriate: *Persist the logging changes*
- 3 Click *Submit*.

5.1.5 Portal Settings

You can use the Portal page to view characteristics of the Identity Manager User Application. The settings are for informational purposes and cannot be changed. The values of these settings are set in the User Application WAR. (*Default Theme* reflects your current theme choice from the Themes page.)

5.1.6 Theme Administration

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 just 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.

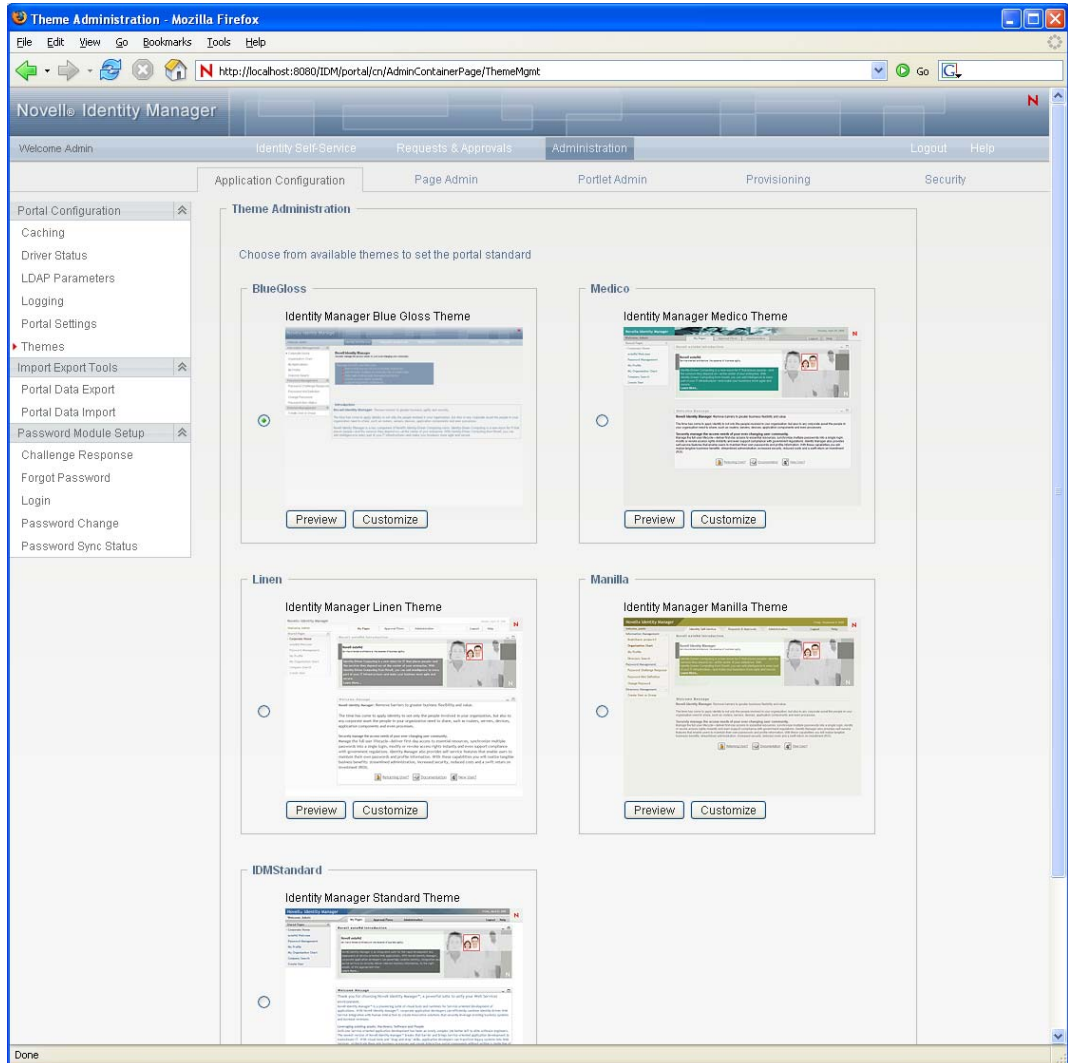
The Themes page also enables you to:

- ♦ Preview each theme choice to see how it looks
- ♦ Customize any theme choice to reflect your own branding (such as a logo)

Previewing a Theme

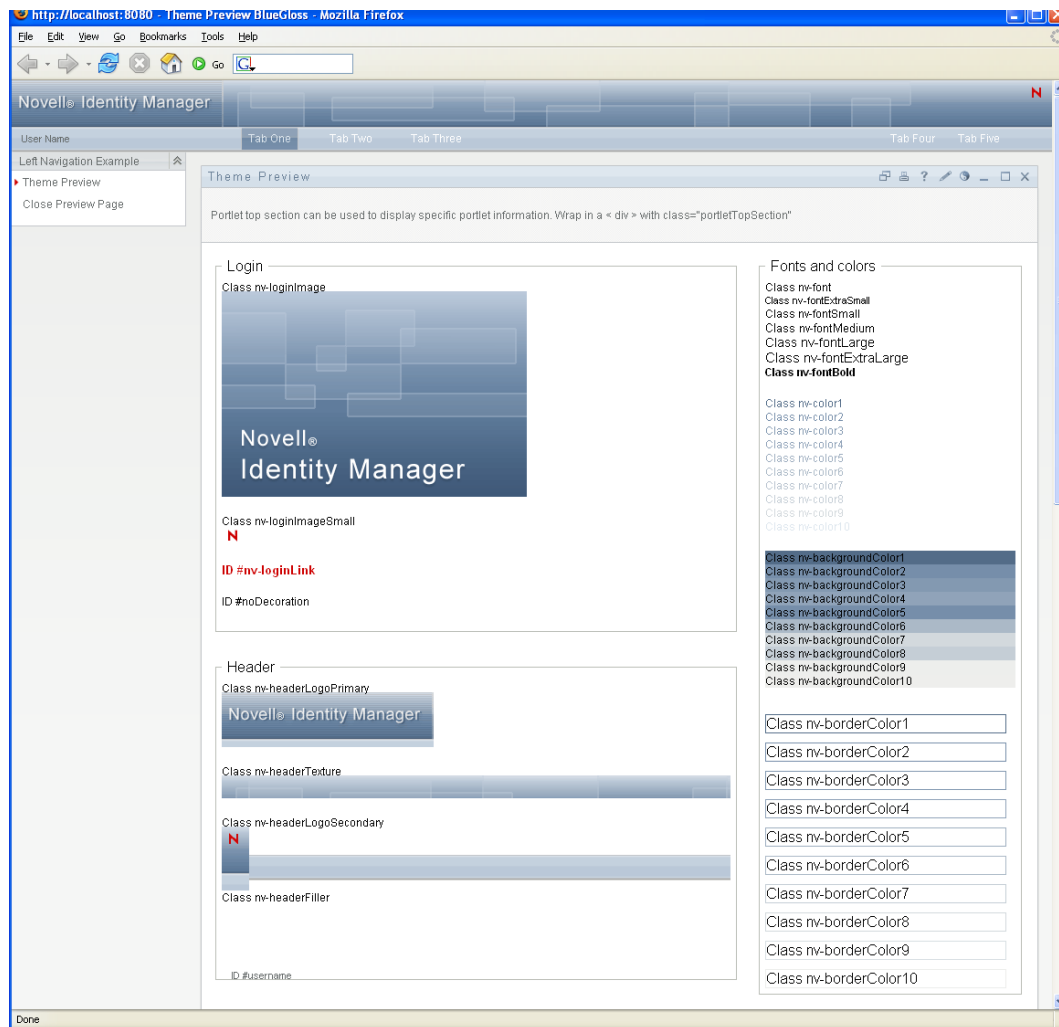
Before choosing a theme, you can preview how it will change the look of the Identity Manager user interface.

- 1 Go to the Themes page:



- 2 Find a theme that you are interested in, then click the corresponding *Preview* button.

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.

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.

- 1 Go to the Themes page.
- 2 Click the radio button for the theme you want.
- 3 Click the *Save* button.

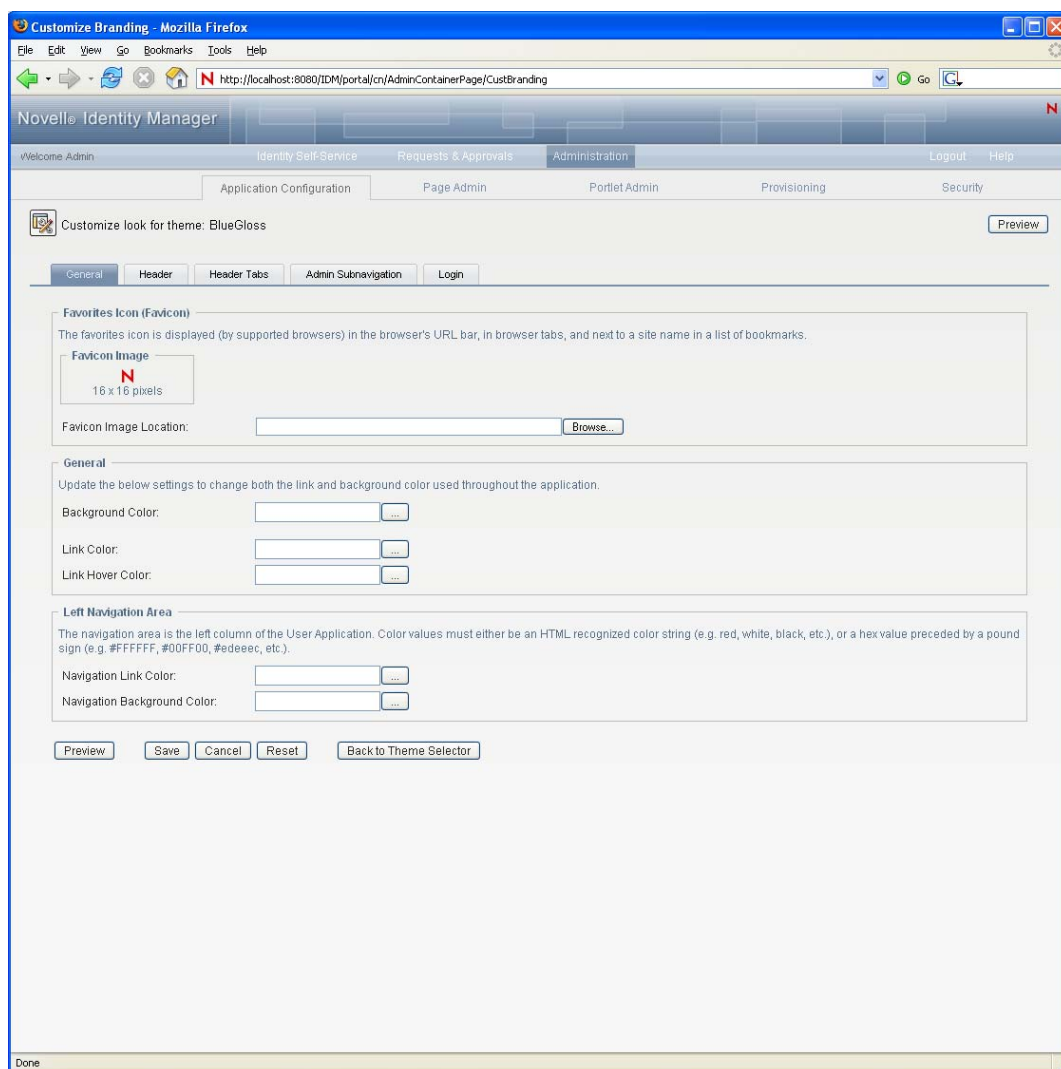
The look of the user interface changes to reflect your chosen theme.

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.

- 1 Go to the Themes page.
- 2 Find a theme that you want to customize, then click the corresponding *Customize* button.

The Themes page displays the Customize Branding settings for that theme:



- 3 Specify your customizations by changing the settings in one or more tabs (as needed). Each tab contains the settings for different parts of the User Application interface. They include:
 - ♦ *General*: Lets you specify general theming properties such as a favorites icon, background, link and hover color, and the left navigation area properties.
 - ♦ *Header*: Lets you specify the header color, texture, logo and username properties.
 - ♦ *Header tabs*: Lets you specify the properties for the header tabs.
 - ♦ *Admin subnavigation*: Lets you specify the properties for the *Admin* tab.

- ♦ *Login*: Lets you specify the properties for the login screen.

Follow the on-screen instructions for specifying each setting. The changes are not reflected in the User Application until you save them. If you have made unsaved changes, the *Save* button displays an asterisk * to indicate that the changes are pending a save.

4 Click *Save*.

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.

5 When you're done working on this theme, click *Back to Theme Selector*.

Defining a Custom Theme

You can also create and deploy your own custom themes and deploy them in their own WAR file. When they are deployed, the custom themes are available through the Themes management page of the *Administration* tab. Before attempting to create your own custom theme, make sure you have a working knowledge of the following technologies:

- ♦ The structure of J2EE WAR files, how to modify the contents of a WAR file, and how to deploy one to a JBoss application server.
- ♦ How to modify CSS and XML files
- ♦ How to create the graphic elements for your theme

Creating a Custom Theme

To create a custom theme, begin with a copy of an existing theme (such as *BlueGloss*) from the User Application WAR:

1 Back up the deployed User Application WAR file (IDM.WAR or IDMPProv.WAR).

This is located in the `jboss\server\serverconfig-name\deploy` subdirectory.

2 In a test environment, extract the contents of the User Application WAR file.

The files that comprise the User Application's themes are located in the `resources/themes` subdirectory. Each theme resides in its own directory with an appropriate name.

3 In the test environment, create a directory for the custom theme.

The directory name can be any valid directory name, but it should reflect the name of the theme, and it should not contain spaces.

4 Copy the contents of the BlueGloss theme from the extracted WAR file to the new subdirectory. You will be working with the following files:

File Name	Description
<code>theme.xml</code>	The theme descriptor file. It includes entries for display name and description. They are used in the <i>Themes</i> page of the <i>Administration</i> tab. The remaining entries correspond to the brandable selectors. The width and height attributes on these entries are used in the branding page to reference the exact dimensions needed when a user uploads a customized version of these images. These entries must match their respective images, width and height as found in the <code>themes.css</code> .

File Name	Description
<code>theme.css</code>	Contains the CSS selectors used to style the look and feel of the user interface.
<code>print.css</code>	Contains the CSS selectors used to style a print friendly version of the user interface.
An images subdirectory	Contains the images used by the theme.

Rules for working with these files:

- ♦ Do not change the names of the `theme.xml`, `theme.css`, and `print.css` files.
 - ♦ The CSS Selector names must remain the same, but you can change the properties of the selectors to establish the look and feel.
 - ♦ The images subdirectory can have any name, but you must reference it correctly in the CSS and XML files.
- 5** Make your changes to the images, CSS style sheets and other theme elements as needed. The following changes are recommended:
- ♦ In the `theme.xml` file:
 - ♦ **display-name:** Change this to a value that represents your theme. It displays as the Theme-name in the Themes page of the User Application's *Administration* tab.
 - ♦ **description:** Change this to a value that describes your theme. It displays as the Description in the Themes page of the User Application's *Administration* tab.
 - ♦ Consider whether to localize the *display-name* and *Description* fields.
 - ♦ In the graphics directory:
 - ♦ **thumbnails.gif:** Replace the copy with your own image. This image displays along with the Theme-name and Description of the theme (described above) that is shown in the Themes page of the *Administration* tab. It typically illustrates what the User Application landing page looks like when the associated theme is applied
 - ♦ **Renaming graphics files:** If you change the names of graphics files (rather than just substituting a different image of the same name), make sure to change the reference to the image in both the `theme.xml` and the `theme.css` file. If the image is not used in the branding interface (for example, if it is not listed as one of the subset of brandable images in the `theme.xml` file), then you will only need to change the reference to the image in the `theme.css` file. Suppose you want to rename `images/header_left.gif` to `images/my_company_name.gif`. Edit the `theme.css` file to reflect the new image name.
- 6** After you make all of the desired changes to the theme files, add your customized theme directory to a new WAR file that contains one or more custom themes. Deploy the new WAR to your test application server.
- Testing tip: Open the Themes page (available under the *Administration* tab). Your theme should display along with the prepackaged themes. Use the Theme Preview action to see how the customized changes to your new theme will render. This is a useful way to ensure that you have completed all of your intended changes to your theme.
- 7** After your changes are fully tested, you can deploy the WAR containing the custom theme to your production application server.

Any number of custom themes can reside in a single WAR. Any number of custom WARs containing custom themes can be deployed.

To undeploy the theme, remove the WAR that contains the theme from the application server's deploy directory. Before undeploying, make sure that any themes it contains are not defined as the User Application's default theme. If you remove the WAR and it does contain the default theme, the Theme Administration screen displays an error message and reverts the User Application theme to the original default theme defined at installation time.

Customizing the Theme for External Password WAR

If you configured Password Management to use an *External Password WAR*, the theme for the Forgot Password page is defined in that external password WAR. The default name for the external password WAR is `IDMPwdMgt.WAR`. The `IDMPwdMgt.WAR` contains one theme; by default, it is *BlueGloss*. It does not include a user interface for modifying or branding this theme.

You can define a custom theme for the external Forgot Password page. The procedure for defining a custom theme is described in [“Defining a Custom Theme” on page 111](#); however, the deployment procedure for the external Forgot Password page is different and the rules about the custom theme WAR are more restrictive. After you define the custom theme:

- ◆ Package the theme in a WAR named `IDMPwdMgtTheme.WAR`.
- ◆ The `IDMPwdMgtTheme.WAR` can contain a single theme, and the theme must be located in the `resource/themes/Theme` directory within the WAR.
- ◆ Deploy the `IDMPwdMgtTheme.WAR` on the application server where the external WAR is located. Only one custom theme can be deployed at a time.

5.2 Working with the Import and Export Tools

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

Table 5-5 *Portal Data Export and Import Tools*

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).

The Export and Import tools enable you to move the portal configuration state from one portal (User Application) to another, as needed. [Table 5-5 on page 113](#) describes how these tools work.

You can use the Portal Data Export and 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

5.2.1 Requirements

To use the Portal Data Export and 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.

5.2.2 Restrictions

You cannot use the Portal Data Export and Import tools to:

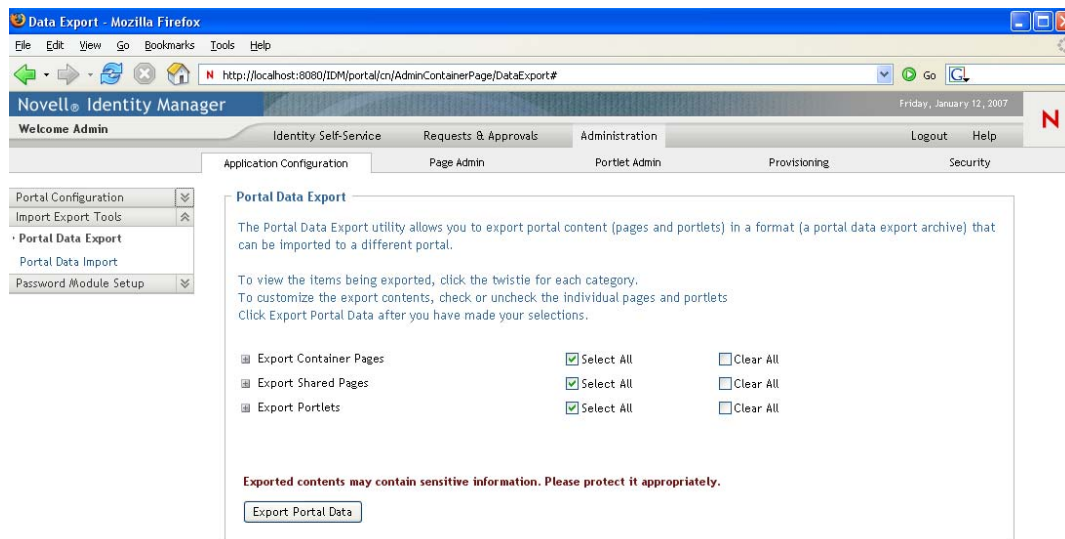
- ♦ Export or import portal configuration state when a server is currently servicing user requests
- ♦ Export or import portal classes and resources
- ♦ Export or import portlet classes and resources
- ♦ Export or import the identity and provisioning data used in a portal
- ♦ Export or import administration 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)

5.2.3 Exporting Portal Data

This section describes how to export a portal's configuration state to a portal data export ZIP file.

- 1 If you are performing an incremental update, back up the target portal.
- 2 On the Application Configuration page, select *Portal Data Export* from the navigation menu on the left.

The Portal Data Export panel displays:



- 3 Follow the on-screen instructions to select the portal pages and portlets that you want to export.

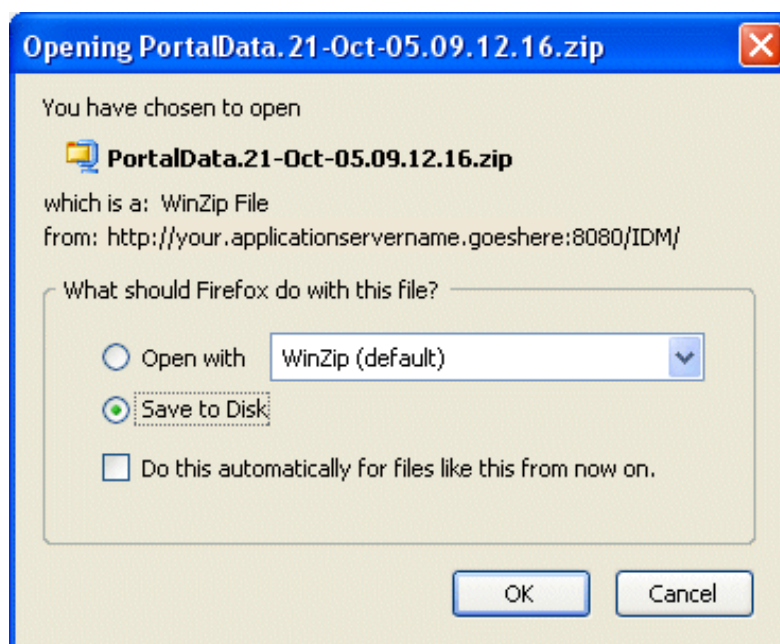
Some portlets that you have not selected for export might still be exported. If you export a page that contains a portlet, but do not select that portlet for export, the portlet is still exported (to ensure that a runtime error does not occur for the exported page).

- 4 When you are done making selections, click *Export Portal Data*.

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:



- 5 Save the portal data export ZIP file to an appropriate location.

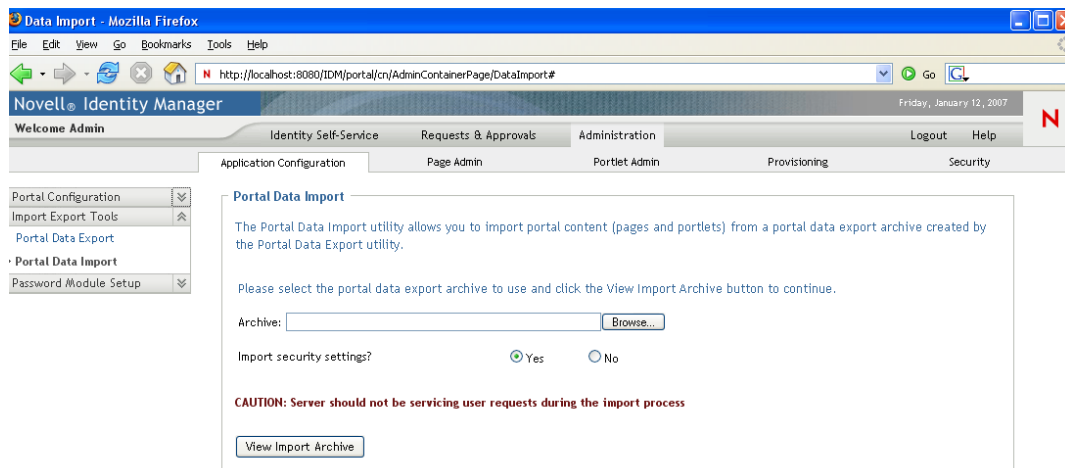
5.2.4 Importing Portal Data

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.

- 1 If you are performing an incremental update, back up the target portal.
- 2 On the Tools page, select *Portal Data Import* from the navigation menu on the left.

The Portal Data Import panel displays:

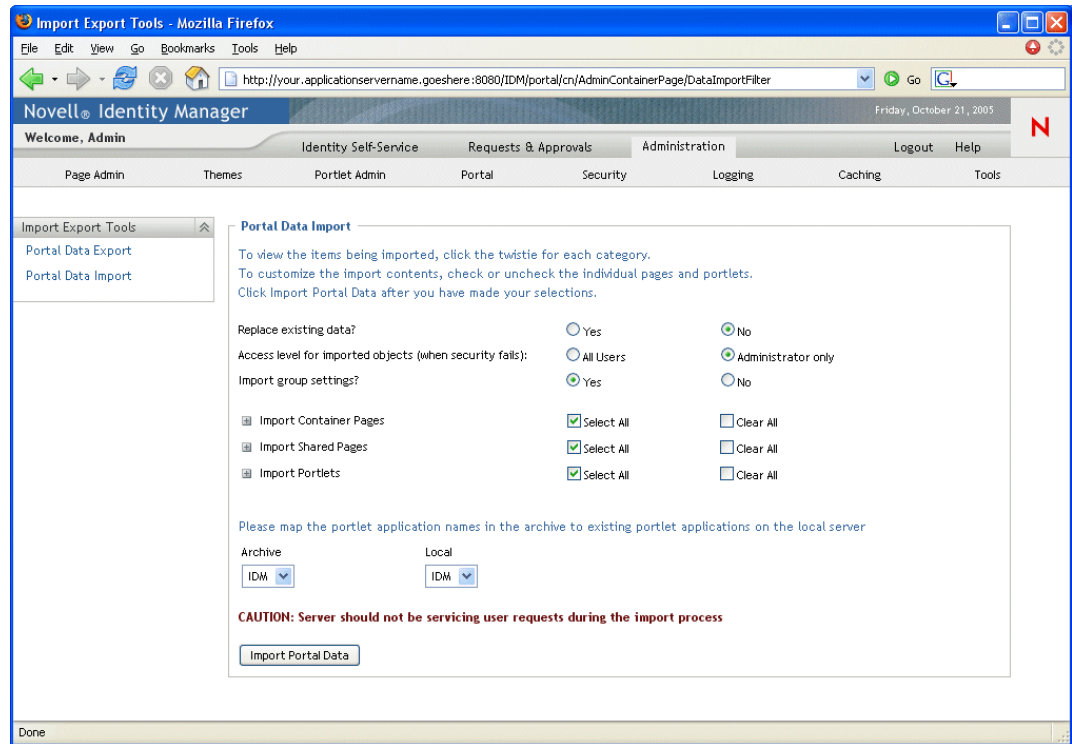


3 Specify the following general import settings:

Setting	What to Do
<i>Archive</i>	Click <i>Browse</i> to select the portal data export ZIP file to import. For example: PortalData.21-Oct-05.09.12.16.zip
<i>Import security settings?</i>	Select one of the following: <ul style="list-style-type: none">♦ 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 fail to be imported.♦ No: If you want to ignore the permissions that the portal data export ZIP file specifies.

4 Click *View Import Archive*.

The panel displays more specifics about your selected portal data export ZIP file and how you want to import it:



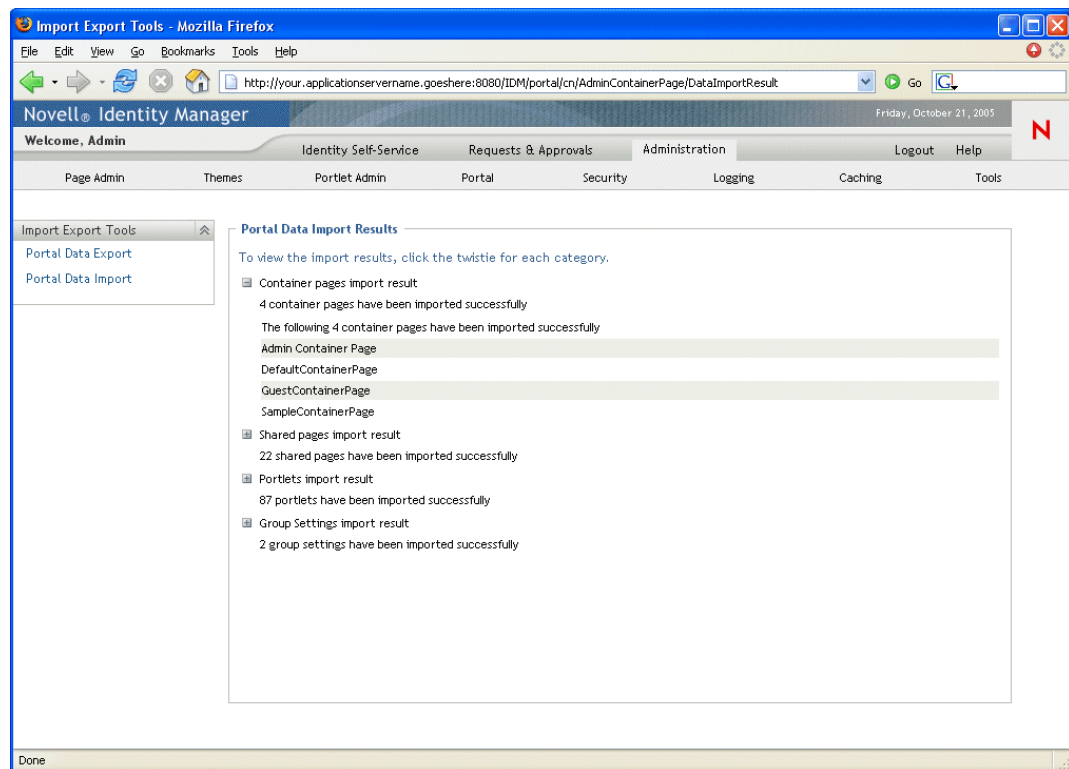
5 Specify the following detailed import settings:

Setting	What to Do
<i>Replace existing data?</i>	<p>Select one of the following:</p> <ul style="list-style-type: none"> ♦ 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 is overwritten in the target portal. ♦ No: If you want to skip the import for all existing pages and portlets.

Setting	What to Do
<i>Access level for imported objects</i>	<p>Select one of the following:</p> <ul style="list-style-type: none"> ♦ <i>All Users</i>: For unrestricted access to imported pages and portlets. ♦ <i>Administrator only</i>: For restricted access to imported pages and portlets. <p>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.</p> <p>If you chose not to import security settings, then this access level is applied to all pages and portlets that are imported.</p>
<i>Import group settings?</i>	<p>(If you chose to import security settings) Select one of the following:</p> <ul style="list-style-type: none"> ♦ <i>Yes</i>: 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 fail to be imported. ♦ <i>No</i>: If you want to ignore the default page assignments that the portal data export ZIP file specifies for groups.
<i>Import Container Pages</i>	<p>Follow the on-screen instructions to select the pages and portlets that you want to import from the portal data export ZIP file to the target portal.</p> <hr/> <p>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.</p> <hr/>
<i>Import Shared Pages</i>	
<i>Import Portlets</i>	
<i>Please map the portlet application names... Archive/Local</i>	<p>Use the <i>Archive</i> and <i>Local</i> drop-down 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.</p>

6 When you're ready to begin the import, click *Import Portal Data*.

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`

- 7 Test the target portal to ensure that you imported the data that you expected.

5.3 Password Management Configuration

This section describes how to configure password self-service and user authentication features to your Identity Manager User Application. Topics include:

- ◆ [Section 5.3.1, "About Password Management Features," on page 121](#)
- ◆ [Section 5.3.2, "Configuring Challenge Response," on page 125](#)
- ◆ [Section 5.3.3, "Configuring Forgotten Password," on page 126](#)
- ◆ [Section 5.3.4, "Configuring Login," on page 129](#)
- ◆ [Section 5.3.7, "Configuring Change Password," on page 135](#)
- ◆ [Section 5.3.5, "Configuring Password Sync Status," on page 131](#)
- ◆ [Section 5.3.6, "Configuring Password Hint Change," on page 134](#)
- ◆ [Section 5.3.7, "Configuring Change Password," on page 135](#)

5.3.1 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 (username 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). You can configure forgotten password self-service to run inside the firewall (the default), or you can configure it to run outside the firewall.
- ♦ Provide users with challenge question self-service
- ♦ Provide users with password hint self-service

Required Setup in eDirectory

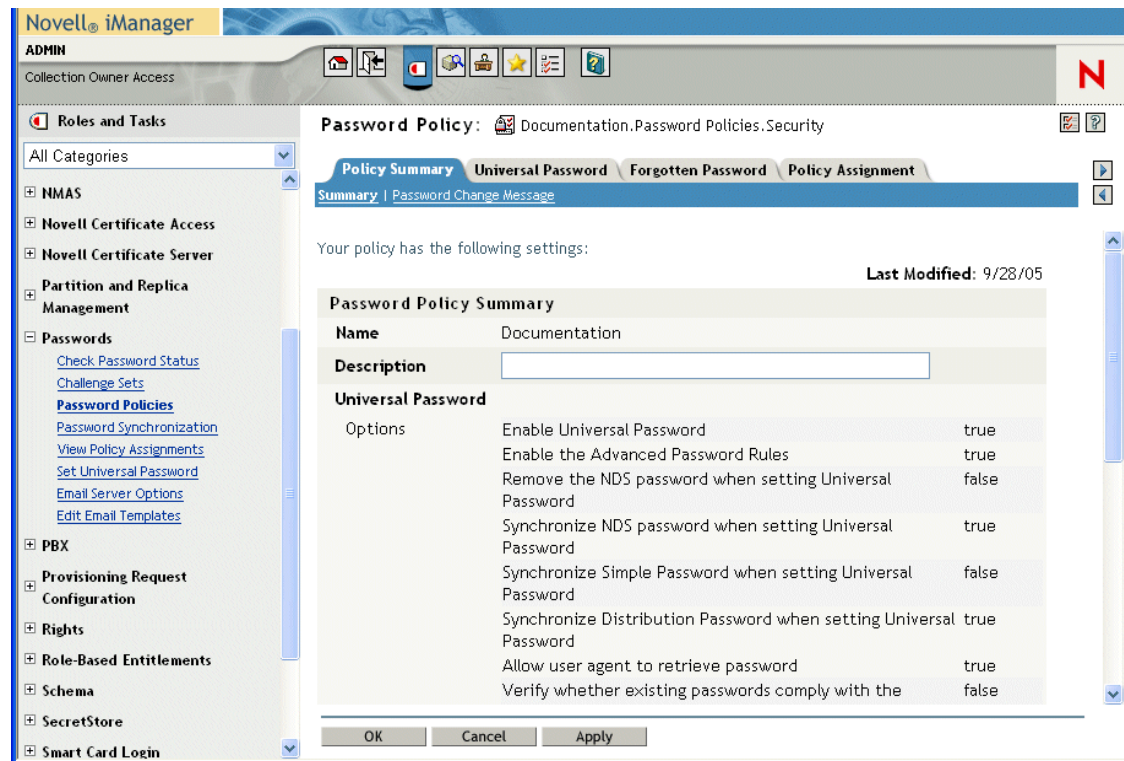
Before you can use most of the password self-service and user authentication features, you need to do the following in eDirectory:

- ♦ 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.

Figure 5-2 Sample Password Policy



This password policy specifies:

- ◆ Universal Password settings

Figure 5-3 Sample Universal Password Settings

The screenshot shows the Novell iManager interface with the 'Universal Password' tab selected under the 'Password Policy' section. The left sidebar lists various administrative tasks, including 'Passwords', 'PBX', 'Provisioning Request Configuration', 'Rights', 'Role-Based Entitlements', 'Schema', 'SecretStore', and 'Smart Card Login'. The main content area displays the 'Advanced Password Rules' configuration for the 'Documentation.Password Policies.Security' policy.

Password Policy: Documentation.Password Policies.Security

Policy Summary | Universal Password | Forgotten Password | Policy Assignment

Advanced Password Rules

Add	Description	Value
Change Password		
<input checked="" type="checkbox"/>	Allow user to initiate password change	
<input checked="" type="checkbox"/>	Require unique passwords	
<input type="checkbox"/>	Limit the number of passwords to store in the history list (1-255)	<input type="text"/> Password(s)
<input checked="" type="checkbox"/>	Limit the number of days to store a password in the history list (0-365)	<input type="text" value="180"/> Day(s)
Password Lifetime		
<input type="checkbox"/>	Number of days before password can be changed (0-365)	<input type="text"/> Day(s)
<input checked="" type="checkbox"/>	Number of days before password expires (0-365)	<input type="text" value="90"/> Day(s)
<input type="checkbox"/>	Limit the number of grace logins allowed (0-254)	<input type="text"/> Attempt(s)
Password Length		
<input checked="" type="checkbox"/>	Minimum number of characters in password (1-512)	<input type="text" value="4"/> Characters

OK Cancel Apply

- ◆ Settings to deal with forgotten-password situations

Figure 5-4 Sample Password Policy

The screenshot shows the Novell iManager interface with the 'Forgotten Password' tab selected under the 'Password Policy' section. The left sidebar is the same as in Figure 5-3. The main content area displays the 'Forgotten Password' configuration for the 'Documentation.Password Policies.Security' policy.

Password Policy: Documentation.Password Policies.Security

Policy Summary | Universal Password | Forgotten Password | Policy Assignment

Select an action for a forgotten password request. The most secure method of user verification is to use challenge sets, which require a user to answer a set of questions to prove his or her identity. Alternatively, you may select an action that occurs without the user answering a challenge set.

☒ Enable Forgotten Password

Challenge Set

☒ Require a challenge set

Use the [Challenge Sets](#) task to add a new Challenge Set to your list.

Action

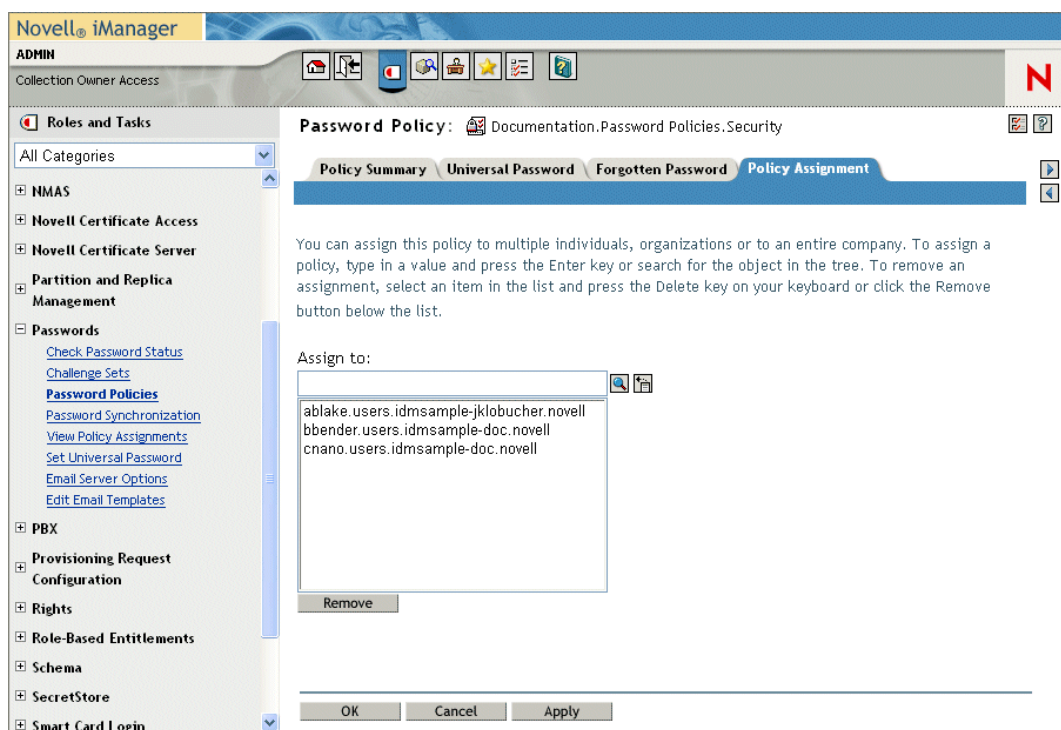
Choose an action:

- ☐ Allow user to reset password (Requires challenge set and Universal Password options)
- ☐ E-mail current password to user (Requires challenge set and Universal Password options)
- ☐ E-mail hint to user
- ☒ Show hint on page

OK Cancel Apply

- ◆ Assignments that apply the policy to specific users

Figure 5-5 Sample Policy Assignments



For more information on setting up Universal Password and password policies in eDirectory, see the [Novell Identity Manager Administration Guide \(http://www.novell.com/documentation/dirxml20/index.html\)](http://www.novell.com/documentation/dirxml20/index.html).

Case-Sensitive Passwords

By default passwords are not case-sensitive. You can create a password policy that allows case-sensitive passwords. You can specify the *Allow the password to be case-sensitive* in the *Password Policies > Universal Password > Advanced Password Rules*. If you enable case-sensitive password, you must also enable the *Allow user to retrieve password* setting. It is enabled by default, but you can verify it through the iManager *Password Policies > Universal Password > Configuration Options* tab.

Password Policy Compliance

If you enable Universal Password, it is recommended that you also configure the system to verify that existing passwords comply with the password policy. You can configure this through iManager. In iManager, go to *Passwords > Password Policies > Universal Password > Configuration Options*. Make sure the following option is selected: *Verify whether existing passwords comply with password policy (verification occurs on login)*. This ensures that users created through the User Application are forwarded to the Change Password page to enter a password that complies with the Identity Manager password policy.

5.3.2 Configuring Challenge Response

The Challenge Response self-service page 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

TIP: If you have localized the Challenge Response questions in iManager set the *Login* Configuration setting **Enable Locale Check** to True.

Figure 5-6 Challenge Response Example

Novell® Identity Manager Wednesday, December 13, 2006

Welcome Abby Identity Self-Service Requests & Approvals Logout Help

Information Management Organization Chart My Applications My Profile Directory Search Password Management Password Challenge Response Password Hint Change Change Password Password Policy Status Password Sync Status

IDM Challenge Response

These questions are assigned to your password policy. For all Admin-Defined Questions, provide a response. For all User-Defined Questions, create your own question and provide a response.

Admin Defined Challenge Questions

Question: What is your mother's maiden name? Response:

Question: What is your childhood pet's name? Response:

User Defined Challenge Questions

Question: Response:

Submit

Requirements

The Challenge Response requirements are described [Table 5-6 on page 125](#).

Table 5-6 Challenge Response Requirements

Topic	Requirements
Password policy	A password policy with forgotten password enabled and a challenge set.
Universal Password	Does not require Universal Password to be enabled.
eDirectory configuration	Requires that you grant supervisor rights to the LDAP 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 <i>[All attribute rights]</i> .

Using the Challenge Response Feature

To use the Challenge Response feature, you need to know about the following:

- ♦ [“How Challenge Response Is Used During Login” on page 126](#)
- ♦ [“How Challenge Response Is Used in the User Application” on page 126](#)

How Challenge Response Is Used During Login

During the login process, the Login page automatically redirects to Challenge Response 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 Challenge Response Is Used in the User Application

By default, the User Application provides users with self-service for changing challenge questions and responses.

Configuring Challenge Response

The Challenge Response Configuration settings (on the *Administration* tab) are described in the following table.

Table 5-7 Challenge Response Configuration Settings

Setting	Description
<i>Mask Response Text</i>	Choosing Yes means that user-entered response text is masked with the *** characters.

5.3.3 Configuring Forgotten Password

This feature uses challenge/response authentication to let users get information about their passwords. The result, which depends on the assigned password policy, can include:

- ♦ Displaying the user’s password hint on the screen
- ♦ E-mailing the hint to the user
- ♦ E-mailing the password to the user
- ♦ Prompting the user to reset (change) the password

Forgotten password self-service is typically available to users inside your corporate firewall through the deployed User Application WAR, but you can also configure your system so that the forgot password management features are stored in a separate password management WAR. You can then deploy the password management WAR on a separate system that can be located inside or outside your corporate firewall. To learn how to setup forgot password outside the core User Application WAR, see [Section 2.5, “Configuring Forgotten Password Self-Service,” on page 50](#).

Requirements

The Forgot Password feature requirements are listed in [Table 5-8 on page 127](#).

Table 5-8 *Forgotten Password Requirements*

Topic	Requirements
Password policy	Requires a password policy with forgotten password enabled and with a challenge set.
Universal Password	Does not require Universal Password to be enabled, unless you want to support resetting the password or e-mailing the password to the user.

Using the Forgotten Password Feature

To use the Forgotten Password feature, you need to know about the following:

- ♦ “How the Forgotten Password feature Is Used During Login” on page 127
- ♦ “Configuring Your Environment for E-mail Actions” on page 127
- ♦ “Forgotten Password Configuration Settings” on page 128

How the Forgotten Password feature Is Used During Login

During the login process, the Login page redirects to the Forgotten Password page if the user clicks the *Forgot Password* link. When Forgotten Password displays, it does the following:

1. Prompts for username.
2. Redirects to the Challenge/Response page 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 Change password page so the user can reset their password
 - ♦ E-mails the password or hint to the user
 - ♦ Displays the hint

Configuring Your Environment for E-mail Actions

If you want to support the forgotten password e-mail actions, you need to make sure your e-mail 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*.

Forgotten Password uses two e-mail templates. In iManager, you 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. The Forgotten Password page determines, based on the user’s preferred locale, whether to display a localized e-mail template.

Forgotten Password Configuration Settings

You set the Forgotten Password page configuration settings in the *Administration* tab. They are described in [Table 5-9 on page 128](#).

Table 5-9 *Forgotten Password Configuration Settings*

Configuration Setting	Description
<i>Login Sequence</i>	The NMAS login sequence to use. In this version, only Challenge Response is supported.
<i>LDAP secure port</i>	The secure LDAP port to use. The default is 636.
<i>Allow Wild Cards in Login</i>	Select True if you want users to be able to type a wildcard character when entering the username. (The default is false.) If set to True, Display DN Information must also be True. When True, the user is able to type a few characters of a username followed by a wild card character and the Forgot Password page returns a list of DNs that match the user-entered string.
<i>Display DN Information</i>	Select True when you want the Forgot Password page to display DN values. This can be used in conjunction with Allow Wild Cards in Login . If set to False, no DN context information is displayed.
<i>Generic Password Policy User DN</i>	Specify the DN of an existing Identity Vault user established to prevent unauthorized users from accessing your system by guessing valid usernames. By default, if the user enters an invalid name, the User Application displays the message <i>User not Found</i> . Under some circumstances an unauthorized user might be able to guess a valid name and answer the challenge questions correctly. One way to prevent this is to specify this value. See “Setting Up a Generic Password Policy User DN” on page 128 for additional required configuration steps.
<i>Encoding</i>	The character encoding to use. The default is utf-8.

Setting Up a Generic Password Policy User DN

To support the [Generic Password Policy User DN](#), you need to set up a user in the users container for this purpose. This user should:

- ♦ Have a password that is difficult to guess.
- ♦ Have his or her e-mail address assigned to a User Application Administrator.

You must set up:

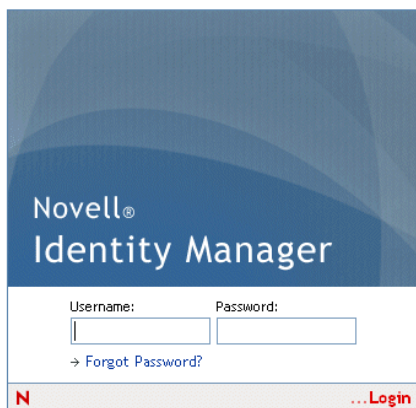
- ♦ A Challenge Set for this user and establish only Admin defined questions.
- ♦ A Password Policy that uses this Challenge Set. The Password Policy should have ForgotPassword enabled

You must log in to the User Application as this user at least once to supply the answers to the Admin-defined questions.

Finally, log in to the User Application as the User Application administrator and go to the *Forgot Password* configuration page of the *Administration* tab. Specify false for **Allow Wild Cards in Login** and **Display DN Information**. Specify this newly established user as the **Generic Password Policy User DN**.

5.3.4 Configuring Login

The Login page performs a very robust user authentication supported by Identity Manager (through Universal Password, password policies, and NMAS). The Login page redirects to the other password pages as needed during the login process.



Requirements

The Login page requirements are listed in **Table 5-10** below.

Table 5-10 *Login Requirements*

Topic	Requirements
Password policy	This page does not require a password policy, unless you want to use advanced password rules or let users click the <i>Forgot Password</i> link.
Universal Password	This page does not require Universal Password to be enabled, unless you want to use a password policy with advanced password rules.
SSL	This page uses SSL, so make sure that your application server is properly configured to support SSL connections to your LDAP realm.

Use the *Password Module Setup Login Action* to configure the following settings:

Table 5-11 *Login Configuration Settings*

Configuration Setting	Description
<i>Allow ID Wildcard</i>	If True, users can specify the first few characters of a username and a list of usernames that include those characters is displayed so the use can select the user to login as.

Configuration Setting	Description
<i>Enable Forgot Password Link</i>	If True, the User Application Login page displays the <i>Forgot Password</i> link.
<i>Forgot Password Link</i>	<p>This value defines the name and path to the Forgotten Password page. This initial value is established during installation. If you do not use an external password management WAR, you can leave the default value.</p> <p>For more information, see Section 2.5, “Configuring Forgotten Password Self-Service,” on page 50.</p>
<i>Forgot Password Return Link</i>	<p>Like the Forgot Password Link, this value is set during installation and you do not need to make any changes if you do not use an external password management WAR.</p> <p>If you do use an external password WAR, use this setting to specify the URL that the Forgot Password page can use to return to the User Application when the user clicks <i>Submit</i>. The return link should take the form of:</p> <pre>protocol://servername:port/userappcontext</pre> <p>For example, <code>https://idmhost:8080/IDMProv</code></p> <p>For more information, see Section 2.5, “Configuring Forgotten Password Self-Service,” on page 50.</p>
Enable SSO	If True, the Username and password are stored in the session and can be accessed by other properly configured portlets. The username is stored in the User ID Key and the password in the Password Key
User ID Key	If Enable SSO is True the username is stored in the session using this key.
Password Key	if Enable SSO is True the password is stored in the session using this key.
Enable Hint Migration	If True, any existing hints are moved from the <code>nsimHint</code> to the <code>nsimPasswordReminder</code> .
Enable Locale Check	If True, and the user has not set their locale preferences, the User Application displays a page that allows them to set their preferred locale.

Using the Login Page

To use the Login page, you need to know about the following:

- ♦ [“How Login Redirects to Other Pages” on page 130](#)
- ♦ [“Using Grace Logins” on page 131](#)

How Login Redirects to Other Pages

At runtime, the Login page redirects to other password pages, depending on what’s needed to complete the login process. [Table 5-12 on page 131](#) directs you to descriptions.

Table 5-12 Login Directions to Other Pages

If the user	Login redirects to
<i>Clicks the link Forgot Password</i>	Forgotten password page
<i>Needs to set up challenge questions and responses</i>	Challenge response page
<i>Needs to set up a password hint</i>	Hint Definition page
<i>Needs to reset an invalid password</i>	Change password page

Using Grace Logins

If you use a grace login, the Login page 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 Login page redirects you to the Change Password page.

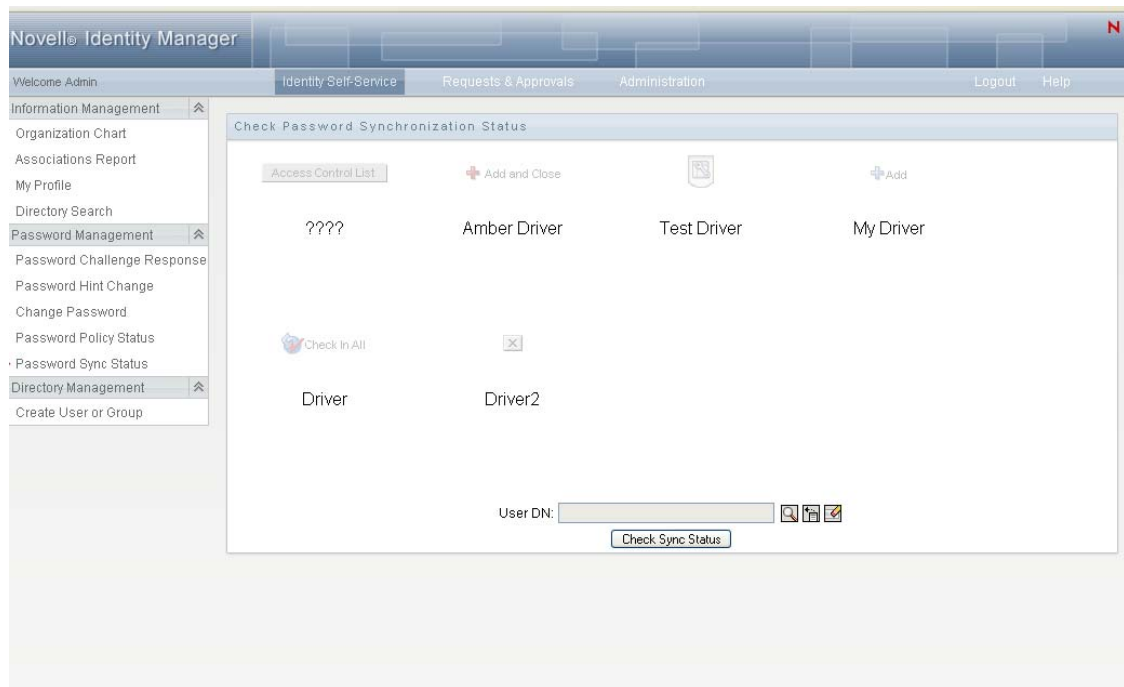
5.3.5 Configuring Password Sync Status

Password Sync Status lets users check the progress of the password change process on connected systems. You can specify a different image to represent each connected system. To set up password sync status checking:

- ◆ Define the connected applications whose status the user should be able to view during the synchronization process. You define the connected applications in the Password Sync Status Application Settings described in [Table 5-14 on page 134](#).
- ◆ Define the settings for the password sync status page displayed to users. These settings are described in [Table 5-13, “Password Sync Status Client Settings,” on page 133](#).

By default, the User Application Administrator can view the password sync status of other users when the User Application Administrator accesses the Password Sync Status page, shown in [Figure 5-7 on page 132](#). The administrator can access the sync status for another user by specifying the other user’s DN, then clicking *Check Sync Status*.

Figure 5-7 Password Sync Status



In addition to the User Application Administrator, you can define a set of users to perform the Check Sync Status for other users (for troubleshooting or other purposes). The members of a group called PasswordManagement are also automatically allowed to view the password synchronization status of other users. This group does not exist by default. If you choose to create this group, it must be:

- ◆ Named PasswordManagement.
- ◆ Given privileges to the Identity Vault. The group must have rights to read the user's eDirectory object attribute for users whose password synchronization status they need to view.

Table 5-13 Password Sync Status Client Settings

Configuration Setting	Description
<i>Password Sync Buffer Time (milliseconds)</i>	<p>The password sync status checking compares time stamps across different Identity Vaults and connected systems. This buffer time is intended to account for differences between the system times on these different machines. This time is added to the time stamp on the user object's password change attribute to determine if a change has occurred. It is used like this:</p> <p>The Password Sync Status process uses the buffer time as follows:</p> <ul style="list-style-type: none">♦ If the time stamp value (password sync time) in DirXML-PasswordSyncStatus for the connected system is older than the last password change time stamp (pwdChangedTime attribute of user object) + password sync buffer time, then the status is considered old and the system continues polling for an updated status for the connected system.♦ If the time stamp value in DirXML-PasswordSyncStatus for the connected system is newer than the last password change time stamp + password sync buffer time, then the password sync functionality returns the status code or message and displays the updated status of the connected system.♦ The last password change time stamp is populated to the user object after the user's password change. This functionality is available in NMAS 3.1.3 and higher.
<i>Image Per Row</i>	The number of application images to display per row in the Identity Self-Service Password Sync Status page.
<i>Individual Application Timeout (milliseconds)</i>	The amount of time that the Password Sync Status process waits for a response for each connected application's status before checking for the next one.
<i>All Application Timeout (milliseconds)</i>	This value indicates the amount of time allowed for the entire password sync status process (of all connected systems) to complete. Before this timeout is reached, the password sync process continues to poll until all status values are updated or this timeout is reached. When the timeout status is reached, the system displays an error message to the user that indicates that a timeout condition has been reached.
<i>Process Count</i>	The number of times each connected system is checked for the password sync status.
<i>Pass Phrase</i>	If the DirXML-PasswordSyncStatus contains a password hash, then the value entered in this field is compared to that value. If they are not equal, the User Application displays an invalid hash message.
<i>Application Image Size Limit (bytes)</i>	Lets you set the maximum size (in bytes) of the application image that can be uploaded. You specify this image in the Application Image setting described in Table 5-14 .

The password Sync Status Application Settings are described in [Table 5-14](#).

Table 5-14 Password Sync Status Application Settings

Configuration Setting	Description
<i>Password Synchronization Application Name</i>	<p>The name used to describe the connected application. You can enter the application name in multiple locales.</p> <p>To add a language:</p> <ol style="list-style-type: none">1. Click <i>Add Language (+)</i>.2. Type the Application Name for the desired localized languages in the appropriate field.3. Click <i>Save</i>. <p>If you do not specify localized application names, the value specified in the <i>Password Synchronization Application Name</i> is used.</p>
<i>Application DirXML-PasswordSyncStatus GUID</i>	<p>You can get the driver GUID by browsing the attributes on the driver object.</p> <p>To obtain the driver GUID:</p> <p>Click the browse button (next to this field) or use iManager (<i>General - Other</i> tab when modifying the object.)</p> <p>If you use the browse button on this page, you can only obtain the GUID of the driver associated with the current User Application.</p> <p>To obtain the User Application Driver GUID stored in a different Identity Vault tree, you must manually copy and paste it into this field.</p>
<i>Application Image</i>	<p>The name of the connected application Image to upload.</p>

5.3.6 Configuring Password Hint Change

This self-service page lets users set up or change their password hints, which can be displayed or e-mailed as a clue in forgotten password situations.

Figure 5-8 Define Password Hint Sample

Requirements

The Password Hint Change requirements are listed in [Table 5-15](#).

Table 5-15 Password Hint Change Requirements

Topic	Requirements
Universal Password	Does not require Universal Password to be enabled.

Using the Password Hint Change Page

To use the Password Hint Change page, you need to know about the following:

- ♦ [“How Password Hint Change Is Used During Login” on page 135](#)
- ♦ [“Using Password Hint Change in the User Application” on page 135](#)

How Password Hint Change Is Used During Login

During the login process, the Login page automatically redirects to the Password Hint Change page whenever users need to set up their password hints. 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 has Forgotten Password enabled and has the action set to *Email hint to user* or *Show hint on page*.

Using Password Hint Change in the User Application

By default, the User Application provides users with self-service for changing a password hint.

5.3.7 Configuring Change Password

This self-service page lets users change (reset) their Universal Passwords, 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 page changes the user’s eDirectory (simple) password, as permitted in the user's Password Restrictions.

Figure 5-9 *Change Password*

Novell® Identity Manager Wednesday, December 13, 2006

Welcome Abby Identity Self-Service Requests & Approvals Logout Help

Information Management
Organization Chart
My Applications
My Profile
Directory Search
Password Management
Password Challenge Response
Password Hint Change
Change Password
Password Policy Status
Password Sync Status

Change Password

Test Password Change Message.

Your password must have the following properties:

You may use numbers in your password.

The password is case-sensitive.

You may use special characters in your password.

You must use a unique password.

- Minimum number of characters in password: 4
- Maximum number of characters in password: 12

Old password:

New password:

Retype password:

There are no Password Change configuration settings.

Requirements

The Change Password page requirements are listed in [Table 5-16](#).

Table 5-16 *Change Password Requirements*

Topic	Requirements
Directory Abstraction Layer configuration	No directory abstraction layer configuration is required for this page.
Password policy	This page does not require a password policy, unless you want to use advanced password rules (with Universal Password enabled).
Universal Password	<p>To use this page for a Universal Password, the setting <i>Allow user to initiate password change</i> must be enabled in the Advanced Password Rules of the user's assigned password policy.</p> <p>To use this page for an eDirectory (simple) password, the setting <i>Allow user to change password</i> must be enabled in the user's Password Restrictions.</p>

Using the Change Password Page

To use the Change Password page, you need to know about the following:

- ♦ “How Change Password Is Used During Login” on page 137
- ♦ “Using Change Password in the User Application” on page 137

How Change Password Is Used During Login

During the login process, the Login page automatically redirects to the Change Password page 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 Forgot Password page also redirects to Change Password automatically if the user's assigned password policy specifies reset password as the action for forgotten password situations.

Using Change Password in the User Application

By default, the User Application provides users with the password change self-service using the Change Password page.

Page Administration

6

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

- ♦ [Section 6.1, “About Page Administration,” on page 139](#)
- ♦ [Section 6.2, “Creating and Maintaining Container Pages,” on page 147](#)
- ♦ [Section 6.3, “Creating and Maintaining Shared Pages,” on page 155](#)
- ♦ [Section 6.4, “Assigning Permissions for Pages,” on page 163](#)
- ♦ [Section 6.5, “Setting Default Pages for Groups,” on page 168](#)
- ♦ [Section 6.6, “Selecting a Default Shared Page for a Container Page,” on page 170](#)

For more general information about accessing and working with the *Administration* tab, see [Chapter 4, “Using the Administration Tab,” on page 83](#).

6.1 About Page Administration

You 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.

Table 6-1 *Page Types*

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 7, “Portlet Administration,” on page 173](#) and [Part IV, “Portlet Reference,” on page 205](#).

6.1.1 About Container Pages

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

- ♦ [“GuestContainerPage” on page 140](#)
- ♦ [“DefaultContainerPage” on page 142](#)
- ♦ [“Admin Container Page” on page 144](#)

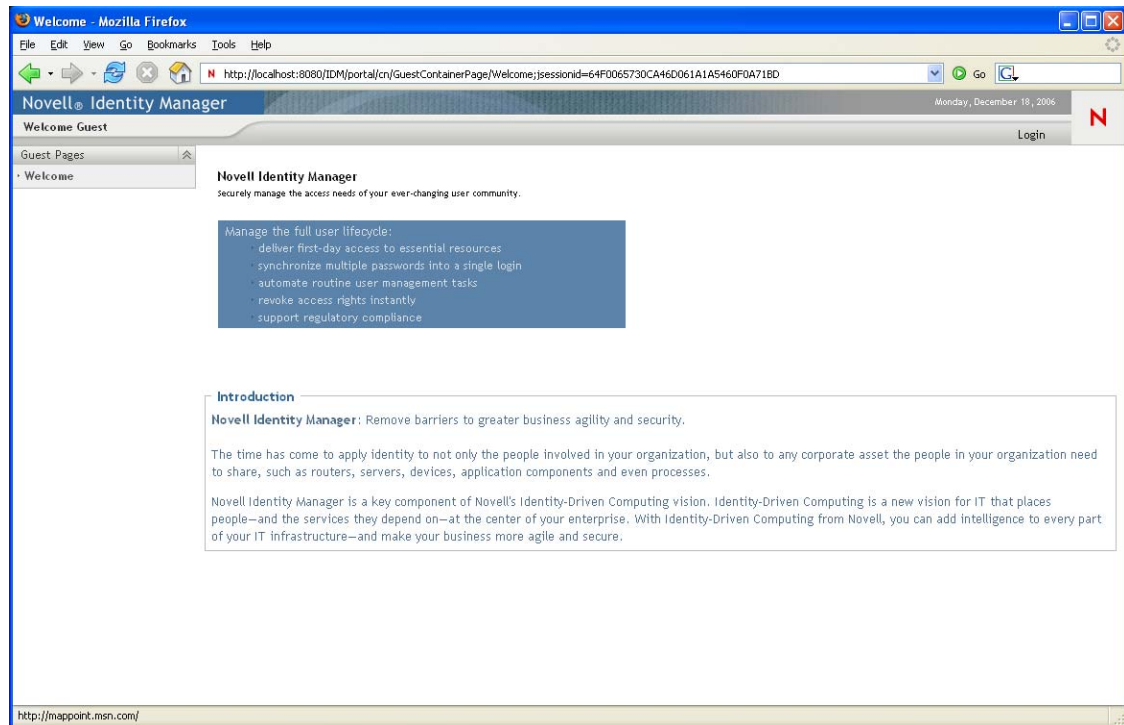
Keep in mind that you can modify these container pages if necessary. You also have the option of adding your own container pages.

To learn about working with container pages, see [Section 6.2, “Creating and Maintaining Container Pages,”](#) on page 147.

GuestContainerPage

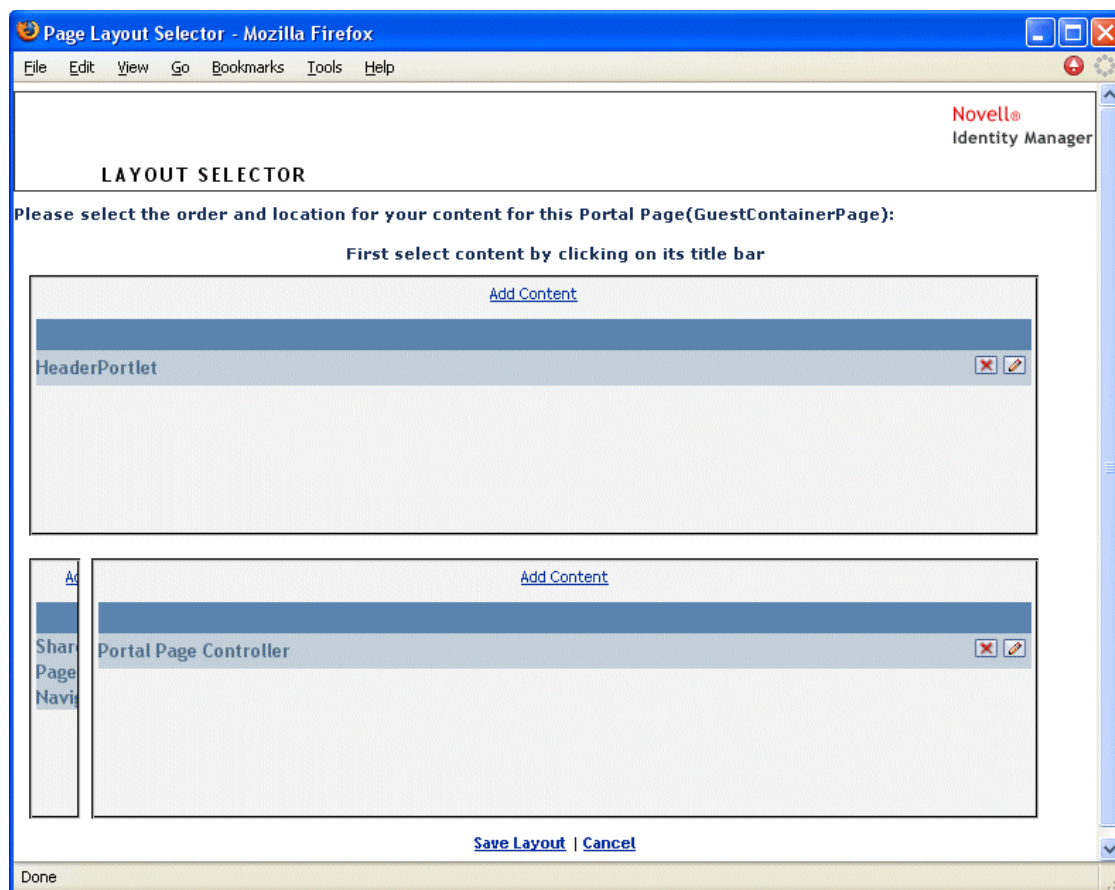
By default, when users arrive at the Identity Manager user interface prior to logging in, they see the container page named GuestContainerPage shown in [Figure 6-1](#).

Figure 6-1 Default Guest Container Page



Internally, GuestContainerPage has the following layout:

Figure 6-2 *GuestContainerPage Layout*



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

Table 6-2 *Layout Regions*

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

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.

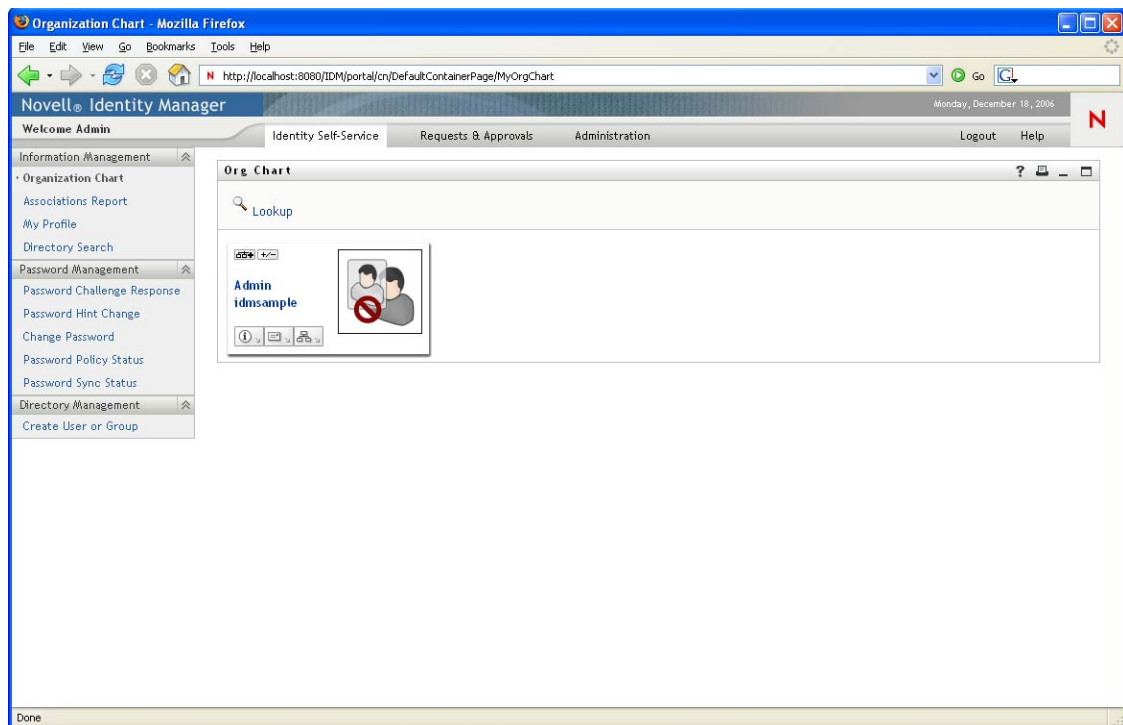
After login, the Shared Page Navigation portlet filters out the Guest Pages category. Instead, it shows other categories of shared pages (as specified in its preferences).

For more information on the Shared Page Navigation portlet, see [Chapter 10, “About Portlets,”](#) on [page 207](#).

DefaultContainerPage

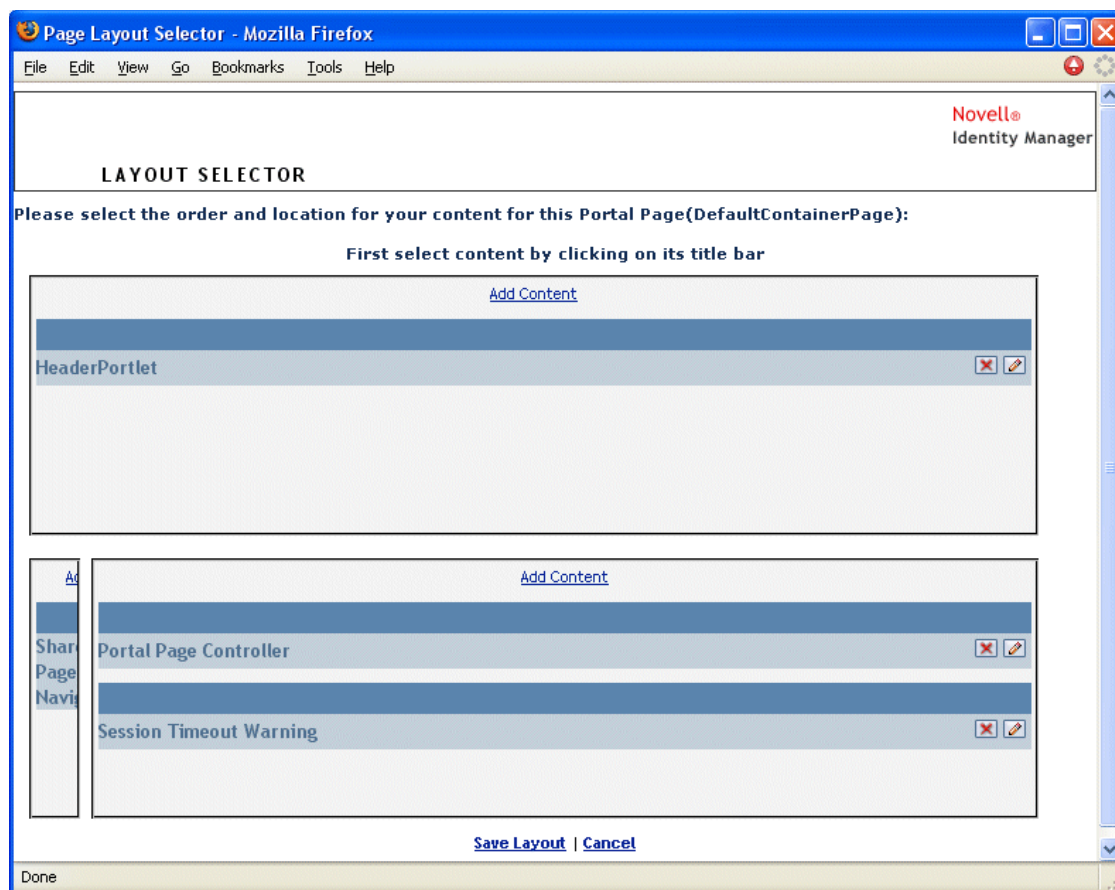
By default, after users log in to the Identity Manager user interface, they go to the container page named DefaultContainerPage shown in [Figure 6-3](#).

Figure 6-3 *Default Container Page*



Internally, DefaultContainerPage has the layout shown in [Figure 6-4](#).

Figure 6-4 Default Container Page Layout



The DefaultContainerPage layout is divided into three regions, which display the portlets described in [Table 6-3](#).

Table 6-3 Default Container Page Portlets

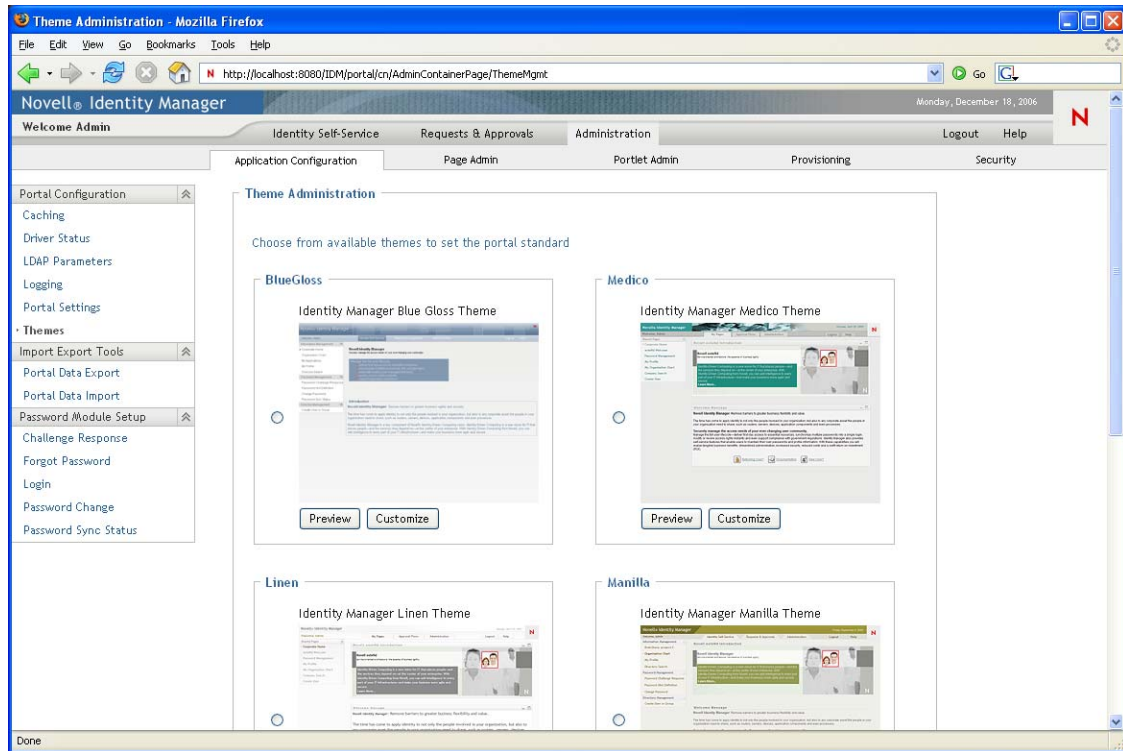
Portlet	Description
HeaderPortlet	Displays the header information and top-level tab controls for the user interface
Shared Page Navigation	Displays a vertical menu from which the user can select a shared page to display
Portal Page Controller	Displays the shared page that the user has currently selected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

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

Admin Container Page

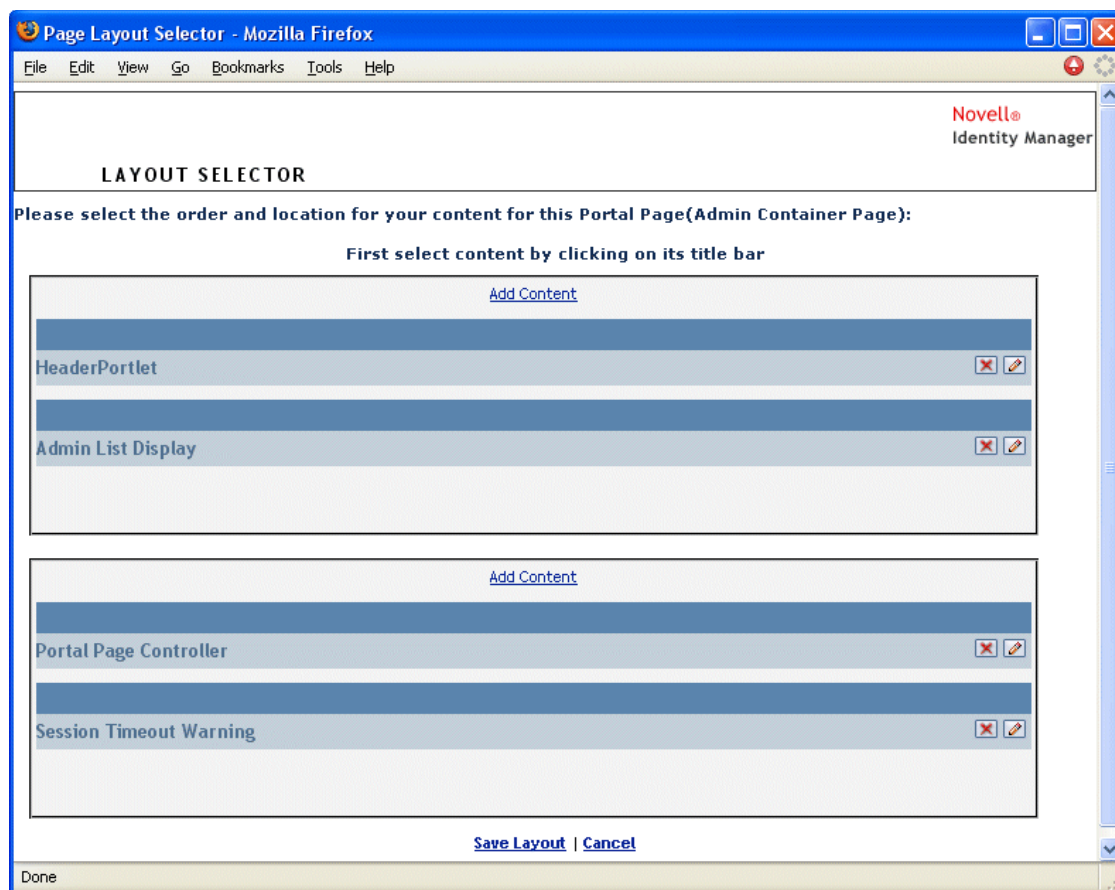
By default, when User Application Administrators (and other authorized users) click the *Administration* tab of the Identity Manager user interface, they go to the container page named Admin Container Page, which displays as shown in [Figure 6-5](#).

Figure 6-5 Default Admin Container Page



Internally, Admin Container Page has the layout shown in [Figure 6-6](#).

Figure 6-6 Admin Container Page Layout



The Admin Container Page layout is divided into two regions, which display the portlets described in [Table 6-4](#).

Table 6-4 Default Admin Container Page 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

6.1.2 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 6.3, “Creating and Maintaining Shared Pages,”](#) on page 155.

A Typical Shared Page

As an example of one of these shared pages, Organization Chart is the default shared page that DefaultContainerPage displays after users log in to the Identity Manager user interface. It is shown in [Figure 6-7](#).

Figure 6-7 Sample Shared Page



Internally, Organization Chart has the layout shown in [Figure 6-8](#).

Figure 6-8 Default Org Chart Layout



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

6.1.3 An Exception to Page Usage

In this section, you have 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

However, the *Requests & Approvals* tab is based on a different architecture and cannot be manipulated through Page Admin.

6.2 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 6.2.1, "Creating Container Pages,"](#) on page 148.

- 2 Add content (in the form of portlets) to the page, as described in [Section 6.2.2, “Adding Content to a Container Page,” on page 150](#).
You can also delete content from the page, as described in [Section 6.2.3, “Deleting Content from a Container Page,” on page 152](#).
- 3 Choose a portal layout, as described in [Section 6.2.4, “Modifying the Layout of a Container Page,” on page 153](#).
- 4 Arrange the order and position of content on the selected layout, as described in [Section 6.2.5, “Arranging Content on the Container Page,” on page 153](#).
- 5 Immediately display the new page by specifying the container page URL in your browser, as described in [Section 6.2.6, “Displaying a Container Page,” on page 155](#).

You can switch layouts for container pages without losing page contents. When you apply a new layout to a container page, portlets in the page are automatically displayed using the new layout. You might need to fine-tune the content placement in the new layout.

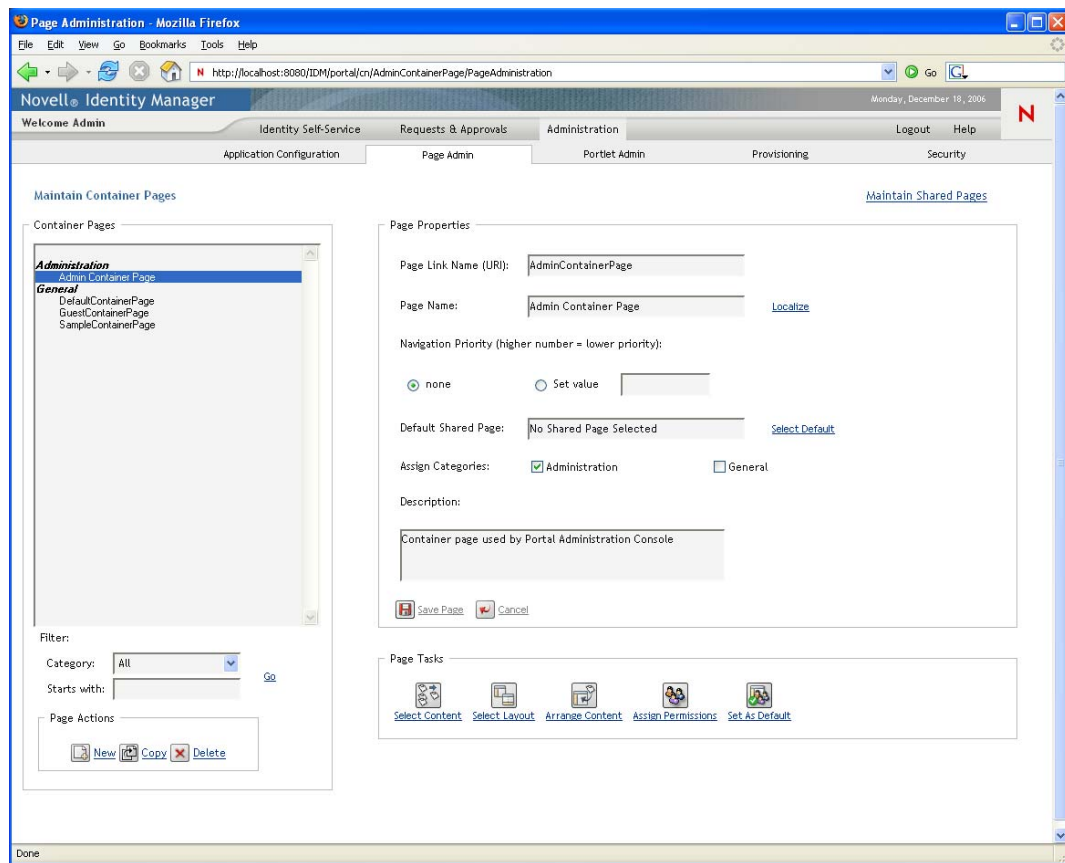
6.2.1 Creating Container Pages

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)	<p>Specify the URI name for the page (as it is to appear within the user interface URL). For example, if you specify the URI:</p> <p><code>MyContainerPage</code></p> <p>it appears within the URL like this:</p> <p><code>http://myappserver:8080/IDM/portal/cn/MyContainerPage</code></p>
Page Name	<p>Specify the display name for the page. For example:</p> <p><code>My Container Page</code></p> <p>Click <i>Localize</i> to specify localized versions of this name for other languages.</p>

Property	What to do
Navigation Priority	<p>Specify one of the following:</p> <ul style="list-style-type: none"> ♦ <i>None</i> if you don't need to assign a priority to this container page. ♦ <i>Set value</i> to assign a priority to this container page, relative to other container pages. The priority must be an integer between 0 and 9999, where 0 is the highest priority and 9999 is the lowest. <p>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).</p>
Default Shared Page	See Section 6.6, "Selecting a Default Shared Page for a Container Page," on page 170.
Assign Categories	<p>Select zero or more of the following categories in which you want the page to belong:</p> <ul style="list-style-type: none"> ♦ Administration ♦ General <p>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.</p>
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.

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).

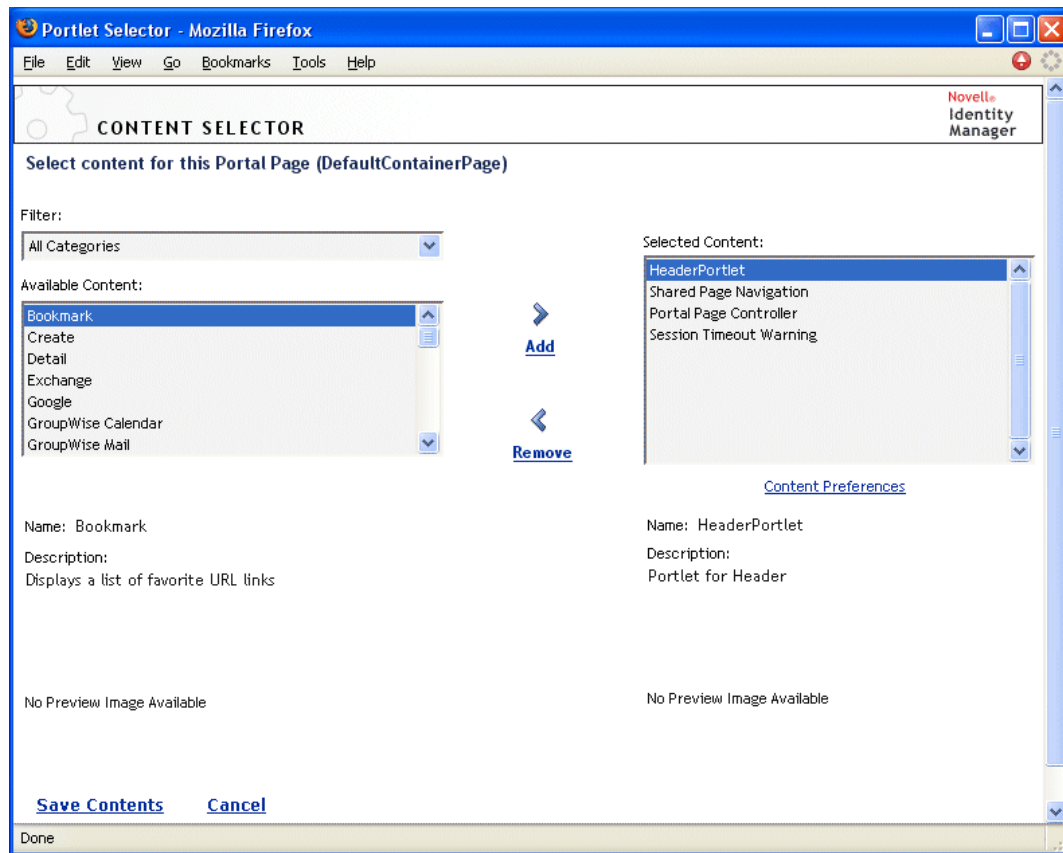
6.2.2 Adding Content to a Container Page

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* list.
- 3 Select one or more portlets from the *Available Content* list.
Hold down Control to select multiple non-contiguous portlets from the list; use Shift to make multiple contiguous selections.
- 4 Click *Add* to move your choices to the *Selected Content* list.
- 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 6.2.4, “Modifying the Layout of a Container Page,” on page 153](#), or arrange the content on the current layout as described in [Section 6.2.5, “Arranging Content on the Container Page,” on page 153](#).

6.2.3 Deleting Content from a Container Page

In the process of creating container pages, you might 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 [Step 1 on page 151](#).

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

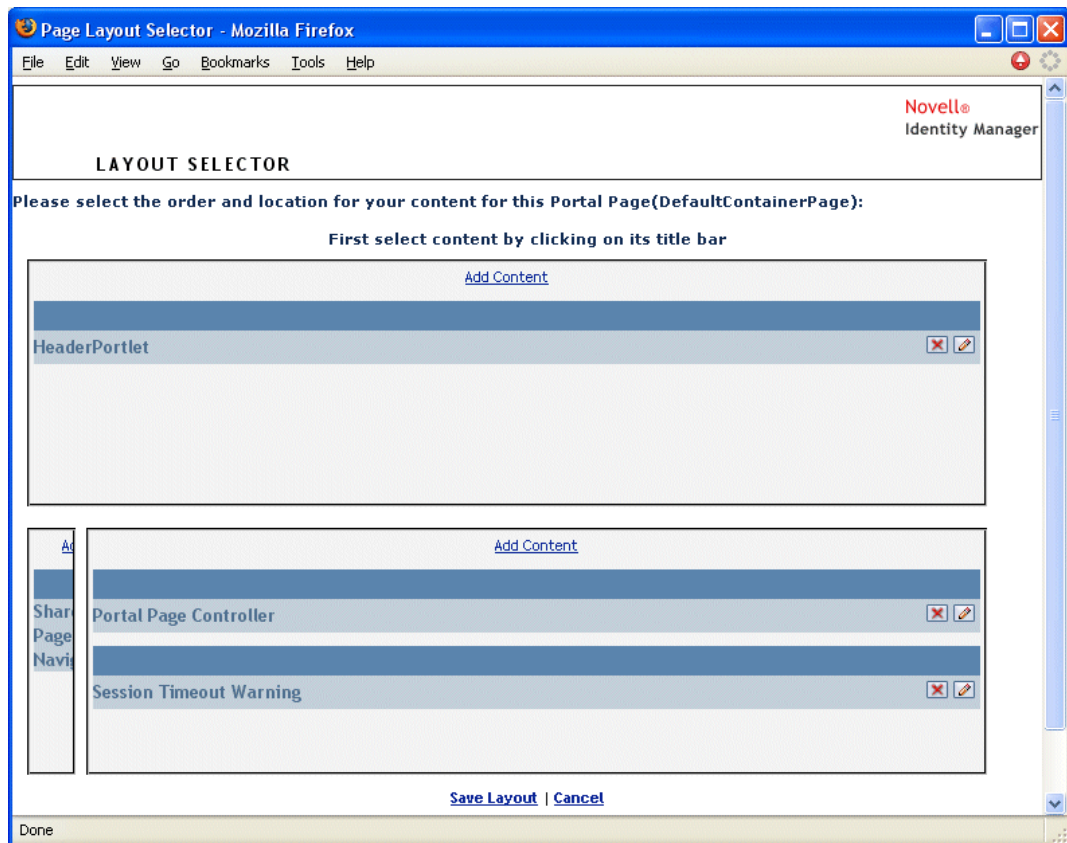
The portlet is removed from the page.

- 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*.

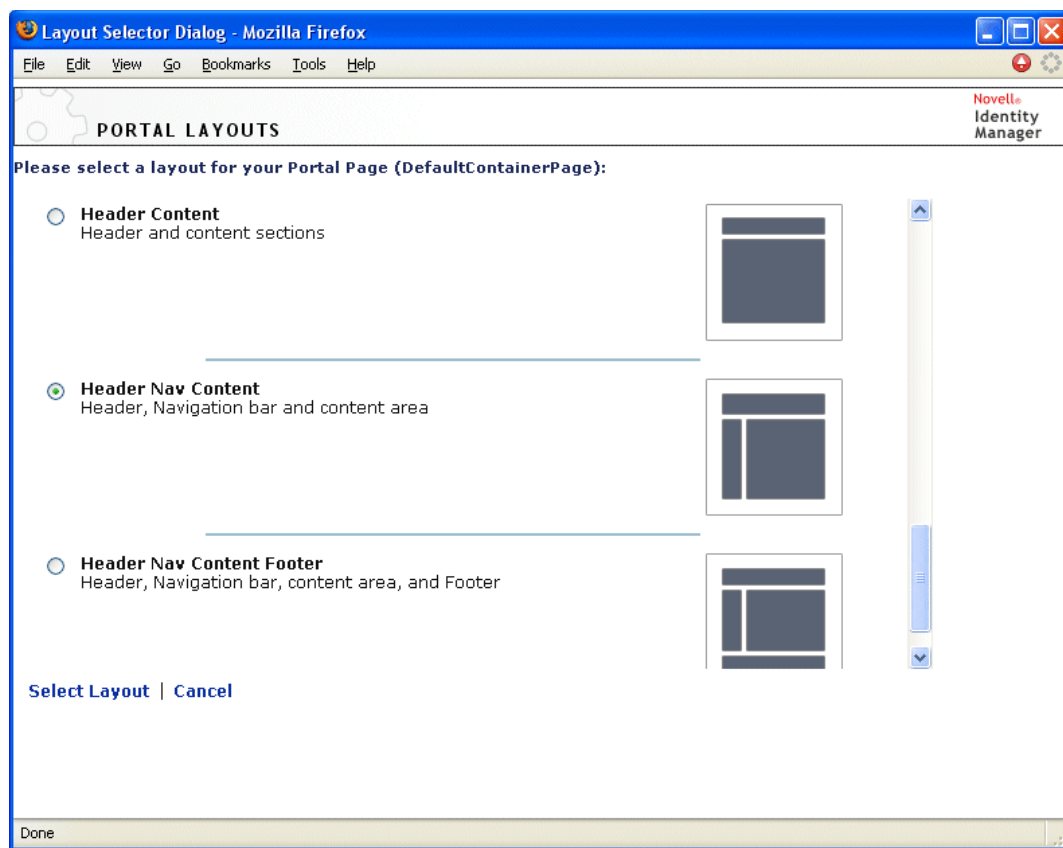
6.2.4 Modifying the Layout of a Container Page

When you modify the layout of a container page, existing content is shifted to accommodate the new layout. In some cases, you might 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*.

6.2.5 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.

- 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 To add a portlet to the page:
 - 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* drop-down list.
 - 2c Select a portlet you want from the *Available Content* list.
 - 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	<ol style="list-style-type: none"> 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.

Browser	What to do
Mozilla	<ol style="list-style-type: none"> 1. Click the portlet you want to move. 2. Click inside the destination layout frame. <p>The portlet moves to the destination.</p>

- 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 To edit the preferences of a portlet:
 - 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.

6.2.6 Displaying a Container Page

You can display your page by going to the container page URL in your browser. Specify the following URL in your web browser:

```
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
```

6.3 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 6.3.1, "Creating Shared Pages," on page 156](#).
- 2 Add content (in the form of portlets) to the page, as described in [Section 6.3.2, "Adding Content to a Shared Page," on page 158](#).
You might also want to delete content from the page, as described in [Section 6.3.3, "Deleting Content from a Shared Page," on page 160](#).
- 3 Choose a portal layout, as described in [Section 6.3.4, "Modifying the Layout of a Shared Page," on page 161](#).
- 4 Arrange the order and position of content on the selected layout, as described in [Section 6.3.5, "Arranging Content on the Shared Page," on page 161](#).
- 5 Display the new page by entering the shared page URL in your browser, as described in [Section 6.3.6, "Displaying a Shared Page," on page 163](#).

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 might need to fine-tune the content placement in the new layout.

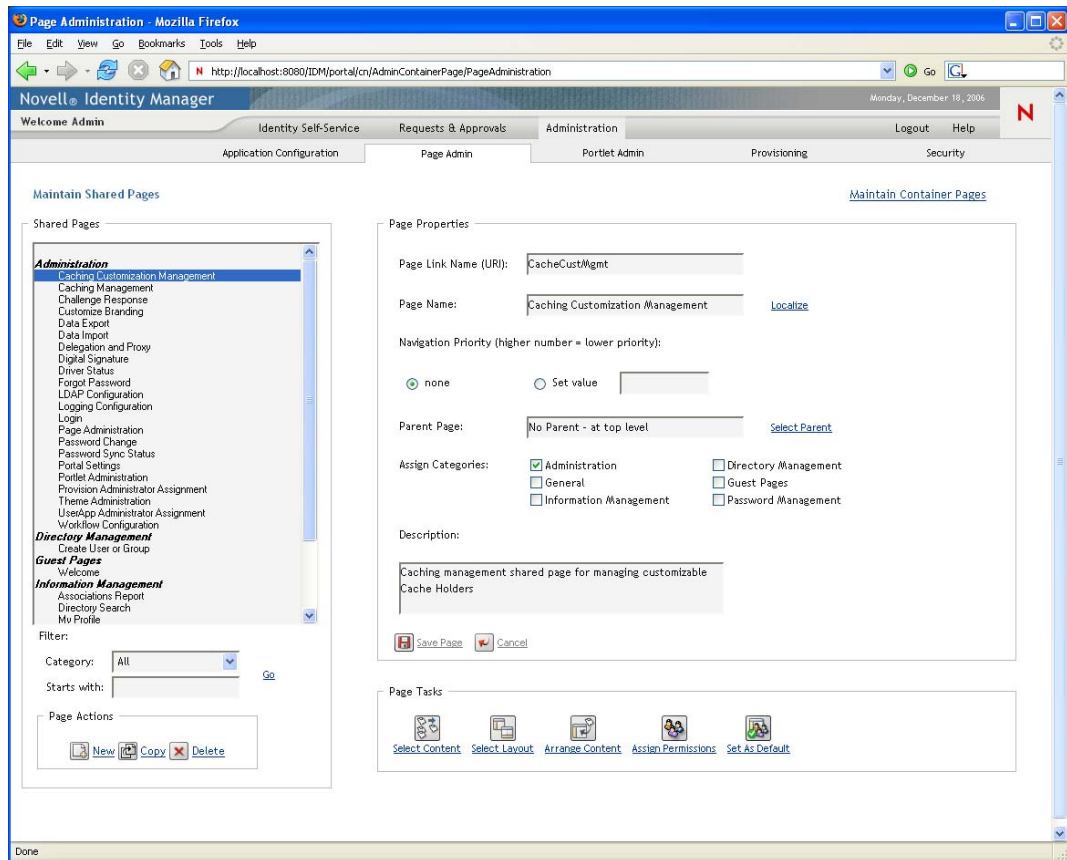
6.3.1 Creating Shared Pages

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)	<p>Specify the URI name for the page (as it is to appear within the user interface URL). For example, if you specify the URI:</p> <p><code>MySharedPage</code></p> <p>it appears within the URL like this:</p> <p><code>http://myappserver:8080/IDM/portal/cn/MyContainerPage/MySharedPage</code></p>
Page Name	<p>Specify the display name for the page. For example:</p> <p><code>My Shared Page</code></p> <p>You can click <i>Localize</i> to specify localized versions of this name for other languages.</p>
Navigation Priority	<p>Specify one of the following:</p> <ul style="list-style-type: none"> ♦ <i>None</i> if you don't need to assign a priority to this shared page. ♦ <i>Set value</i> to assign a priority to this shared page, relative to other shared pages. The priority must be an integer between 0 and 9999, where 0 is the highest priority and 9999 is the lowest. <p>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).</p>
Parent Page	<p>If you want this shared page to be the child of another shared page, click <i>Select Parent</i>. Make sure that both the parent and child pages belong to the <i>same categories</i> (to prevent display problems).</p> <p>At runtime, the end user sees this relationship when using the Shared Page Navigation portlet. When displaying the list of shared pages, it shows children indented under their parents.</p> <p>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.</p>

Property	What to do
Assign Categories	<p>Select zero or more of the following categories in which you want the page to belong:</p> <ul style="list-style-type: none"> ♦ Administration ♦ Directory Management ♦ General ♦ Guest Pages ♦ Information Management ♦ Password Management <p>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.</p> <hr/> <p>NOTE: <i>Guest Pages</i> is a special category used to identify shared pages that can be displayed prior to user login but not after. For more information, see the section on the Shared Page Navigation portlet in Chapter 10, "About Portlets," on page 207.</p>
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 ["To create a shared page from scratch:" on page 156](#).

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

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 ["To create a shared page from scratch:" on page 156](#).

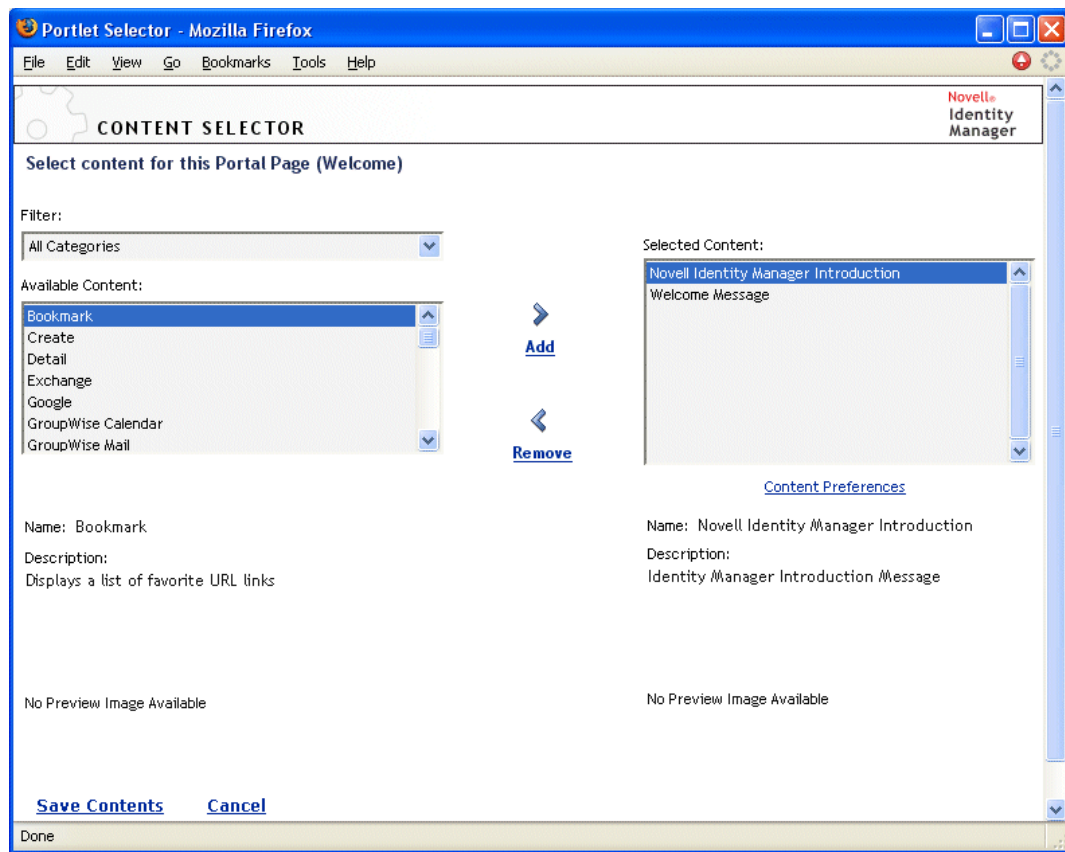
- 5 Click *Save Page* (at the bottom of the page properties section).

6.3.2 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.

- 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* drop-down list.
- 3 Select one or more portlets from the *Available Content* list.
Hold down the Ctrl 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 *Selected Content* list.
- 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 6.3.4, “Modifying the Layout of a Shared Page,” on page 161](#), or arrange the content on the current layout as described in [Section 6.3.5, “Arranging Content on the Shared Page,” on page 161](#).

6.3.3 Deleting Content from a Shared Page

In the process of creating shared pages, you might 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.

- 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 [Section 6.3.2, “Adding Content to a Shared Page,” on page 158](#).

- 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 by 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*.

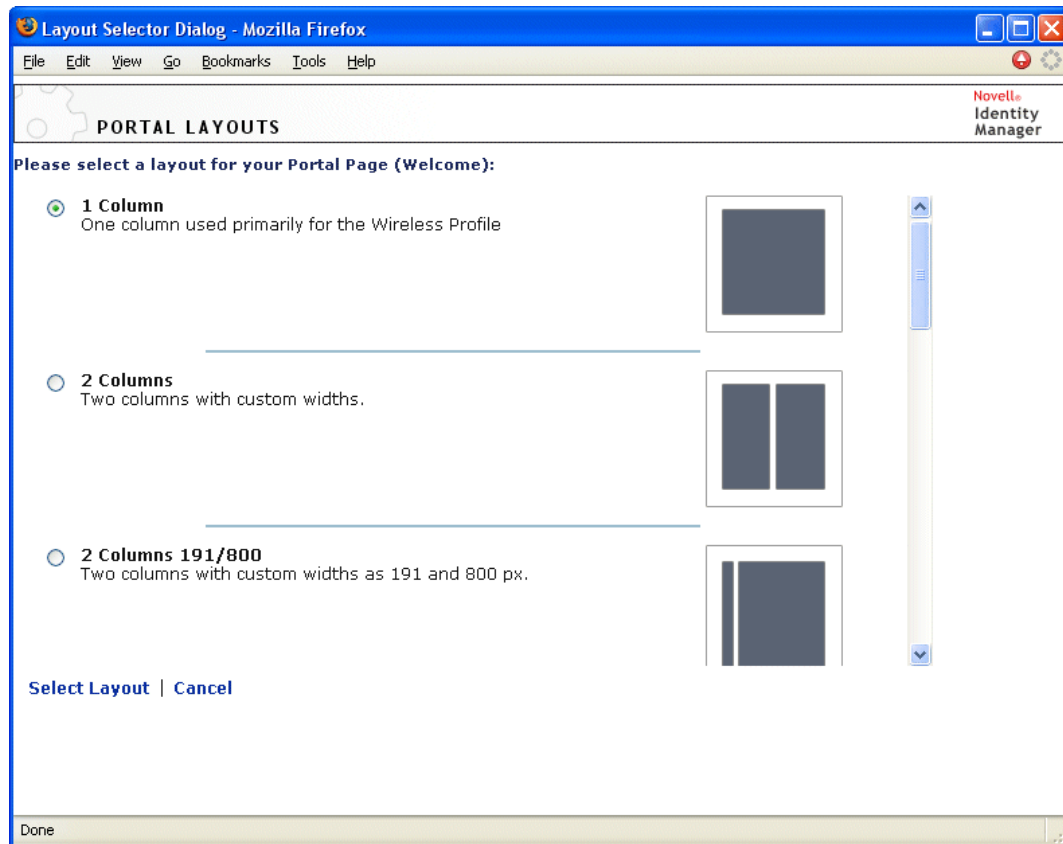
6.3.4 Modifying the Layout of a Shared Page

When you modify the layout of a shared page, existing content is shifted to accommodate the new layout. In some cases, you might 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*.

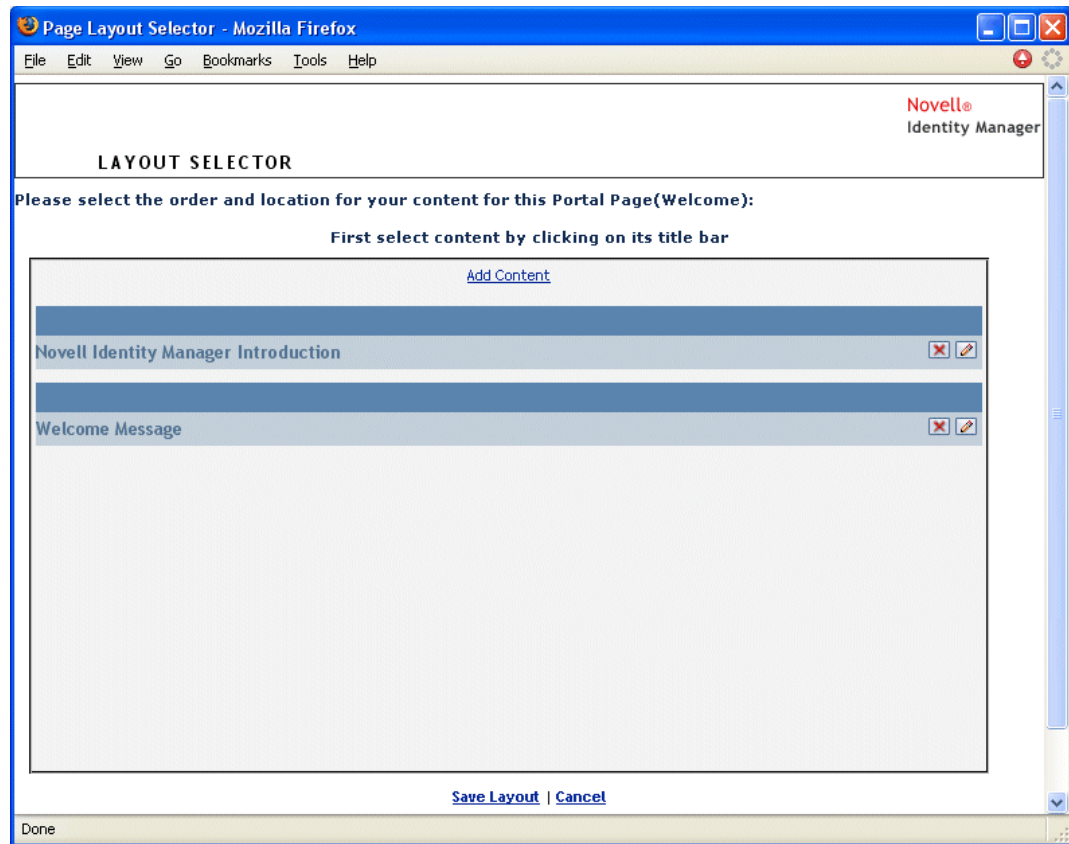
6.3.5 Arranging Content on the Shared Page

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:
 - 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* drop-down list.
 - 2c Select a portlet you want from the *Available Content* list.
 - 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	<ol style="list-style-type: none"> 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.

Browser	What to do
Mozilla Firefox	<ol style="list-style-type: none"> 1. Click the portlet you want to move. 2. Click inside the destination layout frame. <p>The portlet moves to the destination.</p>

4 If you want to remove a portlet from the layout:

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:

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.

6.3.6 Displaying a Shared Page

To display your shared page, go to this URL in your Web browser:

`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`

6.4 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.

Table 6-5 *Page Permissions*

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

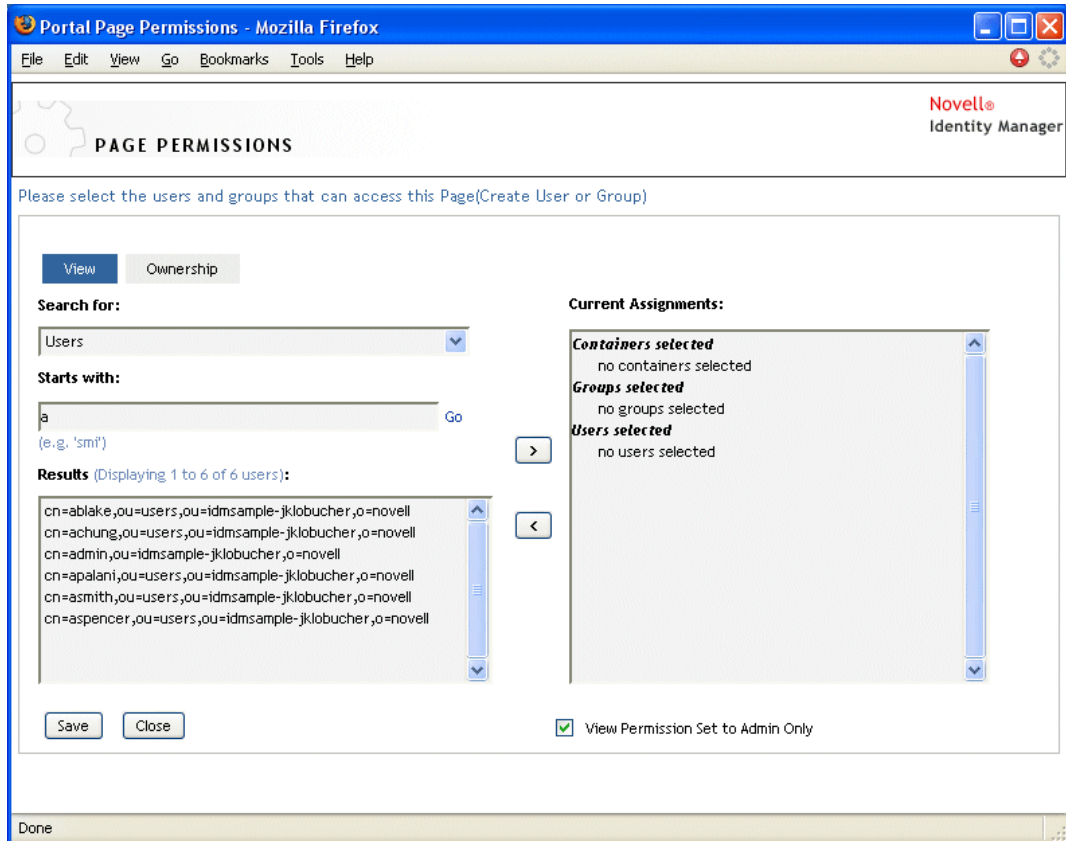
6.4.1 Assigning Page View Permission

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 box displays in a new browser window:



- 2 Go to the *View* tab.
- 3 Specify values for the following search settings:

Setting	What to do
<i>Search for</i>	Select one of the following from the drop-down menu: <ul style="list-style-type: none">◆ Users◆ Groups◆ Containers

Setting	What to do
<i>Starts with</i>	<p>If you want to:</p> <ul style="list-style-type: none"> ♦ 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 characters of the CN values you want. (Case is not considered. Wildcards are not supported.) <p>For example, searching for groups that start with <i>s</i> would narrow your search results to something like this:</p> <pre>cn=Sales,ou=groups,o=MyOrg cn=Service,ou=groups,o=MyOrg cn=Shipping,ou=groups,o=MyOrg</pre> <p>Searching for groups that start with <i>Se</i> would return:</p> <pre>cn=Service,ou=groups,o=MyOrg</pre>

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.

Hold down the Ctrl 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	Select <i>View Permission Set to Admin Only</i>
Allow all assigned users, groups, and containers to view the page	Deselect <i>View Permission Set to Admin Only</i> NOTE: If you deselect this setting but there are no users, groups, or containers explicitly assigned to the page, then everyone has View permission for this page.

7 Click *Save*, then click *Close*.

6.4.2 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 box displays in a new browser window as shown in [Step 1 on page 164](#).

2 Go to the *Ownership* tab.

3 Specify values for the following search settings:

Setting	What to do
<i>Search for</i>	Select one of the following from the drop-down menu: <ul style="list-style-type: none">♦ Users♦ Groups♦ Containers
<i>Starts with</i>	If you want to: <ul style="list-style-type: none">♦ 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 characters of the CN values you want. (Case is not considered. Wildcards are not supported.) <p>For example, searching for groups that start with <i>s</i> would narrow your search results to something like this:</p> <pre>cn=Sales,ou=groups,o=MyOrg cn=Service,ou=groups,o=MyOrg cn=Shipping,ou=groups,o=MyOrg</pre> <p>Searching for groups that start with <i>Se</i> would return:</p> <pre>cn=Service,ou=groups,o=MyOrg</pre>

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.

Hold down the Ctrl 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	Select <i>Ownership Permission Set to Admin Only</i>
Allow all assigned users, groups, and containers to work with the page	Deselect <i>Ownership Permission Set to Admin Only</i>
	NOTE: If you deselect this setting but there are no users, groups, or containers explicitly assigned to the page, then everyone has Ownership permission for this page.

7 Click *Save*, then click *Close*.

6.4.3 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. For instance, selected people in administration or management positions might need the ability to create users, groups, or task groups.

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 CreatePortlet portlet registration (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 7, “Portlet Administration,” on page 173](#).

- 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 containers 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]	<ul style="list-style-type: none">♦ Compare♦ Read♦ Write	Yes (select this check box)
[Entry Rights]	<ul style="list-style-type: none">♦ 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 might 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*.

6.4.4 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 (Application Configuration, Page Admin, Portlet Admin, Provisioning, Security). 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. For example, a small group of users might 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 7, "Portlet Administration," on page 173](#).

6.5 Setting Default Pages for Groups

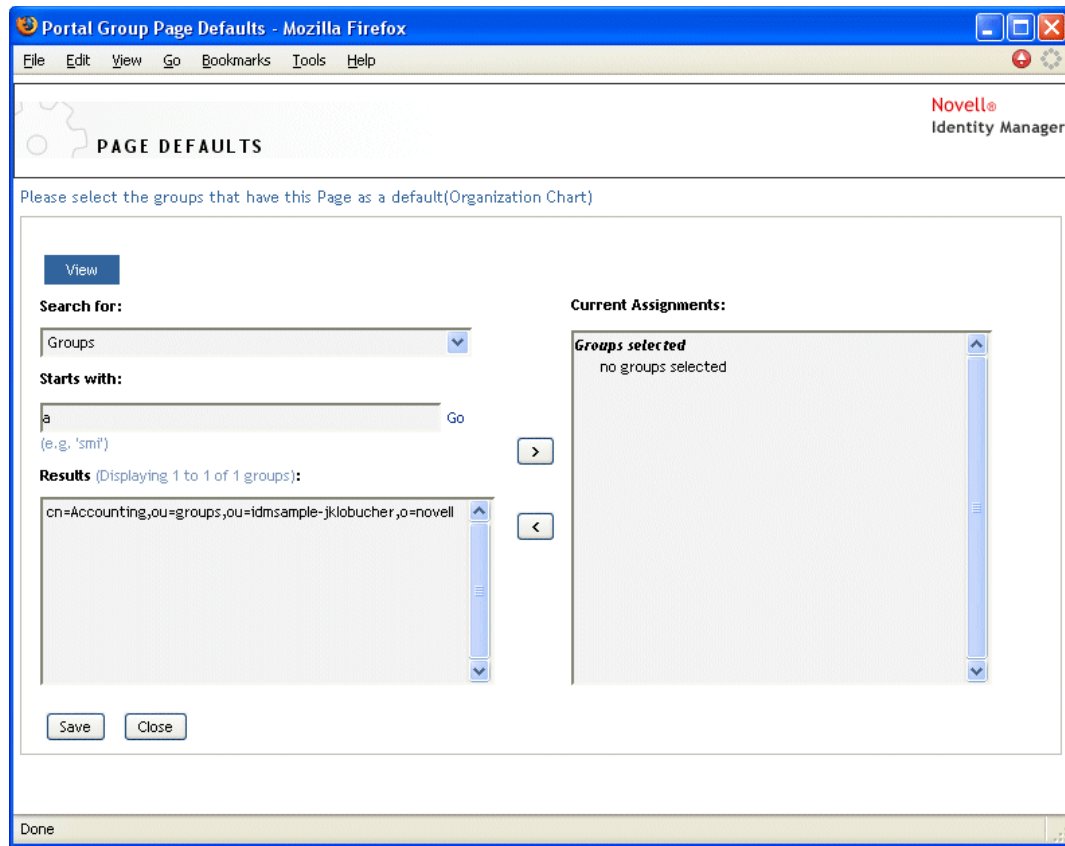
You can assign a default container page and a default shared page for any authorized group of users. These settings affect the container page those users see when they log in and the shared page they see on the container page.

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 box displays in a new browser window:



2 Specify values for the following search settings:

Setting	What to do
<i>Search for</i>	Groups is automatically selected.
<i>Starts with</i>	<p>If you want to:</p> <ul style="list-style-type: none"> Find all available groups, then make this setting blank. Find a subset of those groups, then enter the starting characters of the CN values you want. (Case is not considered. Wildcards are not supported.) <p>For example, searching for groups that start with s would narrow your search results to something like this:</p> <pre>cn=Sales,ou=groups,o=MyOrg cn=Service,ou=groups,o=MyOrg cn=Shipping,ou=groups,o=MyOrg</pre> <p>Searching for groups that start with Se would return:</p> <pre>cn=Service,ou=groups,o=MyOrg</pre>

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.

Hold down the Ctrl key to make multiple selections.

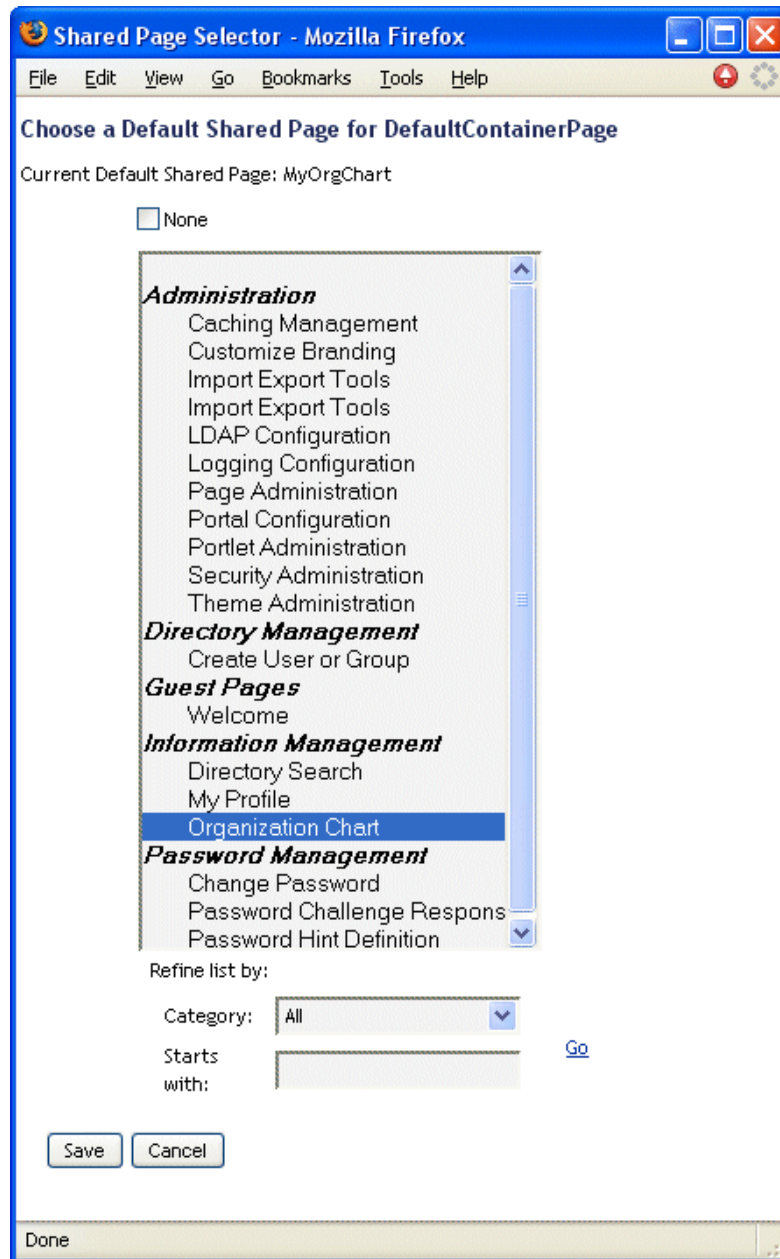
- 5 Click *Save*, then click *Close*.

6.6 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.

- 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 box 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 select *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).

Portlet Administration

7

This section describes how to use the Portlet Admin page on the *Administration* tab of the Identity Manager user interface. Topics include:

- ♦ [Section 7.1, “About Portlet Administration,” on page 173](#)
- ♦ [Section 7.2, “Administering Portlet Definitions,” on page 173](#)
- ♦ [Section 7.3, “Administering Registered Portlets,” on page 177](#)

For more general information about accessing and working with the *Administration* tab, see [Chapter 4, “Using the Administration Tab,” on page 83](#).

7.1 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. [Table 7-1](#) describes how to manage portlets.

Table 7-1 *Managing Portlets*

What you work with	Description
Portlet definitions	Descriptors (read from <code>portlet.xml</code>) that specify portlet configuration parameters. There is one definition for each portlet in an application. See Section 7.2, “Administering Portlet Definitions,” on page 173 .
Portlet registrations	Registrations of portlets, based on their definitions. Multiple registrations of the same portlet can exist in a single portlet application. See Section 7.3, “Administering Registered Portlets,” on page 177 .

For details on the portlets provided with the Identity Manager user interface, see [Part IV, “Portlet Reference,” on page 205](#). To learn about using portlets on container pages and shared pages, see [Chapter 6, “Page Administration,” on page 139](#).

7.2 Administering Portlet Definitions

The Portlet Admin page enables you to perform the following tasks related to portlet definitions in a portlet application:

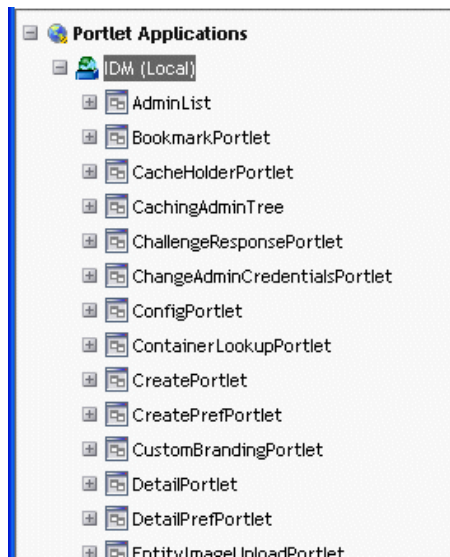
- ♦ [Section 7.2.1, “Accessing Portlet Definitions in the Deployed Portlet Application,” on page 174](#)
- ♦ [Section 7.2.2, “Registering Portlet Definitions,” on page 174](#)
- ♦ [Section 7.2.3, “Viewing Information About Portlet Definitions,” on page 175](#)

7.2.1 Accessing Portlet Definitions in the Deployed Portlet Application

The *Portlet Applications* list shows the portlet definitions in a selected 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:



7.2.2 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 7.3, “Administering Registered Portlets,” on page 177](#)
- ♦ When adding an instance of the portlet to a page. See [Chapter 6, “Page Administration,” on page 139](#)

All portlets that ship with the Identity Manager User Application are automatically registered.

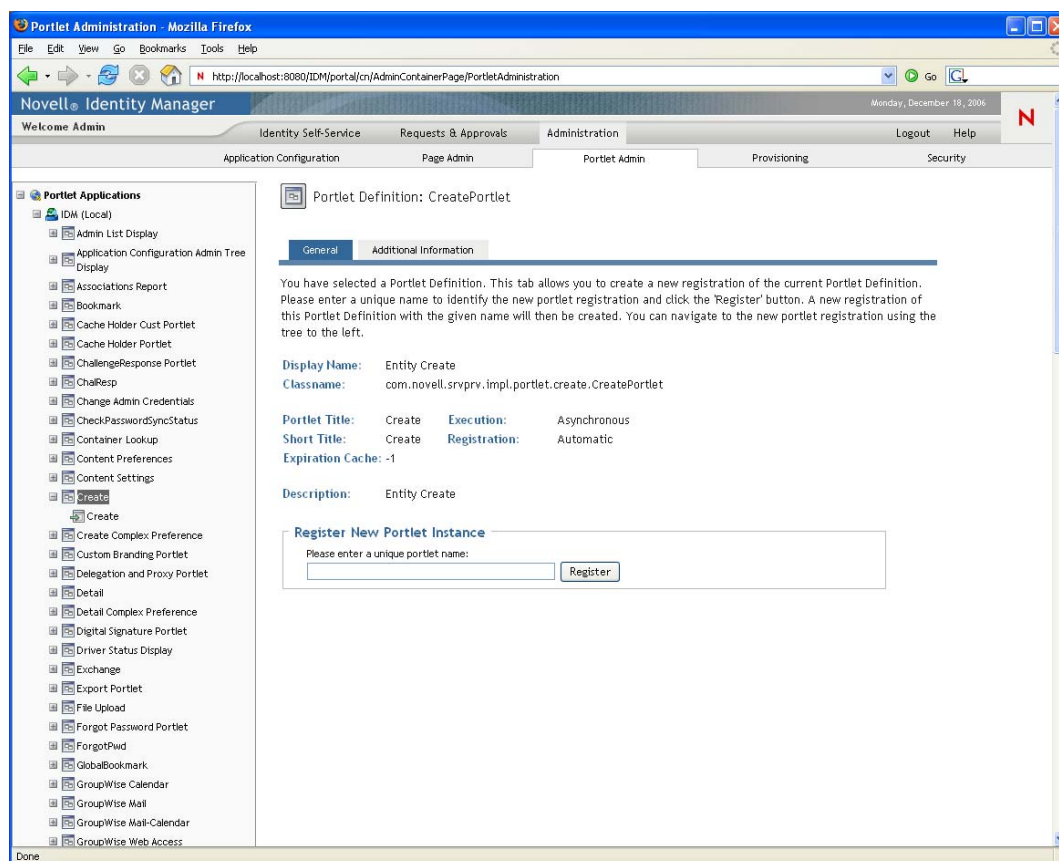
If the portlet definition provides an Edit mode, the end user can modify specific preferences of the portlet registration at runtime, according to the logic of the portlet’s `doEdit()` method.

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:



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, specify 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 7.3, “Administering Registered Portlets,” on page 177](#).

7.2.3 Viewing Information About Portlet Definitions

You can view the following read-only information about a listed portlet definition:

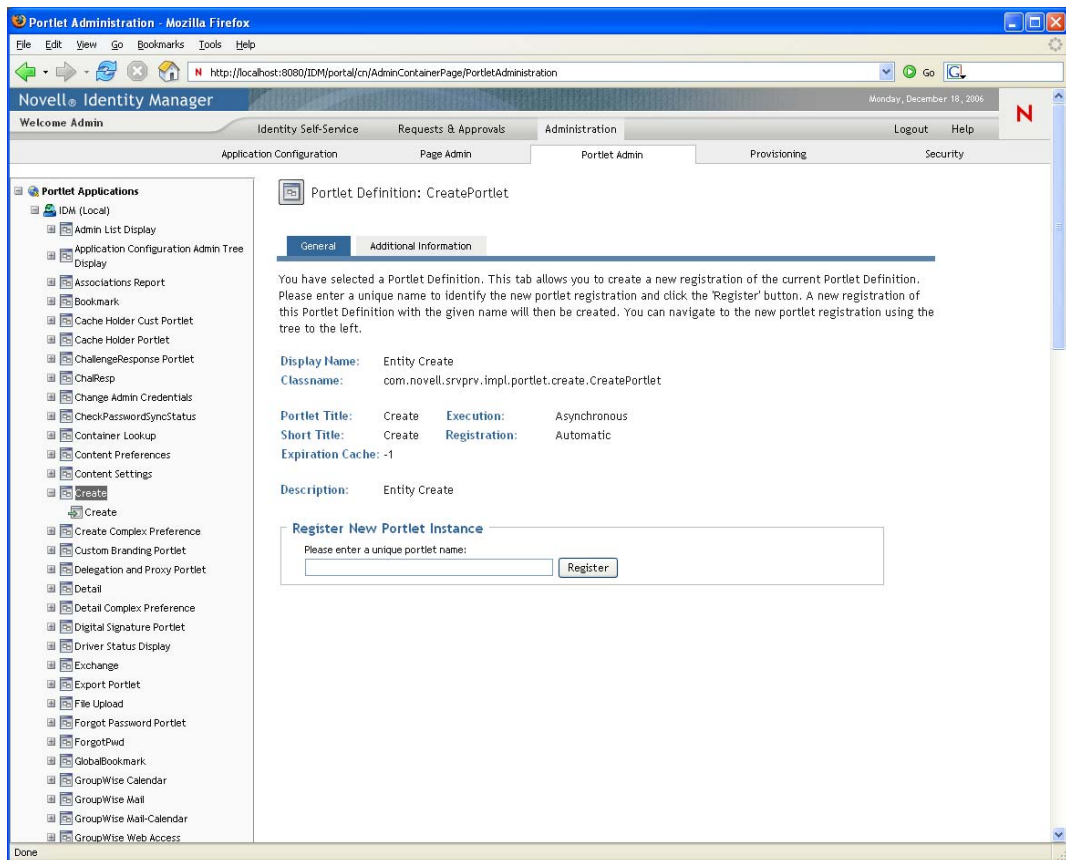
- ♦ Display name
- ♦ Class name
- ♦ Portlet title
- ♦ Type of execution (synchronous or asynchronous)
- ♦ Short title

- ◆ 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

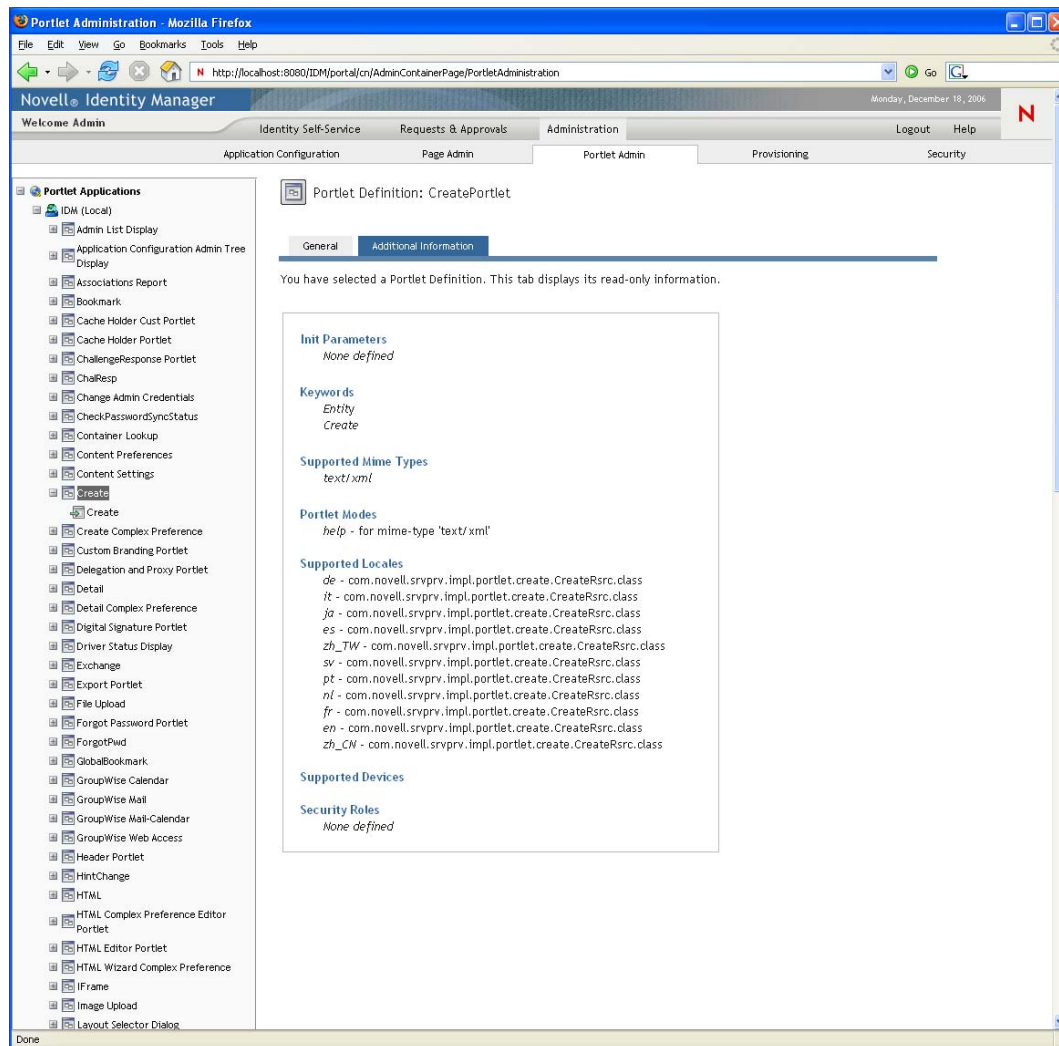
To view information about portlet definitions:

- 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:



7.3 Administering Registered Portlets

The Portlet Admin page enables you to perform the following tasks related to portlet registrations in a portlet application:

- ◆ Section 7.3.1, “Accessing Portlet Registrations in the Deployed Portlet Application,” on page 178
- ◆ Section 7.3.2, “Viewing Information about Portlet Registrations,” on page 179
- ◆ Section 7.3.3, “Assigning Categories to Portlet Registrations,” on page 180
- ◆ Section 7.3.4, “Modifying Settings for Portlet Registrations,” on page 181
- ◆ Section 7.3.5, “Modifying Preferences for Portlet Registrations,” on page 183
- ◆ Section 7.3.6, “Assigning Security Permissions for Portlet Registrations,” on page 184
- ◆ Section 7.3.7, “Unregistering a Portlet,” on page 186

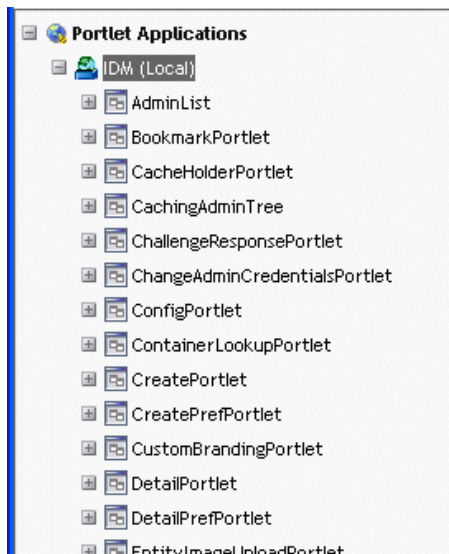
7.3.1 Accessing Portlet Registrations in the Deployed Portlet Application

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:

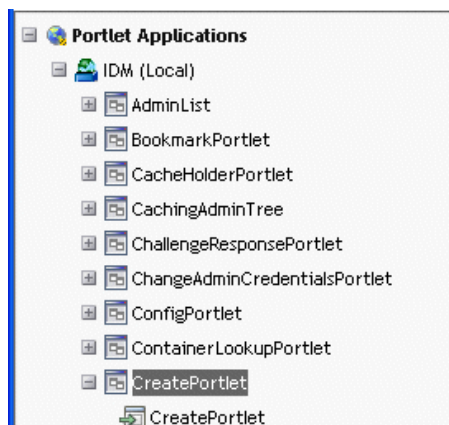
- 1 In the Portlet Applications list, expand the portlet application whose portlet definitions and registrations you want to access.

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:



7.3.2 Viewing Information about Portlet Registrations

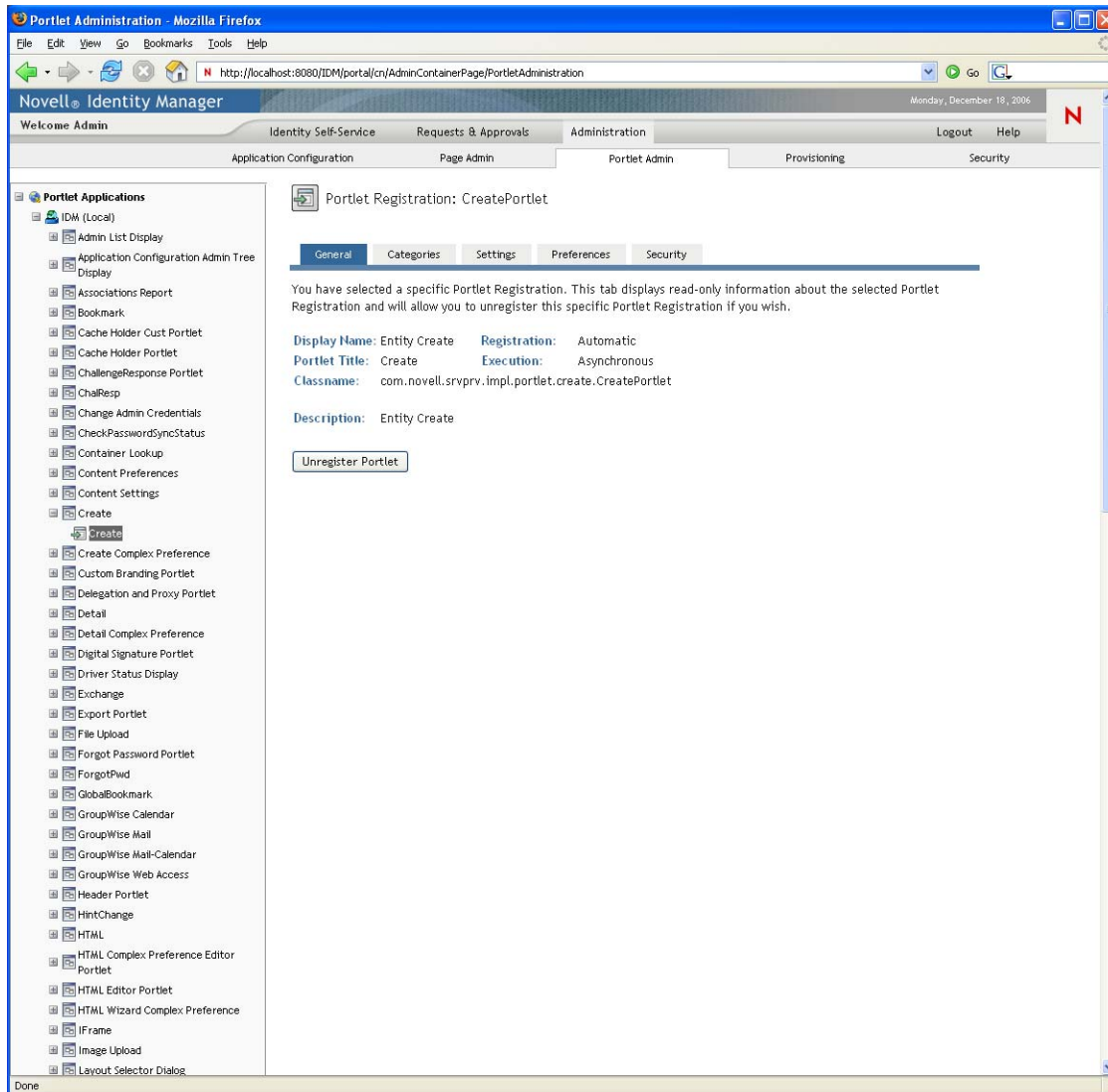
You can view the following read-only information about a listed portlet registration:

- ♦ Display name
- ♦ Type of registration
- ♦ Portlet title
- ♦ Type of execution (synchronous or asynchronous)
- ♦ Class name
- ♦ Description

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 as shown in **Figure 7-1**.

Figure 7-1 Portlet Registration: General Properties



7.3.3 Assigning Categories to Portlet Registrations

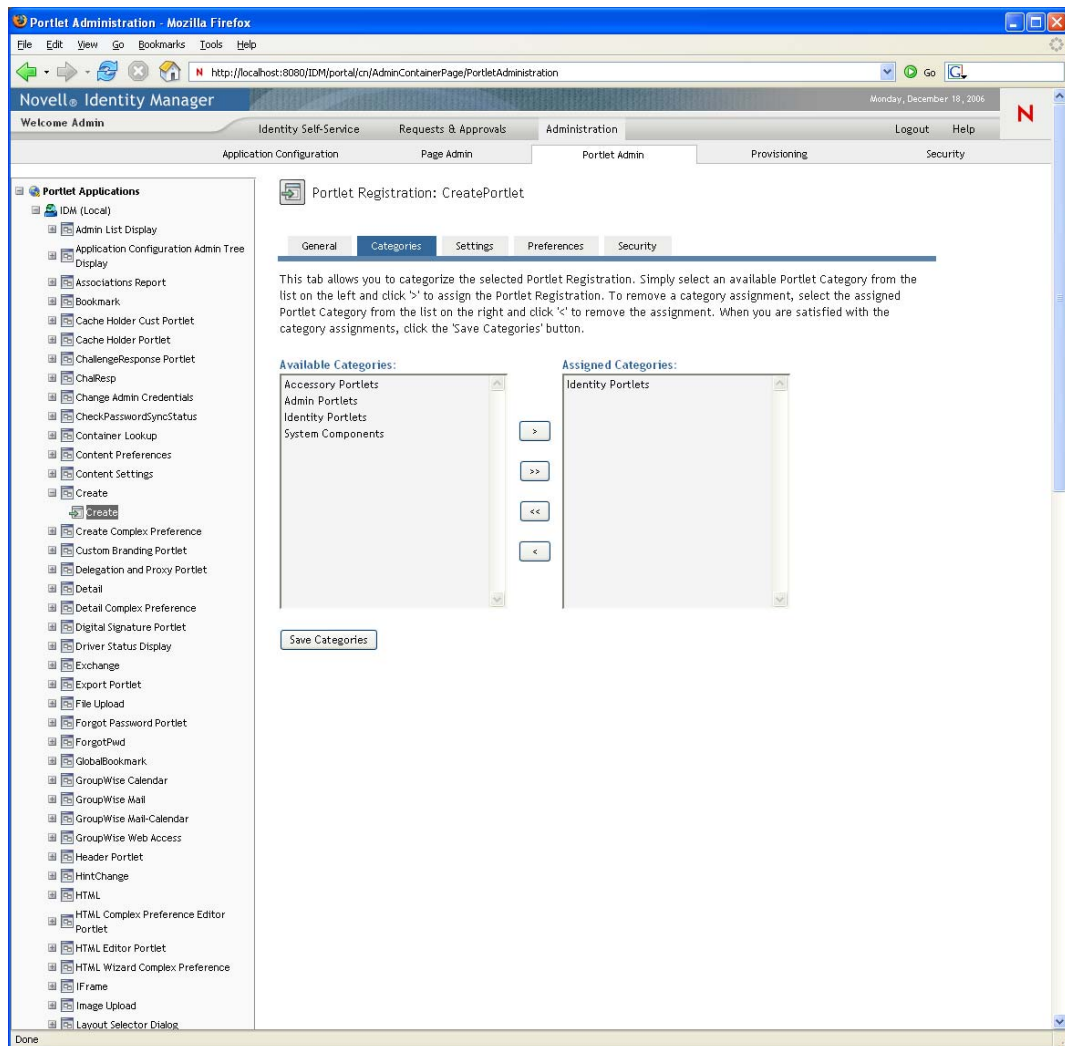
To facilitate searching for specific portlets in a portlet application, you can organize portlet registrations by category.

- 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 >

If you want to	Do this
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*.

7.3.4 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.

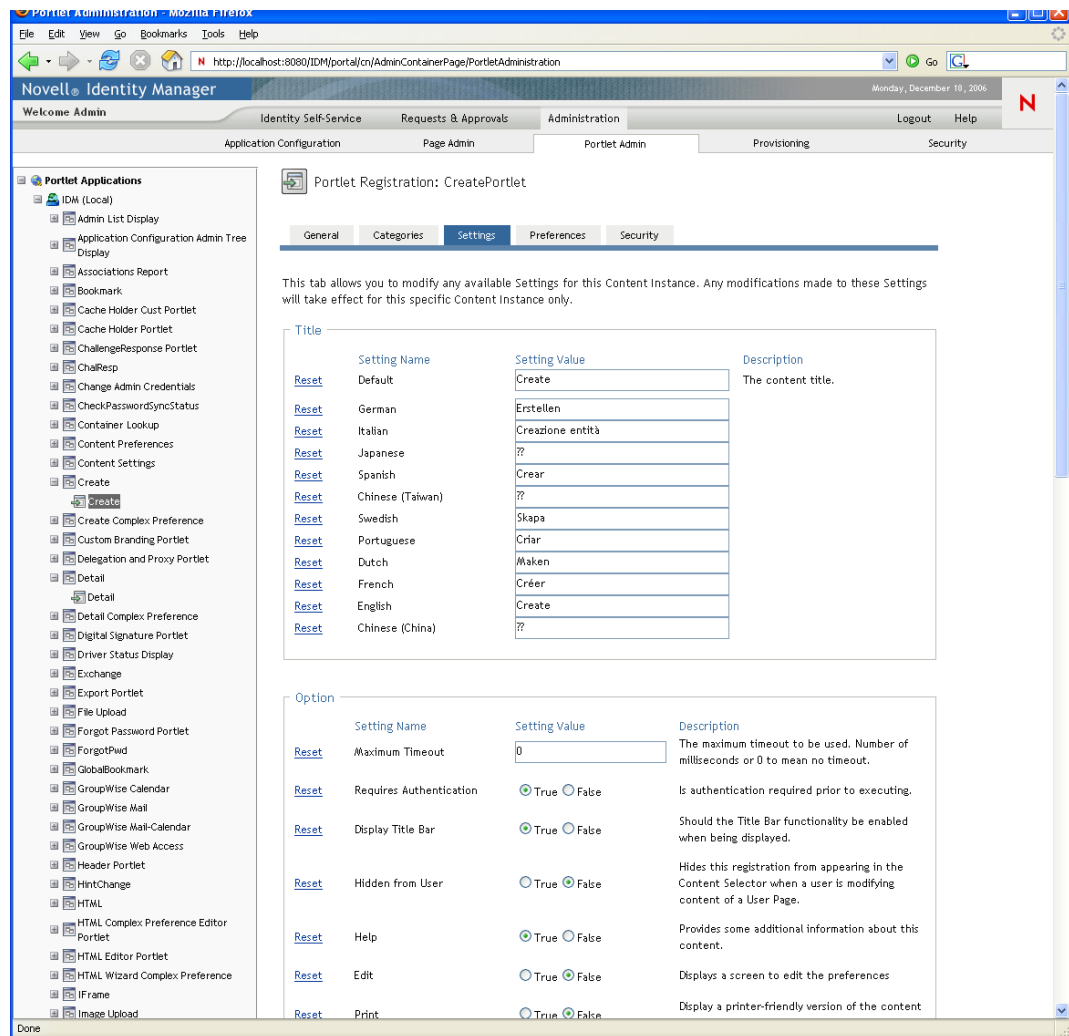
To modify portlet registration settings:

- 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, you can also perform the following actions:

If you want to	Do this
Discard your unsaved changes	Click <i>Cancel</i>
Return all settings for this portlet registration to their default values (as defined in the corresponding portlet definition)	Click <i>Reset All</i>
Return an individual setting to its default value	Click the <i>Reset</i> link beside that setting

4 Click *Save Settings*.

7.3.5 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:

The screenshot shows the Novell Identity Manager Portlet Administration web interface. On the left is a tree view of 'Portlet Applications' including 'IDM (Local)' and various portlets like 'Admin List Display', 'Application Configuration Admin Tree', 'display', 'Associations Report', 'Bookmark', 'Cache Holder Cust Portlet', 'Cache Holder Portlet', 'ChallengeResponse Portlet', 'ChalResp', 'Change Admin Credentials', 'CheckPasswordSyncStatus', 'Container Lookup', 'Content Preferences', 'Content Settings', 'Create', 'Create Complex Preference', 'Custom Branding Portlet', 'Delegation and Proxy Portlet', 'Detail', 'Detail Complex Preference', 'Digital Signature Portlet', 'Driver Status Display', 'Exchange', 'Export Portlet', 'File Upload', and 'Forgot Password Portlet'. The 'Create' portlet is selected. The main area shows the 'Portlet Registration: CreatePortlet' configuration page with tabs for 'General', 'Categories', 'Settings', 'Preferences', and 'Security'. The 'Preferences' tab is active, displaying a table of preferences for the 'CreatePortlet' registration. The table has columns for 'Preference', 'Preference Value', 'Required', and 'Read only'. The preferences listed are: 'Detail Portlet Name' (value: DetailPortlet), 'Custom Class Name' (value: com.novell.srvprv.impl.portlet.create.C), 'Expire password on initial login?' (value: True), 'Display password with attributes?' (value: False), and 'Create Virtual Entity' (value: View/Edit Custom Preference complex preference). At the bottom of the table are buttons for 'Save Preferences', 'Cancel', 'Reset All', and 'Descriptions'.

Preference	Preference Value	Required	Read only
Reset Detail Portlet Name	<input type="text" value="DetailPortlet"/>	Detail <input checked="" type="checkbox"/>	<input type="checkbox"/>
Reset Custom Class Name	<input type="text" value="com.novell.srvprv.impl.portlet.create.C"/>	Detail <input checked="" type="checkbox"/>	<input type="checkbox"/>
Reset Expire password on initial login?	<input checked="" type="radio"/> True <input type="radio"/> False	Detail <input checked="" type="checkbox"/>	<input type="checkbox"/>
Reset Display password with attributes?	<input type="radio"/> True <input checked="" type="radio"/> False	Detail <input checked="" type="checkbox"/>	<input type="checkbox"/>
Reset Create Virtual Entity	View/Edit Custom Preference complex preference	<input checked="" type="checkbox"/>	<input type="checkbox"/>

[Save Preferences](#) [Cancel](#) [Reset All](#) [Descriptions](#)

- 3 Modify preferences, as appropriate.

While working on this panel, you can also perform the following actions:

If you want to	Do this
Display more information about the preferences	Click <i>Descriptions</i>
Discard your unsaved changes	Click <i>Cancel</i>
Return all preferences for this portlet registration to their default values (as defined in the corresponding portlet definition)	Click <i>Reset All</i>

If you want to	Do this
Return an individual preference to its default value	Click the <i>Reset</i> link next to that preference

- 4 To modify the localized version of a preference for each locale specified in the portlet definition:
 - 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*.

7.3.6 Assigning Security Permissions for Portlet Registrations

You can assign the security permissions described in [Table 7-2](#) to users, groups, and containers for portlet registrations.

Table 7-2 *Security Permissions for Portlet Registrations*

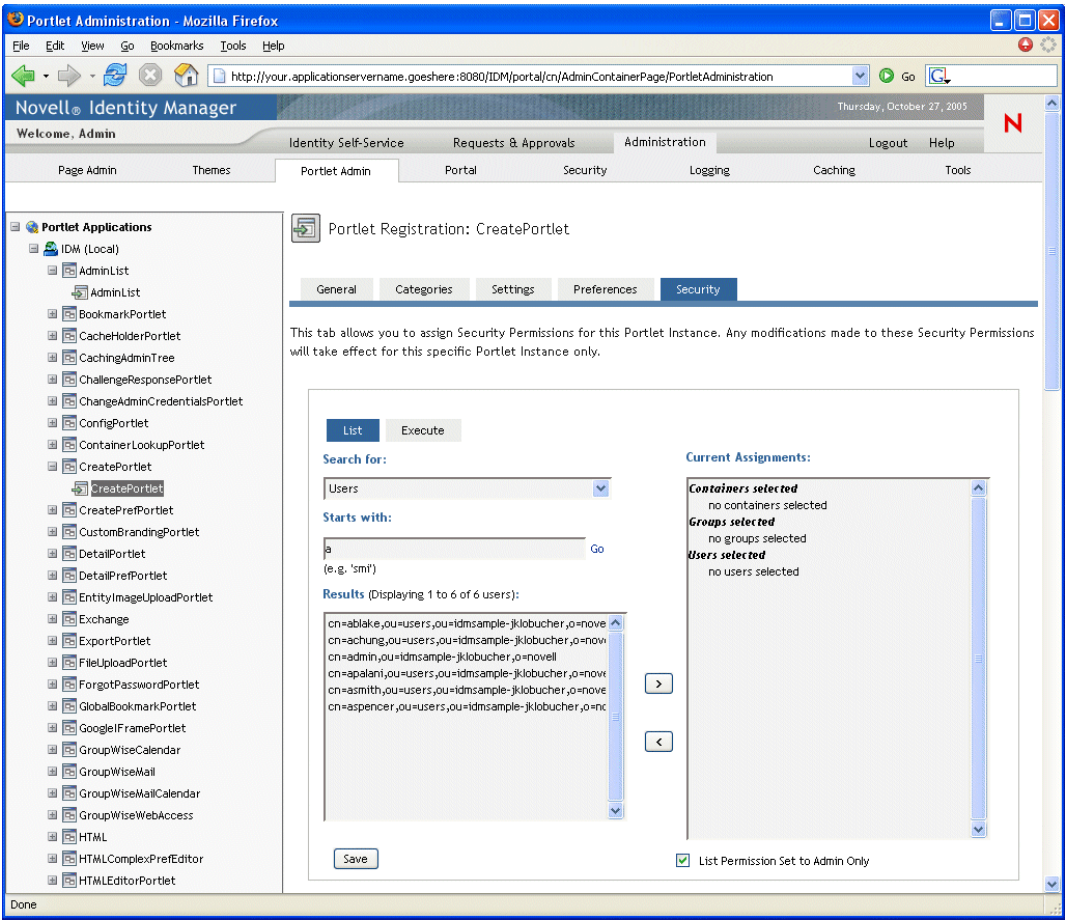
Permission	Description
List	Users can view the portlet registration from a selection list
Execute	Users can run the portlet registration on a portal page

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 drop-down menu: <ul style="list-style-type: none">♦ Users♦ Groups♦ Containers

Setting	What to do
Starts with	<p>If you want to:</p> <ul style="list-style-type: none"> 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 characters of the CN values you want. (Case is not considered. Wildcards are not supported.) <p>For example, searching for groups that start with <code>s</code> would narrow your search results to something like this:</p> <pre>cn=Sales,ou=groups,o=MyOrg cn=Service,ou=groups,o=MyOrg cn=Shipping,ou=groups,o=MyOrg</pre> <p>Searching for groups that start with <code>Se</code> would return:</p> <pre>cn=Service,ou=groups,o=MyOrg</pre>

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.

Hold down the Ctrl 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	Select List/Execute Permission Set to Admin Only
Allow all assigned users, groups, and containers to list/execute the portlet registration	Deselect List/Execute Permission Set to Admin Only
<p>NOTE: If you deselect this setting but there are no users, groups, or containers explicitly assigned to the portlet registration, then everyone has List/Execute permission for this portlet registration.</p>	

8 Click *Save*.

7.3.7 Unregistering a Portlet

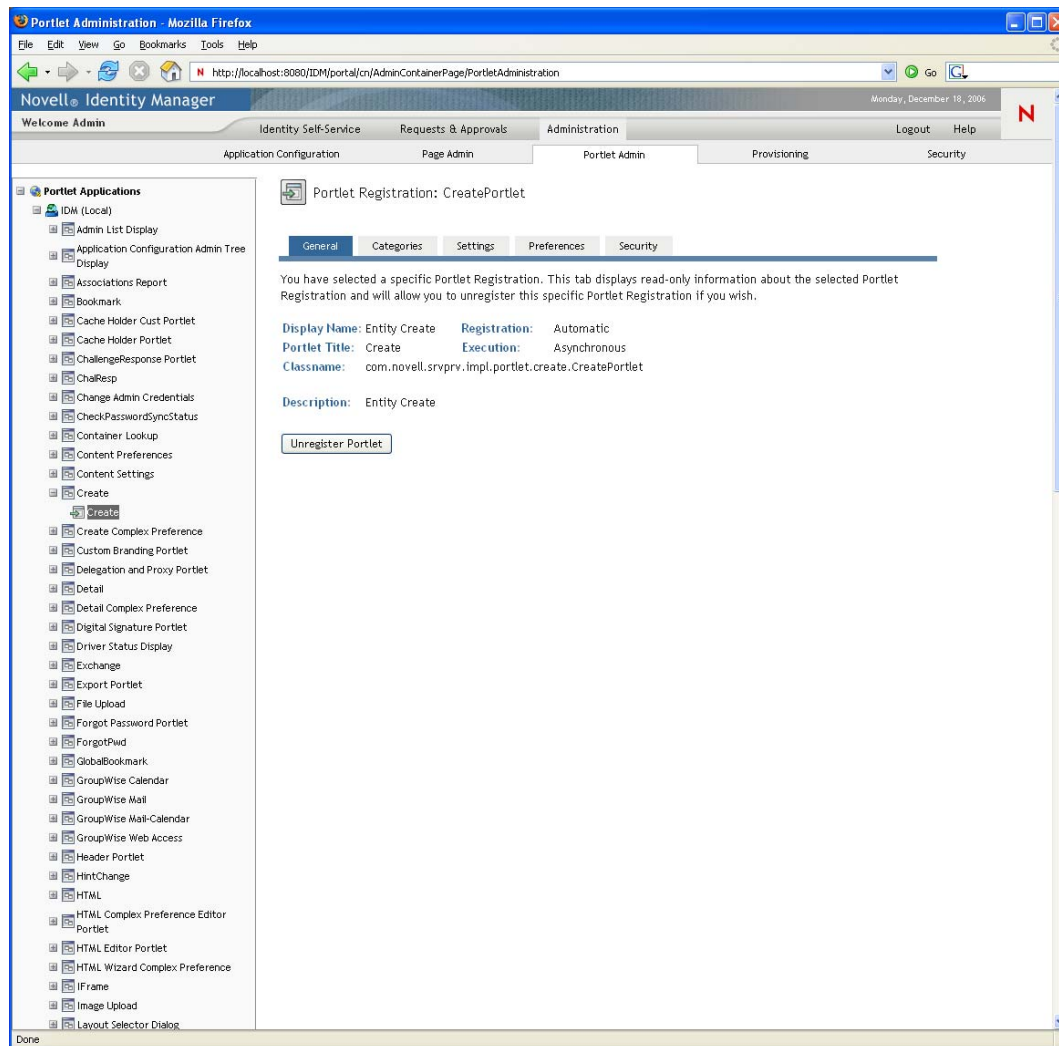
You can use the Portlet Admin page to unregister a portlet if necessary.

NOTE: If you unregister a portlet that is defined as auto-registered, that portlet is 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*.

Provisioning Configuration

8

This section describes the tasks that you can perform from the Provisioning Configuration page. Topics include:

- ♦ [Section 8.1, “About Provisioning Configuration,” on page 189](#)
- ♦ [Section 8.2, “Configuring Delegation, Proxy, and Task Settings,” on page 189](#)
- ♦ [Section 8.3, “Configuring the Digital Signature Service,” on page 193](#)
- ♦ [Section 8.4, “Configuring the Workflow Engine and Cluster Settings,” on page 195](#)

8.1 About Provisioning Configuration

This section provides instructions for using the Provisioning page to administer the workflow-based provisioning features of the User Application. To access the Provisioning page, you need to have the Provisioning Module for Identity Manager. In addition, you need to log on as a Provisioning Application Administrator.

8.2 Configuring Delegation, Proxy, and Task Settings

This section includes information about:

- ♦ [Section 8.2.1, “Configuring the Delegation and Proxy Service,” on page 189](#)
- ♦ [Section 8.2.2, “Scheduling Synchronization and Cleanup,” on page 191](#)
- ♦ [Section 8.2.3, “Configuring Provisioning Interface Display Settings,” on page 192](#)

8.2.1 Configuring the Delegation and Proxy Service

To configure the Delegation and Proxy Service:

- 1 Select the *Provisioning* tab.
- 2 Select *Delegation, Proxy and Tasks* from the left navigation menu.

The user interface displays the Delegation, Proxy and Tasks page. To configure the service, you need to make some changes in the Delegation and Proxy Service Settings box.

- 3 Check the *Allow All Requests* option if you want to display the *All* option in the Resource Search Criteria drop-down list for the Team Delegate Assignments action. When the *All* option is available, a delegate assignment can be defined that applies to all resource categories.
- 4 Define the retention period for delegate, proxy, and availability assignments:

Field	Description
<i>Retention time for Delegation assignments</i>	Specifies the number of minutes to retain delegate assignments in the directory after they have expired. The default is 0, which indicates that the assignments will be removed after the expiration time has been reached.
<i>Retention time for Proxy assignments</i>	Specifies the number of minutes to retain proxy assignments in the directory after they have expired. The default is 0, which indicates that the assignments will be removed after the expiration time has been reached.
<i>Retention time for Availability settings</i>	Specifies the number of minutes to retain availability settings in the directory after they have expired. The default is 0, which indicates that the assignments will be removed after the expiration time has been reached.

- 5 Select the e-mail templates you want to use for delegation, proxy, and availability notifications:

Field	Description
<i>Delegation notification template</i>	<p>Specifies the language-independent name for the template to use for delegation e-mail notifications. After the template name has been specified, the notification engine can determine which language-specific template to use at runtime.</p> <p>For details on creating and editing e-mail templates, see Section 18.4, “Working with E-Mail Templates,” on page 334.</p>
<i>Proxy notification template</i>	<p>Specifies the language-independent name for the template to use for proxy e-mail notifications. After the template name has been specified, the notification engine can determine which language-specific template to use at runtime.</p> <p>For details on creating and editing e-mail templates, see Section 18.4, “Working with E-Mail Templates,” on page 334.</p>
<i>Availability notification template</i>	<p>Specifies the language-independent name for the template to use for availability e-mail notifications. After the template name has been specified, the notification engine can determine which language-specific template to use at runtime.</p> <p>For details on creating and editing e-mail templates, see Section 18.4, “Working with E-Mail Templates,” on page 334.</p>

8.2.2 Scheduling Synchronization and Cleanup

To configure the Synchronization and Cleanup Service:

- 1 Select the *Provisioning* tab.
- 2 Select *Delegation, Proxy and Tasks* from the left navigation menu.

The user interface displays the Delegation, Proxy and Tasks page. To schedule synchronization and cleanup, you need to make some changes in the Synchronization and Cleanup Service box.

Synchronization and Cleanup Service

Set synchronization time for delegation, proxy and availability settings. Activation interval change will take effect the next time application starts up.

Synchronization Service Activation Interval (minutes):

Set cleanup service to delete assignments and settings that have passed retention time, using one of the following methods. Activation interval change will take effect the next time application starts up.

☒ Cleanup Service Activation Interval (minutes):

☐ Cleanup Date:

Last cleanup performed:

- 3 To specify how often you want to activate the synchronization service, type the activation interval (in minutes) in the *Synchronization Service Activation Interval* field. The default value is 0, which means that the service is not activated.

When the synchronization service runs, any modifications (or deletions) made to delegate assignments are synchronized with the corresponding availability settings for the user.

- 4 To specify how often you want to activate the cleanup service, select *Cleanup Service Activation Interval*, then type the activation interval (in minutes). Alternatively, select *Cleanup Date* and use the calendar tool to specify the date when you want to activate the service. The default value is 0, which means that the service is not activated.

When the cleanup service runs, all obsolete proxy and delegate assignments are removed from the system.

If the cleanup service has been activated, the *Last cleanup performed* field indicates when the last cleanup was performed.

8.2.3 Configuring Provisioning Interface Display Settings

To configure the Provisioning Interface display settings:

- 1 Select the *Provisioning* tab.
- 2 Select *Delegation, Proxy and Tasks* from the left navigation menu.

The user interface displays the Delegation, Proxy and Tasks page. To schedule synchronization and cleanup, you need to make some changes in the Provisioning Interface Display Settings box.

Provisioning Interface Display Settings

Changes to display settings will take effect the next time application starts up.

Default landing page:

getAFTaskList.do?apwaLe

Maximum number of results returned from a query:

50

Maximum number of results displayed per page:

5

- 3 To change the default landing page, type the URL for another page in the *Default Landing Page* field. The default page is shown below:

```
getAFTaskList.do?apwaLeftNavItem=JSP_MENU_TASKS&apwaActionType=user
```

The page you specify must be reference a servlet that is available from the *Requests & Approvals* tab. To change the landing page, you can click on the desired page in the left-hand navigation panel on the *Requests & Approval*, and then cut and paste the last part of the URL after the application context (IDMProv) into the *Default Landing Page* field. For example, to set the landing page to My Requests, you could paste the following string into the field:

```
getAFProcessList.do?apwaLeftNavItem=JSP_MENU_REQUESTS&apwaActionScope=user&apwaNewSearch=true
```

- 4 To set the number of rows returned from each query, type the row limit in the *Maximum number of results returned from a query* box. The default is 50.
- 5 To set the number of rows displayed on each page, type the display limit in the *Maximum number of results displayed per page* box. The default is 5.

The *Maximum number of results displayed per page* values affect the display on several screens on the *Requests & Approvals* tab, including *My Tasks*, *My Requests*, *My Proxy Assignments*, and *My Delegate Assignments*.

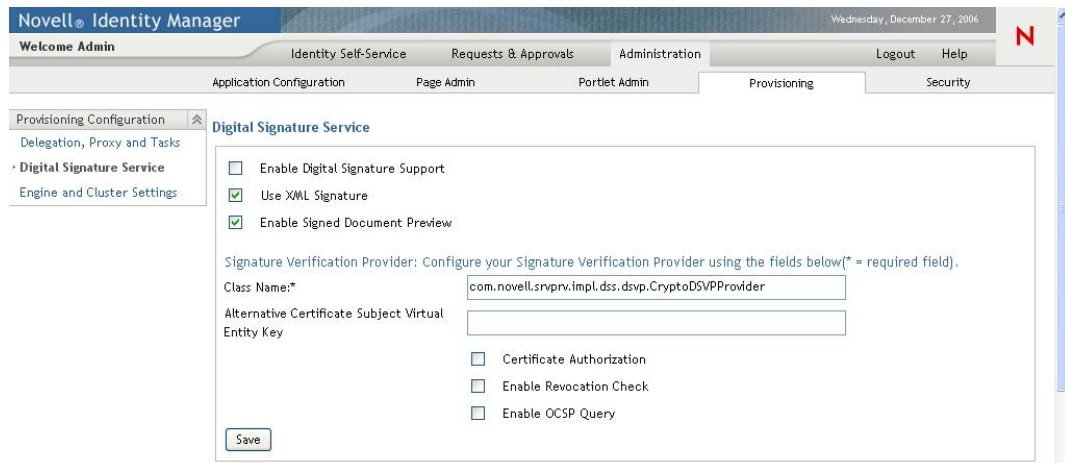
8.3 Configuring the Digital Signature Service

This section provides details on configuring the Digital Signature Service.

To configure the Digital Signature Service:

- 1 Select the *Provisioning* tab.
- 2 Select *Digital Signature Service* from the left navigation menu.

The user interface displays the Digital Signature Service panel:



The screenshot shows the Novell Identity Manager web interface. The top navigation bar includes 'Welcome Admin', 'Identity Self-Service', 'Requests & Approvals', 'Administration', 'Logout', and 'Help'. Below this, a secondary navigation bar shows 'Application Configuration', 'Page Admin', 'Portlet Admin', 'Provisioning', and 'Security'. The 'Provisioning' tab is selected. On the left, a sidebar menu lists 'Provisioning Configuration', 'Delegation, Proxy and Tasks', 'Digital Signature Service' (which is highlighted), and 'Engine and Cluster Settings'. The main content area is titled 'Digital Signature Service' and contains the following configuration options:

- ☐ Enable Digital Signature Support
- ☒ Use XML Signature
- ☒ Enable Signed Document Preview

Below these options, a section titled 'Signature Verification Provider: Configure your Signature Verification Provider using the fields below(* = required field)' contains the following fields:

- Class Name:** A text box containing 'com.novell.srvprv.impl.dss.dsvp.CryptoDSVPProvider'.
- Alternative Certificate Subject Virtual Entity Key:** An empty text box.
- ☐ Certificate Authorization
- ☐ Enable Revocation Check
- ☐ Enable OCSP Query

A 'Save' button is located at the bottom left of the configuration area.

- 3 Perform these steps to configure the Digital Signature Service:

- 3a Select the *Enable Digital Signature Support* check box.

If this check box is not selected, users will see an error message when they try to access any provisioning resource that requires a digital signature.

Before enabling digital signature support, make sure all of the required JARs are present. If any of the JARs are missing, you will see an error message when you select the check box. For details on which JARs are required for digital signatures, see [Section 2.3, “Digital Signature Configuration,” on page 43](#).

- 3b Select the *Use XML Signature* check box if you’re using cryptovision.

- 3c Optionally select the *Enable Signed Document Preview* to allow users to preview signed documents.

- 3d Type the name of the class for your digital signature service in the *Class Name* field.

For details on using cryptovision as your signature verification provider, see <http://www.cryptovision.com/idmdigsig.html>.

- 3e Optionally specify an entity key in the *Alternative Certificate Subject Virtual Entity Key* field. The entity key maps to an entity defined in the data abstraction layer. The entity provides a calculated attribute that can be used instead of the LDAP common name to ensure that only authorized users can perform digital signing. In the Designer, you define

the entity, giving the key any name you like. On the Digital Signature Service configuration panel, you specify the key for the entity you defined. The alternative subject is an optional feature that you can use to add an extra layer of protection.

- 3f** Optionally select the *Certificate Authorization* check box to ensure that the authenticated user matches the user associated with the selected user certificate. When *Certificate Authorization* is enabled, the current user is not permitted to use a certificate on the smart card (or browser) that has been given to a different user.
- 3g** Optionally select the *Enable Revocation Check* check box to cause the application to check the certificate revocation list (CRL) before using a certificate to be sure that it is still valid. A certificate might be revoked for several reasons. For example, the certificate authority might determine that a particular certificate was improperly issued. Alternatively, the certificate might be revoked if the private key for the certificate has been lost or stolen.
- 3h** Optionally select the *Enable OCSP Query* check box to perform a query against an Online Certificate Status Protocol (OCSP) server before using a certificate. OCSP is an alternative to certificate revocation lists that addresses problems associated with using CRLs in a public key infrastructure (PKI). The OCSP access point for the server is specified in the User Application Configuration utility.
- 4** To view the settings for a previously configured applet, select the applet from the *Signature Applet* dropdown list.

Signature Applet

Use the dropdown menu below to view each digital signature applet settings that are currently configured. Click the Add or Remove button to add a new applet or remove the currently selected applet.

Signature Applet	<div>IE Cryptovision Applet</div> <div>IE Cryptovision Applet</div> <div>FireFox Cryptovision Applet</div>	
Class ID:		0805F499D93
Archive Name:	SAfXIE.jar	
Context Root:	/xmlsigner	
Callback Name:	myCallback	
Declaration Template:	<pre><object id="signer_id" classid="\$classid" height=16 width=16> <param name="code" value="com/cryptovision/safx/SAfXIE.class"/> <param name="archive" value="\$root/\$archive"/> <param name="mayscript" value="true"/> </object></pre>	
Invocation Template:	<pre>document.signer_id.applet_sign(\$input,"\$callback")</pre>	
Callback Function Template:	<pre>\$callback=function(res){ \$storerestresult(res) ;}</pre>	
Browsers:	IE_6_0_WIN	
<div>Add</div> <div>Remove</div>		

For details on configuring the cryptovision applet, see <http://www.cryptovision.com/idmdigsig.html>.

- 5** Perform these steps to add a new signature applet configuration:
 - 5a** Click *Add*.

The user interface makes the fields in the Signature Applet panel editable.
 - 5b** Provide a name for this applet configuration in the *Display Name* field.
 - 5c** Specify the class ID for the applet in the *Class ID* field.
 - 5d** Specify the entry of the JAR that contains the applet in the *Archive Name* field.

- 5e** Specify <context root path> of the Web application that contains the applet archive for the *Context Root*. (If the context root points to a different application, always start it with a “/” character.)
- 5f** Specify the callback name in the *Callback Name* field.
- 5g** Specify the XML declaration string in the *Declaration Template* field.
- 5h** Specify the invocation string in the *Invocation Template* field.
- 5i** Specify the callback function in the *Callback Function Template* field.
- 5j** Select the browser type (for example, IE 6.0) in the *Browser Type* select list.
- 6** Click *Save* to save your settings.

8.4 Configuring the Workflow Engine and Cluster Settings

This section provides instructions on configuring the Workflow Engine and on configuring cluster settings. These settings apply to all engines in the cluster. When any of these settings are changed, other engines in the cluster will detect these changes in the database and use the new values. The engines check for changes to these settings at the same rate as specified by the pending process interval.

The process cache settings and heartbeat settings require a server restart to take effect.

8.4.1 Configuring the Workflow Engine

To configure the Workflow Engine settings:

- 1** Select the *Provisioning* tab.
- 2** Select *Engine and Cluster Settings* from the left navigation menu.

The user interface displays the Workflow Configuration Settings page. To configure the engine, you need to make some changes in the Workflow Engine box.

Workflow Engine

Modify any of the settings below to change the current workflow engine configurationAll fields are required

Email Notification (per workflow engine):
☒ Enable
☐ Disable

Web Service Activity Timeout (minute):
(valid range: 1 minute to 7 days)

User Activity Timeout (hour, 0 for no timeout):
(valid range: 0 hour to 365 days)

Completed Process Timeout (day):
(valid range: 0 day to 365 days)

Completed Process Cleanup Interval (hour):

Pending Process Interval (second):

Retry Queue Interval (minute):

Maximum Thread Pool Size:

Minimum Thread Pool Size:

Initial Thread Pool Size:

Thread Keep Alive Time (second):

Process Cache Load Factor:
(valid range: 0 to 1)

Process Cache Initial Capacity:

Process Cache Maximum Capacity:

Maximum Engine Shutdown Timeout (minute):

- To change an engine setting, click the target field for the setting and type the new value. The engine settings are described below:

Engine Setting	Description
<i>Email Notification (per workflow engine)</i>	Enables or disables e-mail notifications for the entire workflow engine. Defaults to enabled.
<i>Web Service Activity Timeout (minute)</i>	Specifies the default Web Service activity timeout in minutes. The default is 50 minutes.
<i>User Activity Timeout (hour, 0 for no timeout)</i>	Specifies the default user activity timeout. The default is 0 days, which indicates no timeout.
<i>Completed Process Timeout (day)</i>	Specifies the number of days that a completed process state is kept in the system. The default is 120 days.
<i>Completed Process Cleanup Interval (hour)</i>	Specifies how often the engine checks for and removes completed processes that have been in the system for longer than the completed process timeout. The default is 12 hours.
<i>Pending Process Interval (second)</i>	User activities that are executed on an engine which the process is not bound to are put into a pending state. This interval specifies how often to check for pending activities in order to continue their execution. The default is 30 seconds.

Engine Setting	Description
<i>Retry Queue Interval (minute)</i>	Activities that fail because of suspected database connectivity issues are put on a retry queue. This interval specifies how often the engine attempts to retry these activities. The default is 15 minutes.
<i>Maximum Thread Pool Size</i>	The maximum number of threads that the engine uses to execute activities. The default is 20.
Minimum Thread Pool Size	The minimum number of threads that the engine uses to execute activities. When a thread is requested and fewer than the minimum are in the pool, a new thread will be created even if there are idle threads in the pool. The default is 10.
<i>Initial Thread Pool Size</i>	Number of prestarted threads in the pool when it is created. The default is 5.
<i>Thread Keep Alive Time (second)</i>	If the pool is larger than the minimum size, excess threads that have been idle for more than the keep alive time will be destroyed. The default is 5 minutes.
<i>Process Cache Load Factor</i>	The load factor specifies how full the cache is allowed to get before increasing its capacity. If the number of entries in the cache exceeds the product of the load factor multiplied by the current capacity, then the capacity is increased. The default is 0.75.
<i>Process Cache Initial Capacity</i>	The process cache is backed by a hash map. The capacity is the number of buckets in the hash map. The initial capacity is the number of buckets at the time the cache is created. The default is 700.

Engine Setting	Description
<i>Process Cache Maximum Capacity</i>	<p>Before adding a process to the cache, if the number of processes in the cache equals or exceeds the Process Cache Maximum Capacity, the cache attempts to remove the oldest inactive process from the cache. The maximum capacity is a soft limit, so the number of processes in the cache might exceed the Process Cache Maximum Capacity if there are no inactive processes (only active processes) in the cache.</p> <p>A good value for this setting should be less than product of the Process Cache Initial Capacity and the Process Cache Load Factor. This gives the cache a chance to remove older inactive processes from the cache before having to increase its capacity.</p> <p>Take the following example:</p> <p>Process Cache Initial Capacity = 700;</p> <p>Process Cache Load Factor = .75;</p> <p>Process Cache Maximum Capacity = 500;</p> <p>Number of processes in cache = 500;</p> <p>In this case, the number of processes in the cache that will trigger the cache to grow its capacity and perform a rehash would be 525, because the Initial capacity multiplied by the load factor is equal to 525.</p> <p>In this example, when there are 500 processes in the cache, the cache is approaching the point where it must increase its size and perform a rehash, which is at 525 processes. When another process is added to the cache, the engine attempts to remove the least recently used inactive process instead of letting the cache get closer to 525 processes.</p> <p>The default is 500.</p>
<i>Maximum Engine Shutdown Timeout (minute)</i>	<p>The engine attempts to shutdown gracefully. When shutting down it stops queuing new activities for execution and attempts to complete any activities already queued. This timeout specifies the maximum time that the engine waits for all queued activities and threads executing activities to complete. If this time is exceeded, the engine halts processing of queued activities and attempts to stop all threads executing activities. The default is 1 minute.</p>

8.4.2 Configuring the Workflow Cluster

To configure the Workflow Cluster settings:

- 1 Select the *Provisioning* tab.
- 2 Select *Engine and Cluster Settings* from the left navigation menu.

The user interface displays the Workflow Configuration Settings page. To configure cluster settings, you need to make some changes in the Workflow Cluster box.

Workflow Cluster

Modify any of the settings below to change the current cluster configuration. Review the list of each workflow engine in the cluster for engine ID and engine stateAll fields are required

Heartbeat Interval (second, minimum 60):

60

Heartbeat Factor (minimum 2):

2

Engine ID(Read Only)

ENGINE

Engine State(Read Only)

Running

- 3 To change a cluster setting, click the target field for the setting and type the new value. The cluster settings are described below:

Cluster Setting	Description
<i>Heartbeat Interval (second, minimum 60)</i>	<p>Specifies the interval at which the workflow engine’s heartbeat is updated.</p> <p>When the workflow engine starts up, it detects if its engine ID is already being used by another node in the cluster and refuses to start if the ID is in use. The User Application database maintains a list of engine IDs and engine states. If an engine crashes and is restarted, its last state in the database indicates that it is still running. The workflow engine therefore uses a heartbeat timer, which writes heartbeats at the specified interval, to determine if an engine with its ID is still running in the cluster. If it’s already running, it refuses to start.</p> <p>The minimum value for the heartbeat interval is 60 seconds.</p>
<i>Heartbeat Factor (minimum 2)</i>	<p>Specifies the factor that is multiplied with the heartbeat interval to arrive at the heartbeat timeout.</p> <p>The timeout is the maximum elapsed time permitted between heartbeats before an engine will be considered timed out.</p> <p>The minimum value for the heartbeat factor is 2.</p>

This section describes how to use the Security page on the *Administration* tab of the Identity Manager User Application. Topics include:

- ♦ [Section 9.1, “About Security Configuration,” on page 201](#)
- ♦ [Section 9.2, “Assigning the User Application Administrator,” on page 202](#)
- ♦ [Section 9.3, “Assigning the Provisioning Administrator,” on page 203](#)

For general information about accessing and working with the *Administration* tab, see [Chapter 4, “Using the Administration Tab,” on page 83](#).

9.1 About Security Configuration

The Identity Manager 3.5 User Application assigns administrative tasks to Provisioning Application Administrators and User Application Administrators.

Table 9-1 *Types of Administrator*

This Role	Can Perform
User Application Administrator	Application administration tasks, in the <i>Administration</i> tab in the User Application.
Provisioning Application Administrator	Provisioning workflow management tasks, in the <i>Requests and Approvals</i> tab in the User Application.

You can assign these roles at installation and on the Security page on the *Administration* tab of the Identity Manager User Application. After you start the JBoss Application Server the first time after installation, you cannot change these assignments with the `configupdate` utility.

9.1.1 The User Application Administrator

The User Application Administrator performs administrative tasks for the Identity Manager User Application, using the *Administration* panel of the Identity Manager User Application. The User Application Administrator does not have provisioning administration rights, and is considered an ordinary user while using the *Requests and Approvals* panel. There can be more than one User Application Administrator.

One user *must* be assigned to the User Application Administrator role at installation. The User Application Administrator created during installation (also known as the locksmith user) can administer everything in the User Application including the Provisioning system and can designate other users as User Application Administrators or Provisioning Application Administrators.

A user who is to be a User Application Administrator should typically be located under the user root container specified in the User Application’s LDAP configuration. This enables the user to log in simply by username (instead of requiring the fully distinguished name each time).

The user who is a User Application Administrator does not need special directory rights because this role controls application-level access.

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 6, “Page Administration,” on page 139.](#))

9.1.2 The Provisioning Application Administrator

The Provisioning Application Administrator administers the Provisioning system and not the User Application. The Provisioning Application Administrator has rights and permissions for all functions (is essentially a “superuser”) within the *Requests and Approvals* panel.

A Provisioning Application Administrator is assigned at installation. Create them as soon as possible after installation to keep your system secure. If there is no Provisioning Application Administrator, every logged-in user is treated as a Provisioning Application Administrator. This is not secure.

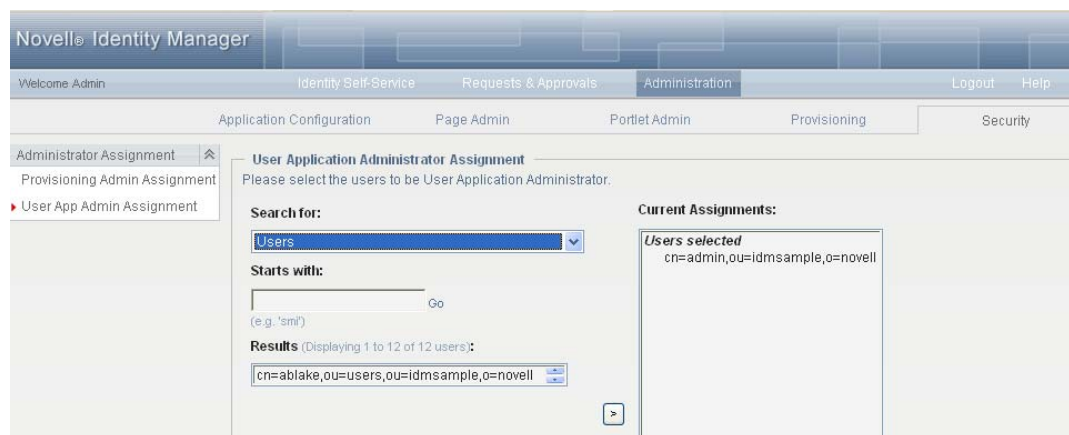
A Provisioning Application Administrator can assign other users to be Provisioning Application Administrators. However, he must be a User Application Administrator in order to get access to the provisioning administrator assignment page in the administration console.

You might prefer to locate a user who is to be a Provisioning Application Administrator under the user root container specified in the User Application’s LDAP configuration. This location enables the user to log in simply by username (instead of requiring the fully distinguished name each time).

9.2 Assigning the User Application Administrator

When assigning User Application Administrators, you can specify users.

- 1 Go to the Security page:



- 2 Under *Administrator Assignment*, select *User App Admin Assignment*.
- 3 Specify values for the following search settings:

Setting	What to Do
Search for	Select one of the following from the drop-down menu: <ul style="list-style-type: none"> ♦ Users
Starts with	If you want to: <ul style="list-style-type: none"> ♦ Find all available objects of your specified type (user), then make this setting blank. ♦ Find a subset of those objects, then enter the starting characters of the CN values you want. (Case is not considered. Wildcards are not supported.)

4 Click *Go*.

The results of your search appear in the Results list.

5 Select the users you want to assign as User Application Administrators, then click *Add* (>).

Hold down the Ctrl key to make multiple selections.

6 Click *Save*.

To unassign User Application Administrators:

1 In the Current Assignments list, select the users you want to unassign as User Application Administrators, then click *Remove* (<).

Hold down the Control key to make multiple selections.

2 Click *Save*.

You cannot delete yourself as User Application Administrator. This is a safeguard to ensure that the User Application always has at least one User Application Administrator.

9.3 Assigning the Provisioning Administrator

When assigning Provisioning Administrators, you can specify users, groups, or containers.

1 Go to the Security Page.

2 Under *Administrator Assignment*, select *Provisioning Admin Assignment*.

The screenshot shows the Novell Identity Manager web interface. The top navigation bar includes 'Welcome Admin', 'Identity Self-Service', 'Requests & Approvals', 'Administration' (selected), 'Logout', and 'Help'. Below this, a sub-navigation bar shows 'Application Configuration', 'Page Admin', 'Portlet Admin', 'Provisioning' (selected), and 'Security'. On the left, a sidebar menu has 'Administrator Assignment' selected, with sub-items 'Provisioning Admin Assignment' and 'User App Admin Assignment'. The main content area is titled 'Provisioning Administrator Assignment' and contains the instruction: 'Please select the users, groups and containers to be Provisioning Administrator.' It features a 'Search for:' dropdown menu set to 'Users', a 'Starts with:' text input field containing 'a', and a 'Go' button. Below these is a 'Results' section showing 'Displaying 1 to 12 of 12 users:'. To the right of the search area is a 'Current Assignments:' box with a list of selected items: 'Containers selected' (no containers selected), 'Groups selected' (no groups selected), and 'Users selected' (cn=admin,ou=idmsample,o=novell). A 'Save' button is located at the bottom left of the main content area.

- 3** Search for the users, groups, or containers you want to assign. Specify values for the following search settings:

Setting	What to Do
Search for	Select one of the following from the drop-down menu: <ul style="list-style-type: none">♦ Users♦ Groups♦ Containers
Starts with	If you want to: <ul style="list-style-type: none">♦ 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 characters of the CN values you want. (Case is not considered. Wildcards are not supported.) For example, searching for groups that start with <i>s</i> would narrow your search results to something like this: <code>cn=Sales,ou=groups,o=MyOrg</code> <code>cn=Service,ou=groups,o=MyOrg</code> <code>cn=Shipping,ou=groups,o=MyOrg</code> Searching for groups that start with <i>se</i> would return: <code>cn=Service,ou=groups,o=MyOrg</code>

- 4** Click *Go*. The results of your search appear in the Results list.
- 5** Select the users, groups, or containers you want to assign as Provisioning Administrators, then click *Add* (>).
- Hold down the Ctrl key to make multiple selections.
- 6** Click *Save*.

To unassign Provisioning Application Administrators:

- 1** In the Current Assignments list, select the users, groups, or containers you want to unassign as User Application Administrators, then click *Remove* (<).
- Hold down the Control key to make multiple selections.
- 2** Click *Save*.

If you delete Provisioning Application Administrators, keep at least one. One is necessary to protect the security of your system. If you attempt to remove the last Provisioning Application Administrator, you receive an alert.

Portlet Reference

IV

These sections describe how to configure the identity and system portlets used in the Identity Manager user interface:

- ♦ [Chapter 10, “About Portlets,” on page 207](#)
- ♦ [Chapter 11, “Create Portlet Reference,” on page 211](#)
- ♦ [Chapter 12, “Detail Portlet Reference,” on page 219](#)
- ♦ [Chapter 14, “Resource Request Portlet,” on page 263](#)
- ♦ [Chapter 13, “Org Chart Portlet Reference,” on page 235](#)
- ♦ [Chapter 15, “Search List Portlet Reference,” on page 265](#)

This section provides information about the portlets you can use in the Identity Manager User Application. Topics include:

- ♦ [Section 10.1, “Accessory Portlets,” on page 207](#)
- ♦ [Section 10.2, “Admin Portlets,” on page 207](#)
- ♦ [Section 10.3, “Identity portlets,” on page 208](#)
- ♦ [Section 10.4, “System Components,” on page 210](#)

For more information about managing portlets, see [Chapter 7, “Portlet Administration,” on page 173](#).

Many of the portlets include preferences that enable you to customize the portlet’s behavior or appearance. You localize the preferences by clicking the Detail link in the *Content Preferences* page. As a general guideline, if the preference value is a free-form text input field, do not localize it unless the value is a message displayed in the user interface. You can; however, localize the preference name and description. Localizing a preference value, that is not a message, can cause the portlet to malfunction.

10.1 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, see the *Identity Manager Accessory Portlet Reference Guide*.

10.2 Admin Portlets

The portlets in the Admin category are used to control the layout and contents of the user interface.

IMPORTANT: You should not use or modify these portlets. They provide framework services to the User Application.

[Table 10-1](#) describes Admin portlets.

Table 10-1 *Admin Portlets*

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.

Portlet Name	Description
Shared Page Navigation	<p>Displays a menu containing the Identity Manager User Application shared pages.</p> <p>Preferences define what is displayed and how it is displayed.</p> <p>See Section 10.2.1, “Shared Page Navigation Portlet,” on page 208.</p>

10.2.1 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. [Table 10-2 on page 208](#) describes the preferences for the Shared Page Navigation portlet.

Table 10-2 *Shared Page Navigation Portlet: Preferences*

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	<p>Specify one or more of the shared pages categories.</p> <p>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.</p>
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.

10.3 Identity portlets

The Identity portlets are used by the *Identity Self-Service* tab of the Identity Manager User Application. [Table 10-3 on page 208](#) lists the Identity portlets.

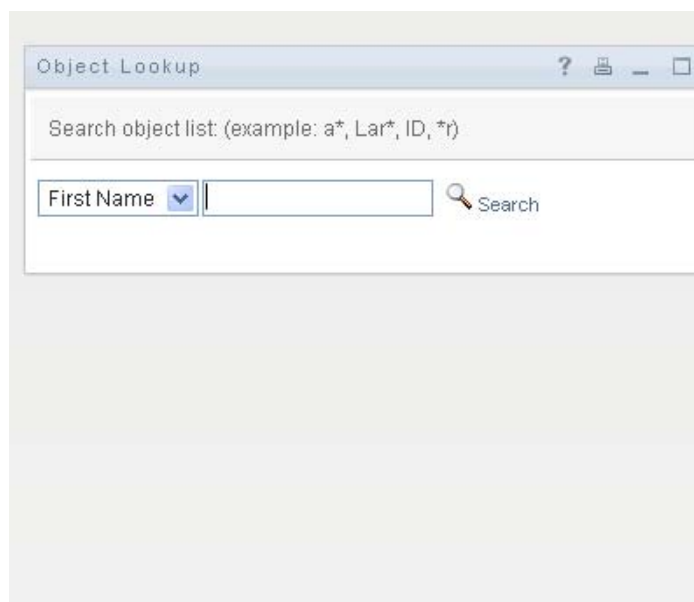
Table 10-3 *Identity Portlets*

Portlet Name	Description
Associations Report	Shows the DirXML-Associations attributes for the logged on user. This attribute maps a user to an external application. There are no preferences for this portlet.
Create	<p>Provides a wizard-based interface that enables users to create objects in the Identity Vault.</p> <p>See Chapter 11, “Create Portlet Reference,” on page 211.</p>

Portlet Name	Description
Detail	Lets users display and manipulate an entity's attribute data. See Chapter 12, "Detail Portlet Reference," on page 219.
Org Chart	Lets users view and browse the hierarchical relationships between objects in the Identity Vault. See Chapter 13, "Org Chart Portlet Reference," on page 235.
Resource Request	Lets you provide access to resource requests to anonymous or guest users. You must create a new shared page for this portlet and ensure that the page is available to guest or anonymous users. See Chapter 14, "Resource Request Portlet," on page 263.
Search List	Allows users to search for objects in the Identity Vault. See Chapter 15, "Search List Portlet Reference," on page 265.

At runtime, the identity portlets might also call the ContainerLookup portlet or the ParamLookup portlet depending on user interaction. The ContainerLookup portlet is launched by the identity portlets when the user performs a lookup on a container object, and the ParamLookup portlet is launched when the user performs a lookup on an attribute. Users launch these portlets by clicking the Lookup button. These portlets have a similar runtime appearance.

Figure 10-1 Sample ParamLookup Portlet



These portlets are also referred to as object selectors, and their contents are defined by the DNLookup definition in the directory abstraction layer. There are no preferences for these portlets, and you cannot add them to a page. The only time you might modify them is when you allow guest access to the identity portlets. The modifications that you need to make for guest access are described in each identity portlet reference section.

10.4 System Components

The system portlets provide services to the Identity Manager User Application.

IMPORTANT: You should not use or modify portlets in this category.

Table 10-4 on page 210 lists the system portlets.

Table 10-4 *System Portlets*

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

11

This section describes how to use the Create portlet in your Identity Manager User Application. Topics include:

- ♦ [Section 11.1, “About the Create portlet,” on page 211](#)
- ♦ [Section 11.2, “Configuring the Create Portlet,” on page 213](#)
- ♦ [Section 11.3, “Setting Preferences,” on page 215](#)
- ♦ [Section 11.4, “Configuring the Create Portlet for Self-Registration,” on page 216](#)

11.1 About the Create portlet

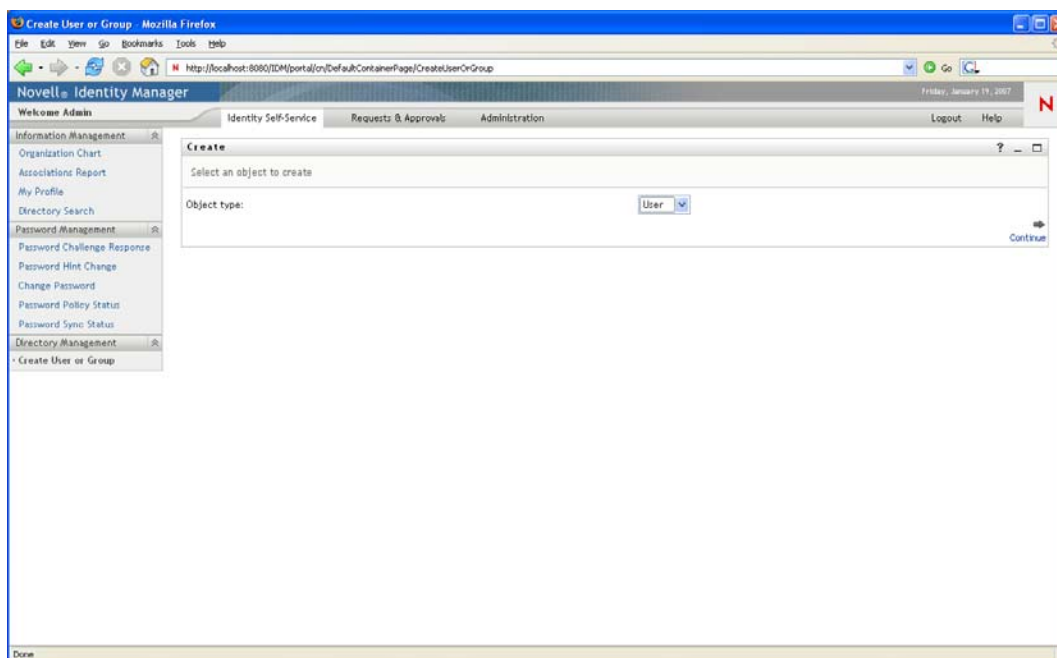
The Create portlet provides an easy-to-use wizard that allows users to create Identity Vault objects of different types. Portlet preferences control the following:

- ♦ The types of objects that the user can create.
- ♦ The attributes that the user can supply.

You can also configure the portlet to allow guest users to self-register.

The default configuration of the Create portlet (accessed via the *Create User or Group* action of the Identity Manager User Application) allows users to create a User or a 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:

Novell Identity Manager

Welcome Admin

Identity Self-Service Requests & Approvals Administration

Logout Help

Friday, January 19, 2007

Information Management

Organization Chart

Associations Report

My Profile

Directory Search

Password Management

Password Challenge Response

Password Hint Change

Change Password

Password Policy Status

Password Sync Status

Directory Management

Create User or Group

Create

User - Set Attributes

* - indicates required.

User ID*

First Name*

Last Name*

Title*

Department*

Region*

Email*

Manager*

Telephone Number*

Back

Continue

- ◆ Prompt for a password, when required by the object type:

Novell Identity Manager

Welcome Admin

Identity Self-Service Requests & Approvals Administration

Logout Help

Friday, January 19, 2007

Information Management

Organization Chart

Associations Report

My Profile

Directory Search

Password Management

Password Challenge Response

Password Hint Change

Change Password

Password Policy Status

Password Sync Status

Directory Management

Create User or Group

Create

Create Password

Password*

Confirm Password*

Back

Continue

If a password policy is assigned, the portlet displays any custom policy messages.

- ◆ Provide an informational message when the object is successfully created. The message contains a link to the Detail portlet for that object for further editing (assuming the Detail portlet is likewise configured).

11.2 Configuring the Create Portlet

Follow the steps in [Table 11-1 on page 213](#) to configure the Create portlet.

Table 11-1 Steps to Configure the Create Portlet

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; otherwise complete the remaining steps.
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 Section 1.2.2, "Directory Abstraction Layer," on page 24 .
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 6, "Page Administration," on page 139 .
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 6, "Page Administration," on page 139 . Edit the portlet instance to change security. For more information on restricting user access to portlets, see Chapter 7, "Portlet Administration," on page 173 . Do you want anonymous users to access this portlet? For more information on setting up the Create portlet specifically for anonymous access, see Section 11.4, "Configuring the Create Portlet for Self-Registration," on page 216 .
5	Set preferences for the portlet.	Preferences let you define: <ul style="list-style-type: none">♦ Which objects users can create.♦ Which attributes to supply during the create. For more information, see Section 11.3, "Setting Preferences," on page 215 .
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 users.	Make sure the users have sufficient rights to create the object.

11.2.1 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 described in [Table 11-2 on page 214](#).

Table 11-2 *Settings for the Directory Abstraction Layer*

Definition Type	Property	Value
entity	<i>create</i>	Selected.
	<i>view</i>	Selected.
		If it is not selected, the entity does not display in the list of entities that can be created.
	<i>Create</i>	<p><i>Container for Create:</i> Specify a valid Identity Vault container. If you do not assign a container, the user is prompted to select one. The user is allowed to select any container beginning with the root container specified during the User Application installation. For anonymous users, it is recommended that you specify a <i>Container for Create</i>. If you do not, then you must also modify the security setting for the <i>ContainerLookupPortlet</i>, as described in Section 11.4, “Configuring the Create Portlet for Self-Registration,” on page 216.</p> <p><i>Create naming attribute:</i> Specify the entity's naming attribute. This shows up in the Create portlet as the Object ID. You can specify different text to display by using the <i>Create naming label</i>.</p> <hr/> <p>NOTE: Because the naming attribute is defined in this way, you do not need to add it to the directory abstraction layer as a separate attribute.</p> <hr/> <p><i>Password Management: Password Required When Entity is Created</i></p> <p>Selected, if the entity type requires a password on create.</p> <p>If the Create portlet is configured to create users and you want to assign the users to an iManager password policy, then you must also assign this container to the same iManager password policy. This ensures that users created in the User Application are automatically assigned to the default iManager password policy.</p> <p>By default, anyone who has access to the Create Users and Groups action and has Trustee rights to the OU can create users and assign the initial password. When the new user first logs in, he or she is redirected to the Change Password page to modify the initial password. You can change the default behavior via the <i>Expire password on initial login</i> preference.</p> <p>For more information on this preference, see Section 11.3, “Setting Preferences,” on page 215.</p> <p>For more information on the Change Password page, Section 5.3.1, “About Password Management Features,” on page 121.</p>
attribute	<i>enabled</i>	Selected.
	<i>viewable</i>	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 [Section 1.2.2, “Directory Abstraction Layer,” on page 24](#).

11.3 Setting Preferences


Preferences allow you to configure the types of objects and the attributes that users are prompted for. There are two types of preferences: general and complex. The general preferences are described in [Table 11-3 on page 215](#) followed by the complex preferences in [Table 11-4 on page 215](#).

Table 11-3 *Create Portlet: General Preferences*

Preference	Description
<i>Detail Portlet Name</i>	Specify the instance of the Detail Portlet to display when the user clicks the <i>Object Created</i> link after the object is successfully created. It defaults to the standard DetailPortlet. See Section 12.6, “Setting up Detail for Anonymous Access,” on page 233 .
<i>Custom Class Name</i>	Specify the name of the class for processing create events. The default is <code>com.novell.srvprv.impl.portlet.create.CreateCustomEventDefaultHandler</code> .
<i>Expire password on initial login</i>	Specify whether to expire the newly created user’s password on initial login (True), or whether to default to the Identity Vault’s password policy <i>GraceLogin</i> setting.
<i>Display password with attributes</i>	Specify whether to display the password on the same page as the other attributes (True) or on its own page (false).
<i>Create Virtual Entity complex preference</i>	Click <i>View/Edit Custom Preference</i> to access the Entity and Attribute definitions for the create portlet. The preferences are described in Table 11-4 on page 215 .

Table 11-4 *Create Portlet: Complex Preferences*

Preference	Description
<i>Entity Definition</i>	<p>The name of the object type to create. This represents the beginning of an entity definition block where you define how the portlet handles the create operation.</p> <p>Objects listed in the complex preferences are displayed to the user in a drop-down list. To restrict the objects that users can create, remove objects from this preference sheet with the delete button. To add other entities, click <i>Add Entity Definition</i> and complete the wizard.</p>

Preference	Description
<i>Attributes</i>	<p>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 do not save properly if a required attribute is missing.</p> <p>To add or remove an attribute:</p> <ul style="list-style-type: none"> ◆ Click the <i>Modify Attributes</i> button.  <ul style="list-style-type: none"> ◆ 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 the attribute to the <i>Selected</i> list. Do the reverse to remove an attribute. ◆ To reorder the attributes list, click the up and down arrows to the right of the <i>Selected</i> list. Click <i>Submit</i>. <p>Attributes and data types:</p> <p>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 list box.</p> <hr/> <p>NOTE: The create portlet automatically prompts for an object ID. (The label displays as the entity type and appends the string ID, for example, user ID or Group ID.) The object ID is the naming attribute for the object. for the object. You do not have to add the CN as an attribute.</p> <hr/> <p>For more information, see the <i>Novel Identity Manager 3.5 User Application: Design Guide</i>.</p>

Completing the Preferences Panel

To verify that you submitted valid entries, click *Submit*. If an entry is invalid, an error message is displayed at the top of the preferences page. Click *Return to List View* when you are able to click *Submit* and no errors occur. You must click *Save Preferences* when you return to List View.

11.4 Configuring the Create Portlet for Self-Registration

You can configure the Create portlet so that guest users are able to self-register. Enabling anonymous access to the create portlet is a two-step process. First, configure a Create portlet instance for anonymous use, then create a shared page to host the new portlet instance. You have the option to force the newly registered user to log in or to allow anonymous access to other identity self-service features. To create a portlet instance:

- 1 Go to the Portlet Admin page.
- 2 Register and name a new instance of the CreatePortlet, for example, *Self Registration*.
- 3 Select the new portlet instance, then click *Settings*.

4 Set *Require Authentication* to false, then click *Save Settings*.

5 Select *Preferences* and modify the preferences as needed.

For example, you could specify a DetailPortlet that supports anonymous access, or you could limit the set of attributes displayed by the default instance. (The changes you make to the default instance are reflected in other parts of the User Application that use that instance.)

TIP: If you do specify the default DetailPortlet, the user is forced to log in when viewing the detail of the newly created object. For details, see [Section 11.4.1, “Guest Access Required Settings,” on page 217](#)

To create a shared page:

1 Go to the *Page Admin* tab.

2 Create a new page.

3 Under *Assign Categories*, select *Guest Pages*. You can select other categories if you also want logged-in users to see this.

4 Click *Save Page*.

5 Click *Select Content*, add the new instance to the page, then click *Save Contents*.

6 Click *Assign Permissions* and make sure that *View Permissions Set to Admin Only* is unselected.

7 Save the page.

11.4.1 Guest Access Required Settings

Other required settings include:

- ♦ *Create container*: Every entity requires a create container. You can define a default create container for each entity type in the directory abstraction layer, or you can allow the user to select one. When you specify a default create container for the entity type, the user is never prompted for the container. When you do not specify a default, the user must select one. To allow anonymous users access to the selection list, you must change the ContainerLookupPortlet setting *Require Authentication* to false. For more information about the default Create container, see the section on the directory abstraction layer editor in the *Identity Manager 3.5 User Application: Design Guide*.
- ♦ *Identity Vault Rights*: The user is initially the guest user. When he or she self-registers, the User Application writes an object to the create container. To create a user object, the guest user must have create [Entry rights] in the container where new users are created. This could be inherited or restricted by using an inherited rights filter. The guest user must also have Write rights to the attribute(s) that they are allowed to create.
- ♦ *DNLookup controls*: If the user is required to provide a value for an attribute defined as a control type of DNLookup, you need to change the ParamlistPortlet setting *Requires authentication* to false.
- ♦ *Detail portlet*: When the object is successfully created, the portlet displays a link to the object displayed, via the Detail portlet. The default Detail portlet requires authentication so that users are forced to log in with the new identity credentials before they are able to view the detail. You can create a separate instance of the detail portlet for anonymous login, or you can modify the default detail portlet so that *Requires authentication* is set to false. See [Section 12.6, “Setting up Detail for Anonymous Access,” on page 233](#).

- ♦ Passwords: If you allow an anonymous user to create an entity that requires a password, you must ensure that the anonymous account has the rights to create a password.

This section describes the Detail portlet, which lets users display and manipulate an entity's attribute data. The detail portlet is the basis for the *My Profile* action in the Identity Manager User Application's *Identity Self-Service* tab. Topics include:

- ♦ [Section 12.1, “About the Detail portlet,” on page 219](#)
- ♦ [Section 12.2, “Prerequisites,” on page 228](#)
- ♦ [Section 12.3, “Launching Detail from Other Portlets,” on page 229](#)
- ♦ [Section 12.4, “Using Detail on a Page,” on page 230](#)
- ♦ [Section 12.5, “Setting Preferences,” on page 230](#)
- ♦ [Section 12.6, “Setting up Detail for Anonymous Access,” on page 233](#)

12.1 About the Detail portlet

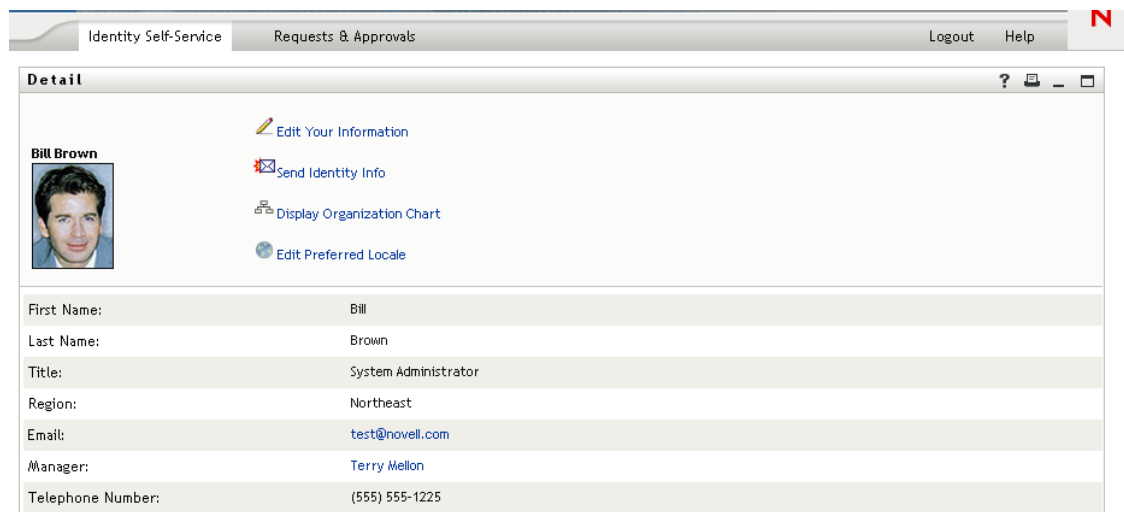
The Detail portlet provides users with a detailed view of an entity's attributes and their values. The portlet has two modes: display and edit. When accessing the Detail portlet, users can take advantage of its built-in capabilities to work with this information, including:

- ♦ [Section 12.1.1, “Displaying Entity Data,” on page 219](#)
- ♦ [Section 12.1.2, “Editing Entity Data,” on page 223](#)
- ♦ [Section 12.1.3, “E-Mailing Entity Data,” on page 225](#) (display mode only)
- ♦ [Section 12.1.4, “Linking to an organization chart,” on page 226](#) (display mode only)
- ♦ [Section 12.1.5, “Linking to Details of Other Entities,” on page 226](#) (display mode only)
- ♦ [Section 12.1.6, “Printing Entity Data,” on page 227](#) (display mode only)
- ♦ [Section 12.1.7, “Setting Preferred Locale,” on page 228](#) (display mode only)

12.1.1 Displaying Entity Data

When accessed, the Detail portlet displays attribute data about a selected entity, such as a user or group. For example, [Figure 12-1](#) displays what the Detail portlet might display when user Bill Brown selects the *My Profile* action.

Figure 12-1 Sample MyProfile Data



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 from that location instead.

For more information, see [“Dynamically loading images.” on page 223](#).

Determining Which Attributes Display

The Detail portlet (display mode) displays the attributes that

- ◆ Your directory abstraction layer data definitions make available for viewing.

For more information on directory abstraction layer configuration, see [Section 1.2.2, “Directory Abstraction Layer,” on page 24](#).

- ◆ Are specified in the *Attributes to display in view mode* preference.

To learn about specifying which attributes display in the Detail portlet, see [Section 12.5, “Setting Preferences,” on page 230](#).

- ◆ 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 12.2.2, “Assigning rights to entities,” on page 229](#).

- ◆ Are currently populated with a value.

Determining How Attributes Display

When displaying attributes, Detail formats the data as text, with some exceptions. Exceptions are listed in [Table 12-1 on page 221](#).

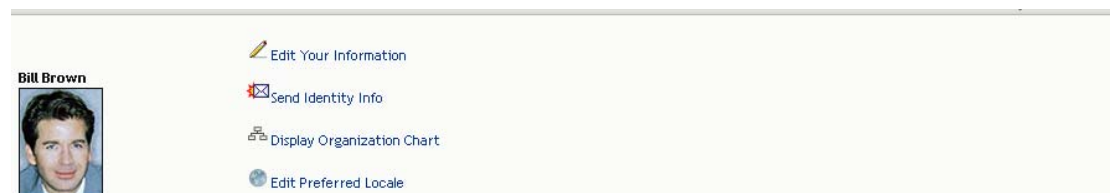
Table 12-1 *Detail Portlet: Attributes That Do Not Display As Text*

Format Specification in Directory Abstraction Layer Definition	How It Displays
Format: email	As a mail-to link
Format:	As an icon that initiates a chat and adds that user
<ul style="list-style-type: none"> ◆ groupwise-im ◆ aol-im ◆ yahoo-im 	
Data type: Binary	As the image
Format: 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	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:	As the display-label rather than the actual (key) value
<ul style="list-style-type: none"> ◆ Local List ◆ Global List 	For example, the EmployeeType attribute displays Full Time instead of the actual value ft.

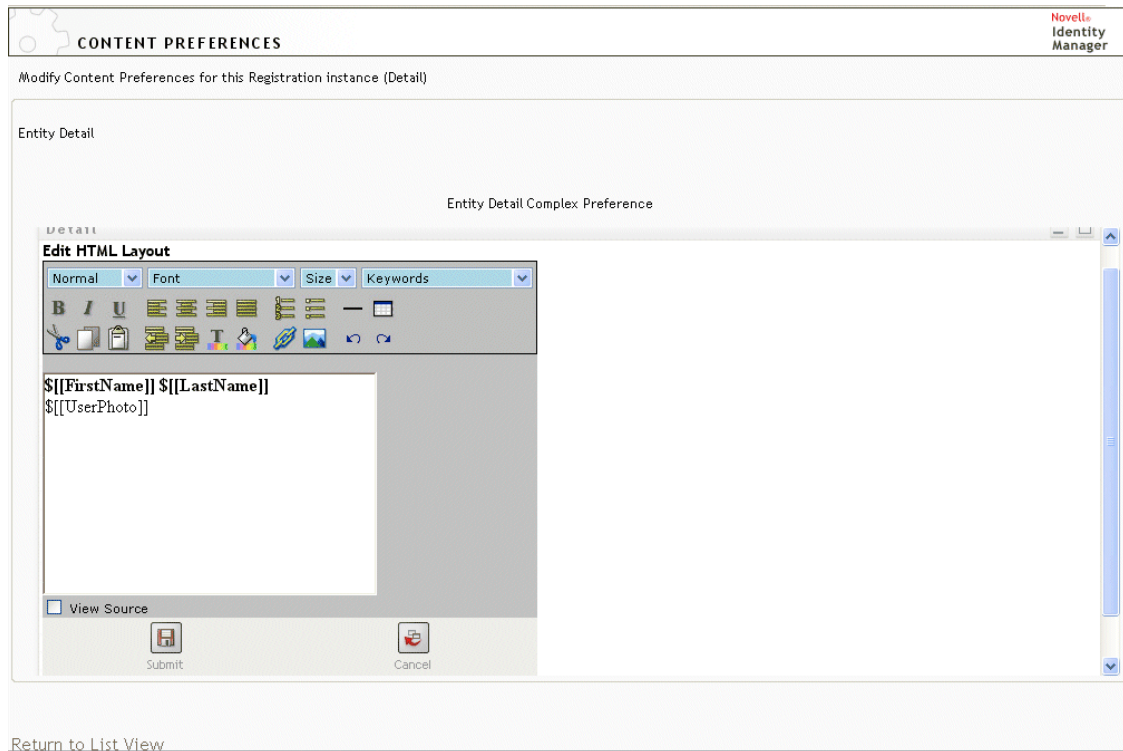
Determining What the Heading Area Displays

You can lay out the heading area of the Detail portlet using standard HTML features.

Figure 12-2 *Detail Portlet: Heading Area*



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 for specifying anchors, images, and so on.

Keywords. When designing your layout, you can use the *Keywords* drop-down list 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 renders the values of the keywords and the quoted text.

NOTE: When manually typing a keyword placeholder instead of selecting it from the dropdown list, make sure that it does not contain HTML formatting. It is recommended that you use the View Source mode for manual entry of keywords. When a keyword is mistyped in a layout, it is rendered as-is at runtime (including the `${[]}`).

Dynamically loading images. To display images that are stored in your Identity Vault (such as user photos), you can add the attribute name using the HTML Layout Editor. For example, adding the User Photo attribute 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 would 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.

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 + ...]]
```

12.1.2 Editing Entity Data

The Detail portlet automatically provides an *Edit* link (such as *Edit Your Information* or *Edit User*) 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:

Figure 12-3 *MyProfile Edit Mode*

Attribute	Value
First Name:*	Bill
Last Name:*	Brown
Title:	System Administrator
Department:	
Region:	Northeast
Email:	test@novell.com
Manager:	Terry Mellon
Group:	Information Technology
Telephone Number:	(555) 555-1225
User Photo:	<input type="radio"/> Hide <input checked="" type="radio"/> Display
<div>Add Image</div> <div></div> <div>Replace or Delete Image</div>	

Save Changes Cancel

NOTE: For Boolean attributes, when both radio buttons are unselected it means that the attribute does not exist for the user. Selecting *true* or *false* creates the attribute for the user and also sets its value.

Determining Which Attributes Display

In edit mode, you can specify the attributes to display and their display order by using the Detail portlet's *Attributes to display in edit mode* preference. In addition, the Detail portlet displays only attributes that

- ◆ Are defined as viewable in the directory abstraction layer data definitions.

For more information on data definitions, see [Section 1.2.2, "Directory Abstraction Layer," on page 24](#).

- ◆ 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 12.2.2, "Assigning rights to entities," on page 229](#).

Determining How Attributes Display

In edit mode, Detail formats each editable attribute as a text box, except in the following cases:

Table 12-2 *Detail Portlet: Recognizing Non-Text-Box Editable Attributes*

Attribute Type Specification (in directory abstraction layer)	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 radio buttons labeled <i>Hide</i> and <i>Display</i>
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: <ul style="list-style-type: none">♦ Local list♦ Global list	As a drop-down list (allowing multiple selections if applicable)

Attributes that can't be edited (either by definition or because of 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. However, such values are flagged as out-of-range, and validation prevents them from being submitted.

12.1.3 E-Mailing Entity Data

The Detail portlet automatically provides a link named *Send Identity Info*. Users can click it to e-mail the URL of the current entity's Detail to one or more other users. By e-mailing the Detail URL rather than the actual information, security is maintained because anyone receiving the URL will need appropriate authority to use it.

12.1.4 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:

Figure 12-4 *My Profile: Linking to Org Chart*



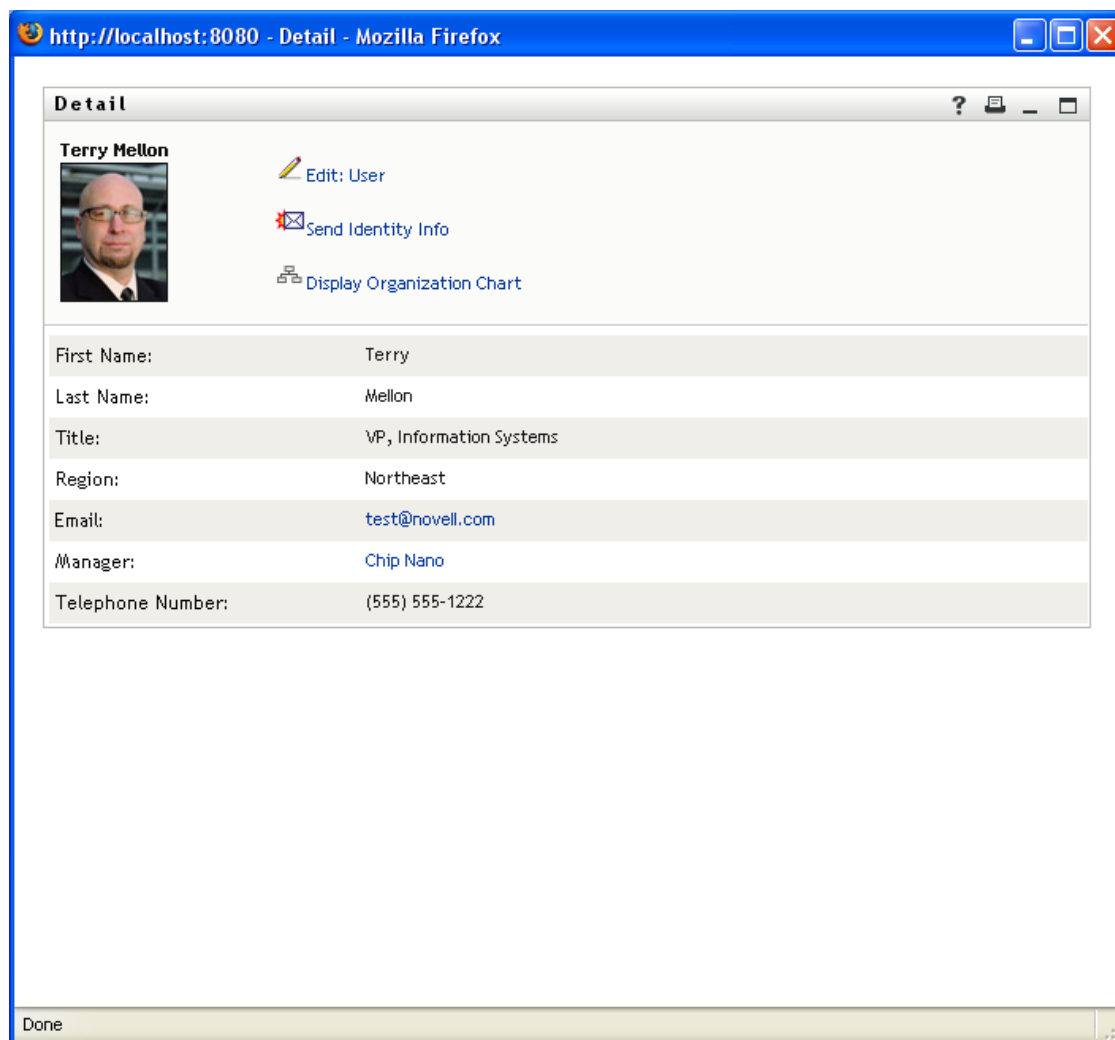
You can suppress automatic linking to the Org Chart by setting Detail's *Enable org chart display* preference to false. See [Section 12.5, "Setting Preferences," on page 230](#).

12.1.5 Linking to Details of Other Entities

When configuring the Detail portlet, you might want to enable users to link to related entities from the current one. You can do that by including attributes that are defined with the control type DNLookup (in your directory abstraction layer).

When the Manager attribute is displayed in a user's Detail, it appears as a link. Clicking that link displays Detail for the Manager.

Figure 12-5 *Linking to Other Entities from My Profile*



For more information on the directory abstraction layer, see [Section 1.2.2, “Directory Abstraction Layer,”](#) on page 24.

To learn about specifying which attributes display in the Detail portlet, see [Section 12.5, “Setting Preferences,”](#) on page 230.

12.1.6 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.

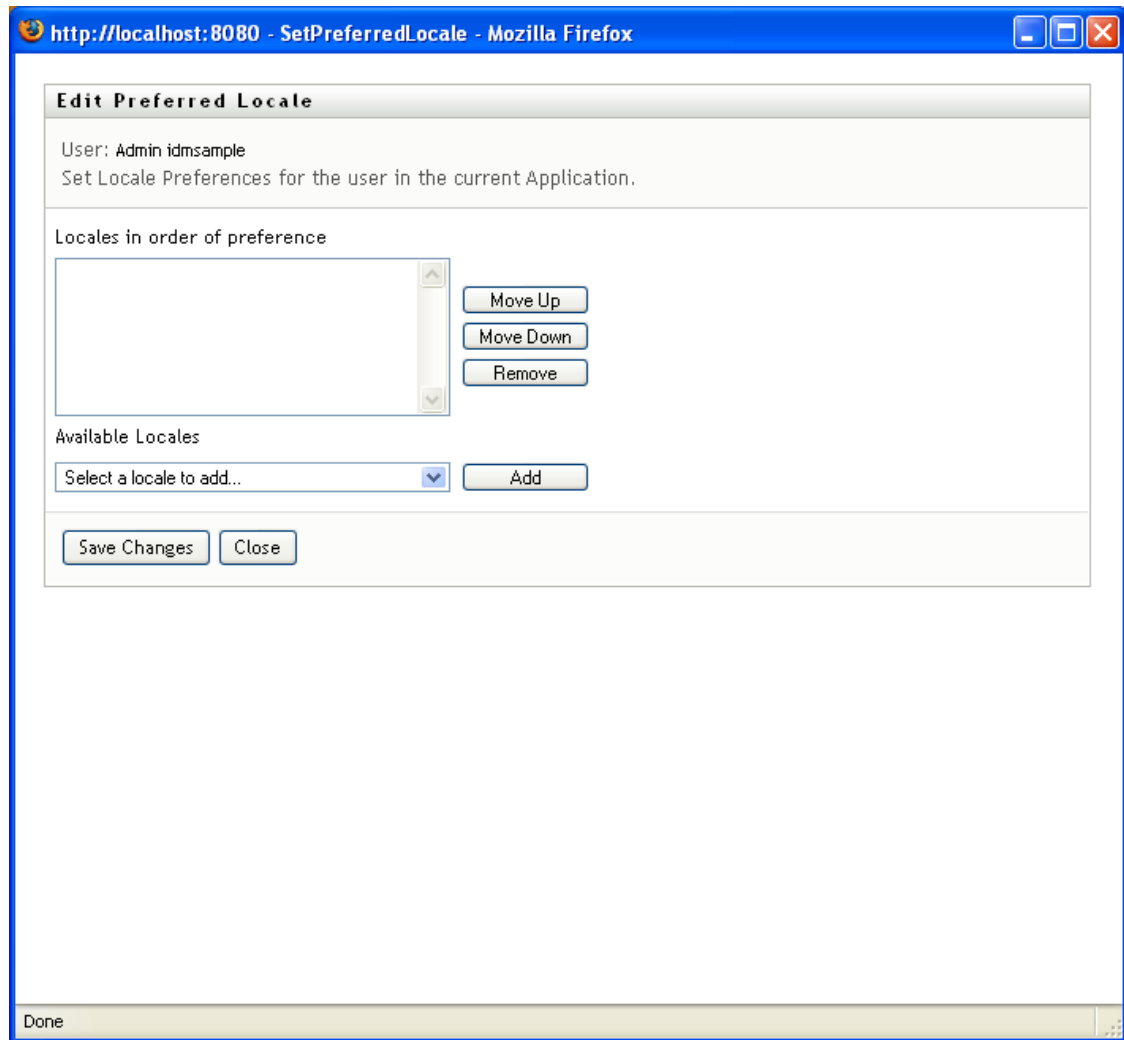
To change this or other settings for the Detail portlet, use the *Administration* tab to update the Portlet Registration for *DetailPortlet* (on the Portlet Administration page).

For more information, see [Chapter 7, “Portlet Administration,”](#) on page 173.

12.1.7 Setting Preferred Locale

The Detail portlet automatically provides a link named *Edit Preferred Locale*. It appears for an administrator or for a user editing their own information. Users can click it to display the settings, and they can use the dialog to change it. Changes to the preferred locale require that the user logout and log back in for the proper locale to display, otherwise, inconsistent locales can be displayed. For example, if you are viewing Detail for user Bill Brown, clicking this link displays:

Figure 12-6 Sample Edit Preferred Locale Dialog



You can suppress the link by setting the *Enable edit of preferred locale* preference to false.

12.2 Prerequisites

Before you start using the Detail portlet, review the following information.

- ♦ [Section 12.2.1, “Configuring the Directory Abstraction Layer,” on page 229](#)
- ♦ [Section 12.2.2, “Assigning rights to entities,” on page 229](#)

12.2.1 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:

- ♦ [Section 12.1.1, “Displaying Entity Data,” on page 219](#)
- ♦ [Section 12.1.2, “Editing Entity Data,” on page 223](#)
- ♦ [Section 12.4, “Using Detail on a Page,” on page 230](#)

For more information on configuration, see [Section 1.2.2, “Directory Abstraction Layer,” on page 24](#).

12.2.2 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). You can also specify the rights to assign for each of the attributes that are available via the Detail portlet.

12.3 Launching Detail from Other Portlets

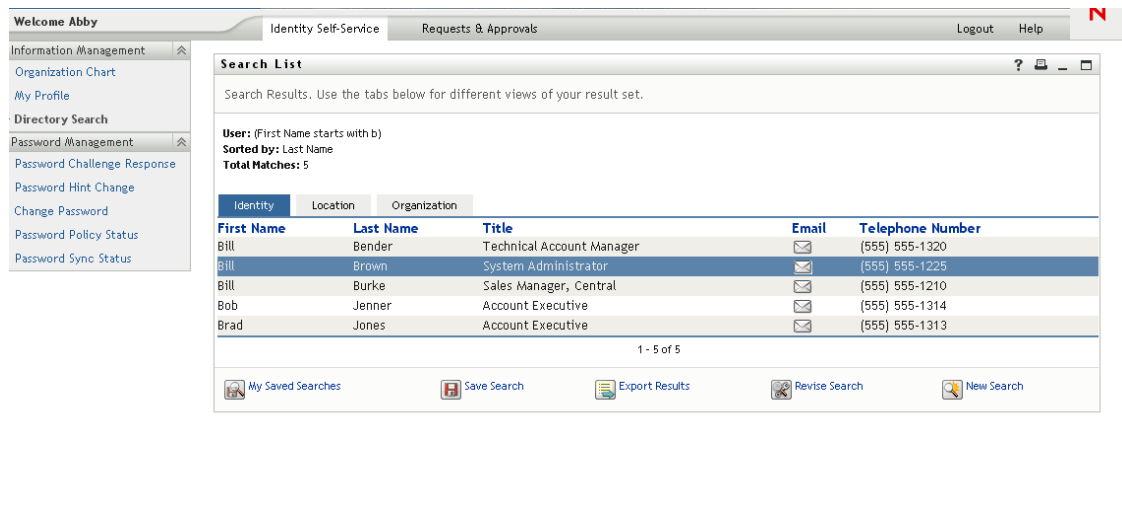
A common use of the Detail portlet is to launch it after selecting an entity from one of the other identity portlets. You can launch Detail from the Search List portlet or from the Org Chart portlet:

- ♦ [Section 12.3.1, “Launching Detail from the Search List Portlet,” on page 229](#)
- ♦ [Section 12.3.2, “From the Org Chart Portlet,” on page 230](#)

12.3.1 Launching Detail from the Search List Portlet

In the Search List portlet, users can click an entity row in the search results in order to display Detail for that entity. For example, clicking the Bill Brown row in the following list displays the Detail portlet with his attribute data:

Figure 12-7 Launching Detail from Directory Search



For more information on the Search List portlet, see [Chapter 15, “Search List Portlet Reference,”](#) on [page 265](#).

12.3.2 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 more information on the Org Chart portlet, see [Chapter 13, “Org Chart Portlet Reference,”](#) on [page 235](#).

12.4 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 page, the Detail portlet automatically accesses the data of the current user.

12.5 Setting Preferences

To define the contents and appearance of the Detail portlet, you set preferences. 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 6, “Page Administration,”](#) on [page 139](#).
- ♦ To learn about accessing portlet preferences for a portlet registration, see [Chapter 7, “Portlet Administration,”](#) on [page 173](#).

12.5.1 About the Preferences

The Detail portlet has two preference pages: one for general preferences (shown in [Figure 12-8](#) on [page 231](#)) and one for complex preferences.

Figure 12-8 Detail Preferences: General Preferences

CONTENT PREFERENCES

Modify Content Preferences for this Registration instance (Detail)

Entity Detail

	Preference	Preference Value		Req.	Read only	Hide
Reset	OrgChart Portlet Name:	<input type="text" value="OrgChartPortlet"/>	Detail	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Reset	Entity Detail Complex Preference:	View/Edit Custom Preference		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Save Preferences

Cancel

Descriptions

Table 12-3 Detail Portlet: General Preferences

Preference	Description
OrgChart Portlet Name	The name of the registered instance of the org chart portlet that you want to launch if the enable org chart display preference is set to true.
Entity Detail Complex Preference	Click View/Edit Custom Preferences to access the detail portlet's complex preferences.

When you open this complex preference, the individual Detail preferences are presented:

Figure 12-9 *Detail Portlet: Complex Preferences*

CONTENT PREFERENCES

Modify Content Preferences for this Registration instance (Detail)

Entity Detail

Entity Detail Complex Preference

Detail

Summary

Entity Definition User

Attributes to display in view mode

First Name

Last Name

Title

Department

Region

Email

Manager

Telephone Number

Attributes to display in edit mode

First Name

Last Name

Title

Department

Region

Email

Manager

Group

[Return to List View](#)

Table 12-4 *Detail Portlet: Complex Preferences*

Preference	Details
Entity Definition	<p>Specifies the attribute list and HTML layout to display when Detail is used for a particular entity type (such as User, Device, or Group).</p> <p>You can click <i>Add Entity Definition</i> to specify Detail support for additional entity types.</p>
Attributes to display in view mode	<p>Specifies which attributes of the selected entity you want the portlet to display in view mode. These attributes are listed in the order you choose.</p> <p>A button is provided to let you add or remove attributes as needed.</p>
Attributes to display in edit mode	<p>Specifies which attributes of the selected entity you want the portlet to display in edit mode. These attributes are listed in the order you choose.</p> <p>A button is provided to let you add or remove attributes as needed.</p>

Preference	Details
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 221 .
Enable edit entity	Choose True if you want to enable the <i>Edit Your Information</i> link in the header of the detail portlet.
Enable send entity info	Choose True if you want to enable the <i>Send Identity Info</i> link in the header of the detail portlet.
Enable org chart display	Choose True if you want to enable the <i>Display Organization Chart</i> link in the header of the detail portlet.
Enable edit of preferred locale	Choose True if you want to display the <i>Edit Preferred Locale</i> link in the header of the detail portlet.

12.6 Setting up Detail for Anonymous Access

An anonymous user might navigate to the Detail portlet after completing the Create portlet or performing a Search. You can set up a special instance of the Detail portlet just for access by an anonymous or guest user. If you do not set up a separate instance for anonymous access, the user might be prompted to log in before being allowed to access any details of an Identity Vault object. As an alternative to setting up a unique instance for guest access, you could also change the authentication requirement of the standard detail portlet

To set up the detail portlet for anonymous access:

- 1 Go to *Administration > Portlet Admin*.
- 2 Register and name a new instance of the DetailPortlet, for example, Public Detail.
- 3 Select the new detail portlet instance.
- 4 Go to *Settings*. Set *Requires authentication* to false.
- 5 Click *Save Settings*.
- 6 Go to *Preferences* and modify the preferences as required. For example, you might want to change the entities or the attributes to display in view and edit mode.

If the anonymous user is allowed to view the detail without logging in, Detail does not display *Edit User* or *Edit Your Information* because the portlet detects that the user is not logged in and has no Edit rights. If the anonymous user is forced to log in, edit rights are determined by any policies set in eDirectory for new users in that container.

This section describes how to modify or add new org chart features to your Identity Manager User Application. Topics include:

- ♦ [Section 13.1, “About Org Chart,” on page 235](#)
- ♦ [Section 13.2, “Configuring the Org Chart Portlet,” on page 240](#)
- ♦ [Section 13.2.2, “Setting Preferences,” on page 241](#)
- ♦ [Section 13.3, “Configuring Org Chart for Guest Access,” on page 262](#)

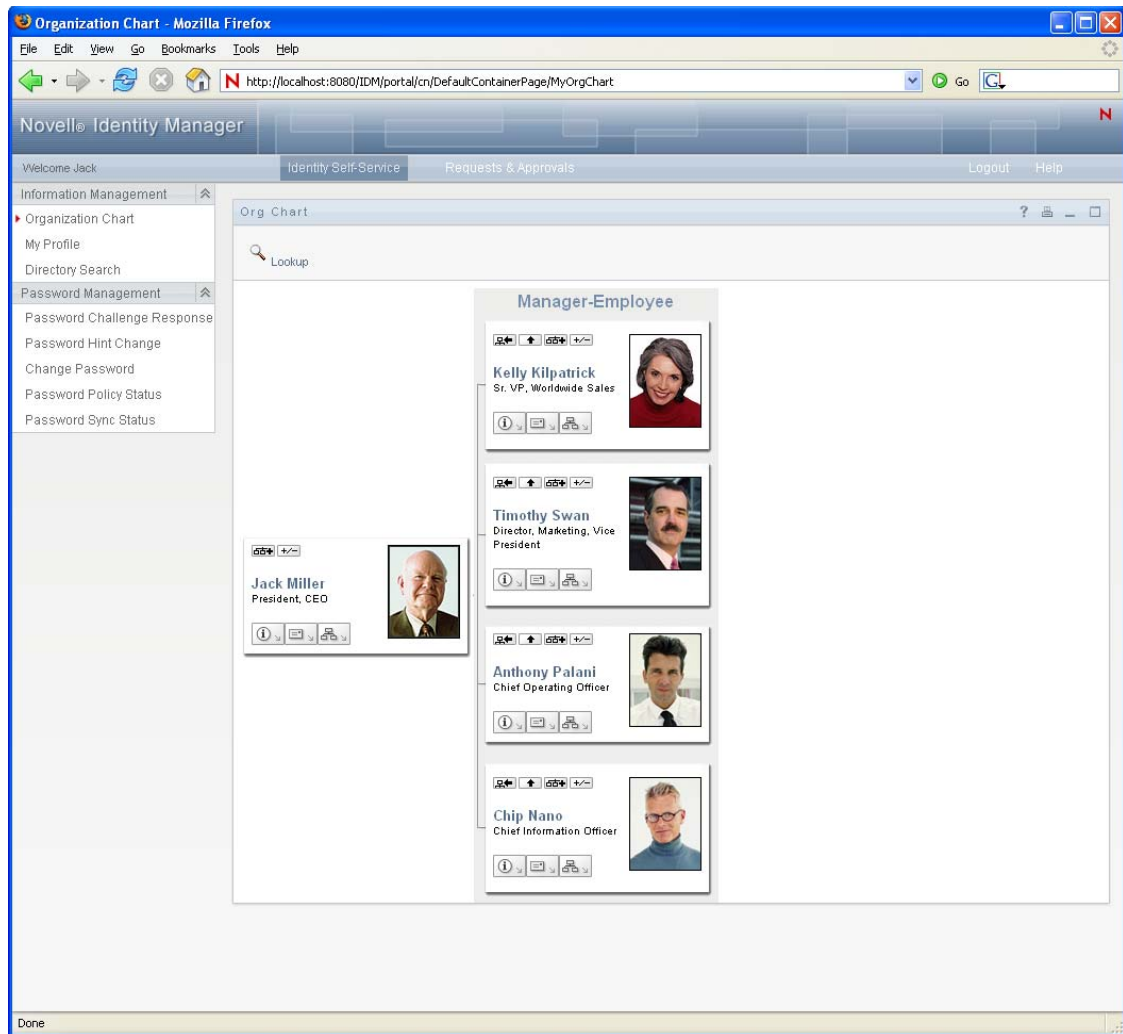
13.1 About Org Chart

The Org Chart portlet allows users to view and browse a graphical representation of the relationships between objects in the Identity Vault. For example, you can define Org Chart portlets that show relationships, such as





- ♦ 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).

Figure 13-1 *Default Org Chart*



Built-in links. The Org Chart portlet includes these built-in links. The built-in links are configurable via the Org Chart Layout Preferences described in [“Org Chart Presentation Layout Preferences” on page 251](#).

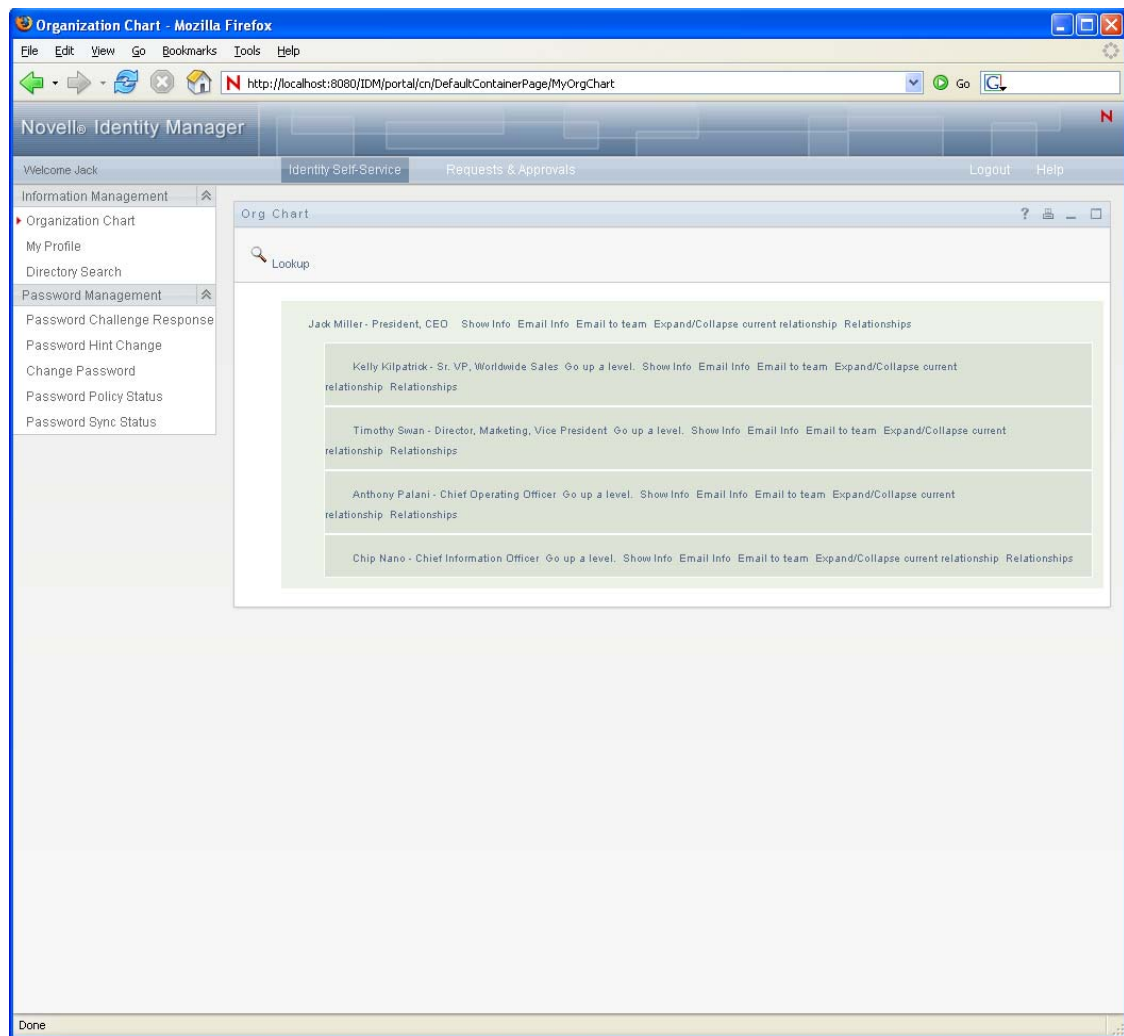
Link	Description
	Allows the user to navigate to the next upper level. This is only available when viewing a relationship where the target and source entities are the same type (such as user). Relationships are defined in the directory abstraction layer editor.
	Lets users expand or collapse the default relationship. The default relationship is defined in the preferences. It is the relationship that is initially displayed.
	Lets users reset the root of the org chart currently displayed. The root is the starting point or orientation point of the org chart.
	Lets users choose a relationship to expand or collapse from a drop-down list. If users choose to expand a relationship, Org Chart allows them to choose which direction to expand it (left or right).

Link	Description
	Launches the Detail portlet.
	<p>Displays a list of org charts. Lets users choose one or more org charts to view.</p> <p>This list of org charts is dynamic. It displays other org charts that share the same source entity type. For example, if you are viewing a manager/employee org chart (the source entity is user) and you click this icon, then the list of org charts you can view only contains relationships where the source entity is also user.</p> <p>.</p>
	<p>Launches an e-mail tool to:</p> <ul style="list-style-type: none"> ♦ Send the identity details of the currently selected user. ♦ Compose an e-mail.
 <u>Lookup</u>	Allows users to perform entity searches. The searches result in the found entity becoming the top node of the chart displayed. (This is not configurable via preferences.)

For more information about adding and restricting the built-in links on your org charts, see “[Org Chart Presentation Layout Preferences](#)” on page 251.

Org Chart also provides a view of the relationships in a 508-compliant format. You can set preferences that display this view by default or as an option. [Figure 13-2](#) shows the same Org Chart data as [Figure 13-1](#) but in the 508-compliant format.

Figure 13-2 *Org Chart Accessible View*



13.1.1 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 [Section 1.2.2, "Directory Abstraction Layer," on page 24](#).

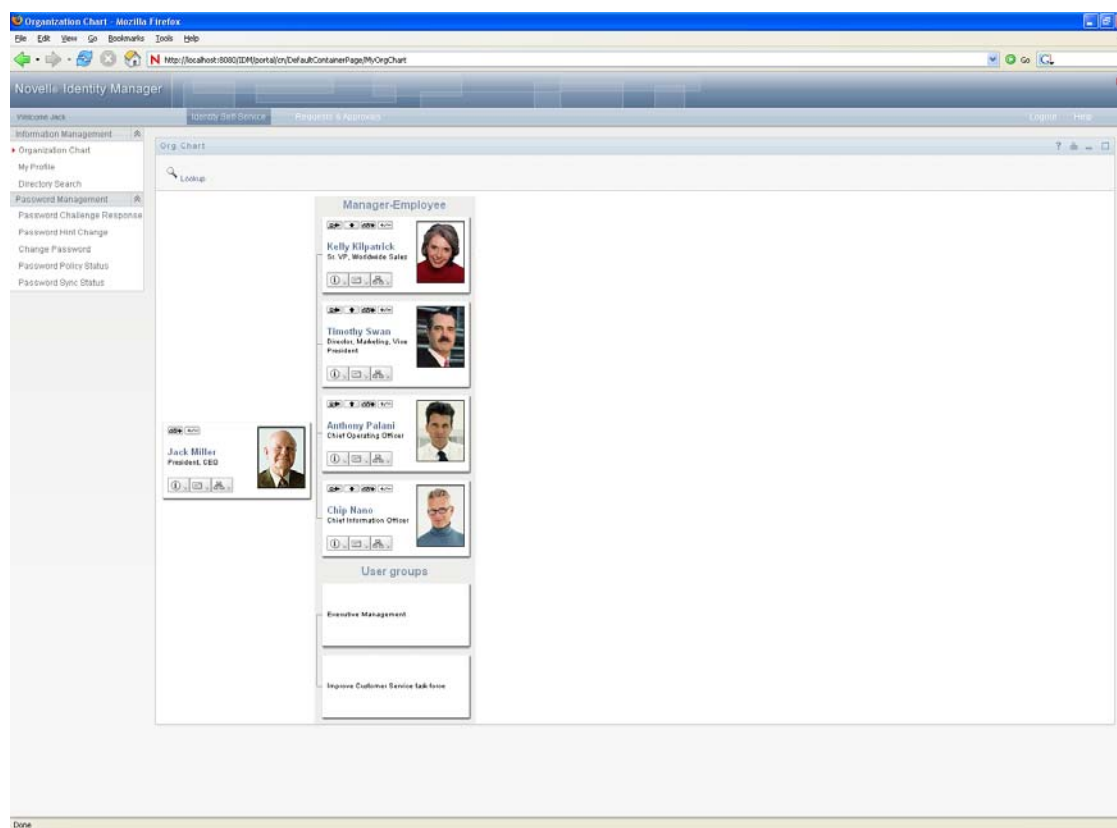
NOTE: Dynamic groups are not fully supported by the Org Chart portlet. You cannot define a dynamic group as the source entity of a relationship, but you can define a dynamic group as the target entity in a relationship.

13.1.2 About Org Chart Display

The Org Chart portlet can display in HTML mode (the default) or in Accessible mode which is the 508-compliant mode. You can enable or disable these views via the portlet preferences. When both modes are enabled, users see a tabbed page. You can control the tab titles through preference definitions.

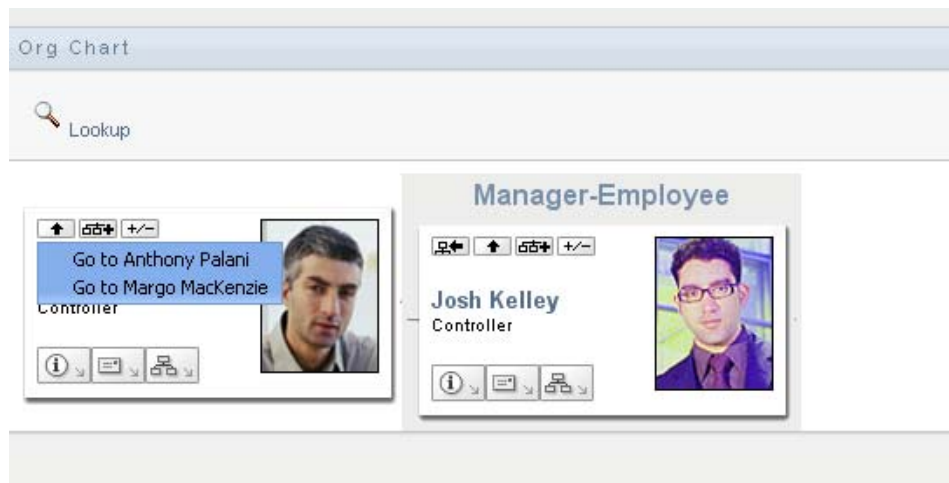
Users are able to display multiple relationships in one org chart as long as the relationships share the source entity. For example, [Figure 13-3](#) shows the org chart with both manager-employees, and users-groups for the root entity.

Figure 13-3 Org Chart Displaying Multiple Relationships



If the manager attribute is multi-valued, the org chart automatically allows users to choose which manager's org chart to display, as shown in [Figure 13-4 on page 240](#).

Figure 13-4 *Displaying Multi-valued Manager Attributes*



User Images

By default, the org chart HTML 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 13.2.3, “Dynamically Loading Images,”](#) on page 261.

13.2 Configuring the Org Chart Portlet

To configure the Org Chart portlet, complete the steps in [Table 13-1](#).

Table 13-1 *Org Chart Portlet: Configuration Steps*

Step	Task	Description
1	Define the relationship that you want to display.	<p>You can use one of the predefined relationships that are installed with the Identity Manager User Application, or you can create your own.</p> <p>For more information about defining a relationship, see Section 1.2.2, “Directory Abstraction Layer,” on page 24.</p>
2	Verify that the entities and attributes that you want to use in the relationship are available in the directory abstraction layer.	<p>For more information about defining a relationship, see Section 13.2.1, “Directory Abstraction Layer Setup,” on page 241.</p>
3	Determine where you want to display this relationship.	<p>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?</p> <p>For more information about creating pages and adding portlets to those pages, see Chapter 6, “Page Administration,” on page 139.</p>

Step	Task	Description
4	Set preferences for the portlet.	<p>Preferences let you define:</p> <ul style="list-style-type: none"> ♦ Which attributes to display. ♦ How to display them (their HTML layout). <p>For more information, see Section 13.2.2, “Setting Preferences,” on page 241.</p>
5	Test.	Test the relationship definitions and layout.
6	Set eDirectory™ rights and establish any indexes needed to enhance performance.	<p>Effective rights. To display attributes defined by the portlet, users must have Read rights to the attributes.</p> <p>Performance enhancement. The performance of the org chart display can be enhanced by adding an eDirectory value index to the relationship’s target attribute because the target attribute is used to do the LDAP search.</p>

13.2.1 Directory Abstraction Layer Setup

The entities and attributes displayed within an Org Chart must be defined in the directory abstraction layer. [Table 13-2 on page 241](#) shows the attributes and properties that you must set for each entity and attribute displayed in an org chart.

Table 13-2 *Org Chart Portlet: Entity and Attribute Settings*

Definition Type	Setting	Value
entity	view	Selected (true)
attribute	read	Selected (true)
	search	Selected (true)

Lookup Link requirements. *Lookup Link* allows users to navigate the org chart by performing searches for other objects of the same type as the Source Entity key. The Lookup Link requires that the source 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 is empty when displayed.

For more information on entity and attribute configuration, see [Section 1.2.2, “Directory Abstraction Layer,” on page 24](#).

13.2.2 Setting Preferences

You can define preferences for the relationships, the presentation (such as attributes and their order) and general display preferences. For more information, see:

- ♦ [“Org Chart General Preferences” on page 242](#)
- ♦ [“Org Chart Data/Relationship Preferences” on page 248](#)

- ♦ “Org Chart Presentation Layout Preferences” on page 251

Org Chart General Preferences

This category includes the preferences on the main preferences page and excludes the custom preferences. The preference page is shown in [Figure 13-5](#) and [Figure 13-6](#).

Figure 13-5 Org Chart Preferences

Modify Content Preferences for this Registration instance (Org Chart)

Entity Org Chart

Preference	Preference Value		Required	Read only						
Data:	View/Edit Custom Preference		<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Enable HTML Pane:	<input checked="" type="radio"/> True <input type="radio"/> False	Detail	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
HTML Pane Title:	<input type="text" value="Standard View"/>	Detail	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Enable Accessible Pane:	<input type="radio"/> True <input checked="" type="radio"/> False	Detail	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Accessible Pane Title:	<input type="text" value="Accessible View"/>	Detail	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Default Pane:	<input type="text" value="HTML Pane"/> <div> <p>Choices</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Display</th> </tr> </thead> <tbody> <tr> <td>HTML</td> <td>HTML Pane</td> </tr> <tr> <td>508</td> <td>Accessible Pane</td> </tr> </tbody> </table> <p>Add</p> </div>	Value	Display	HTML	HTML Pane	508	Accessible Pane	Detail	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Value	Display									
HTML	HTML Pane									
508	Accessible Pane									
Detail portlet name:	<input type="text" value="DetailPortlet"/>	Detail	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Presentation Layouts:	View/Edit Custom Preference		<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Maximum Depth:	<input type="text" value="10"/>	Detail	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Maximum initial depth:	<input type="text" value="3"/>	Detail	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Show Scrollbars:	<input type="radio"/> True <input checked="" type="radio"/> False	Detail	<input checked="" type="checkbox"/>	<input type="checkbox"/>						

Figure 13-6 Org Chart Preferences (continued)

OrgChart Skin: Business Card Detail ☒ ☐

Choices		
Value	Display	
Card	Business Card	Ins Del
eGuide	eGuide	Ins Del
Novell	Novell.com	Ins Del
Wired	Wired	Ins Del
NewBleu	True Blue	Ins Del

[Add](#)

Connect wires to items: ☒ True ☐ False Detail ☒ ☐

Show Relationships: ☒ True ☐ False Detail ☒ ☐

Tree Presentation: 4 [Ins](#) [Del](#) Detail ☒ ☐
[Add](#)

Leaf Presentation: Vertical List of Lines Detail ☒ ☐

Choices		
Value	Display	
0	Vertical List of Place Ca	Ins Del
1	Vertical List of Lines	Ins Del
2	Horizontal List of Place	Ins Del
3	Horizontal List of Lines	Ins Del

[Add](#)

Minimum item width: 220 Detail ☒ ☐

Minimum item height: 100 Detail ☒ ☐

Multi-valued Separator: , Detail ☒ ☐

[Save Preferences](#) [Cancel](#) [Descriptions](#)

Done

Table 13-3 Org Chart Portlet: Preferences

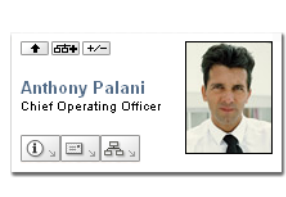
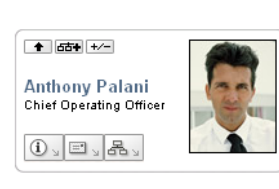
Preference	What to Do
Data	Click <i>View/Edit Custom Preferences</i> to access the preferences that define the org chart's relationships. See "Org Chart Data/Relationship Preferences" on page 248.
Enable HTML Pane	Click <i>True</i> to enable the HTML display of related objects. This is the default display. It displays the related objects as business cards.
HTML Pane Title	Type the text to display in the <i>HTML Pane</i> tab. If you enable the display of the <i>Accessible Pane</i> and the <i>HTML Pane</i> , this text is displayed as the title of the tab containing the HTML display.

Preference	What to Do
Enable Accessible Pane	Click <i>True</i> to enable the Accessible display of related objects. The Accessible pane displays the objects and links as text strings. This display provides 508-compliant access.
Accessible Pane Title	Type the text to display in the Accessible Pane tab. If the HTML Pane and the Accessible Pane are enabled, this text is displayed as the title of the tab containing the Accessible display.
Default Pane	Choose the pane to display as the default when a user clicks the <i>Organization Chart</i> action. It must be enabled.
Detail Portlet Name	Specify the name of the Detail portlet instance to launch when the user clicks the <i>Show Info</i> link.
Presentation Layouts	Click <i>View/Edit Custom Preferences</i> to access the layout preferences. They are described in “Org Chart Presentation Layout Preferences” on page 251 .
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.
Maximum Initial Depth	Defines the depth of the initial display.
Show Scrollbars	Click <i>True</i> to enable scrollbars.

Preference**What to Do**

OrgChart Skin

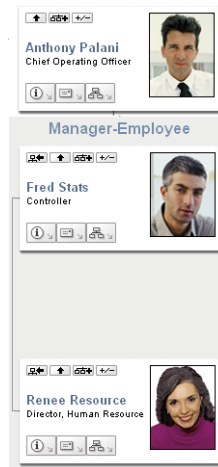
Specify one of the skins for the org chart listed below:

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.

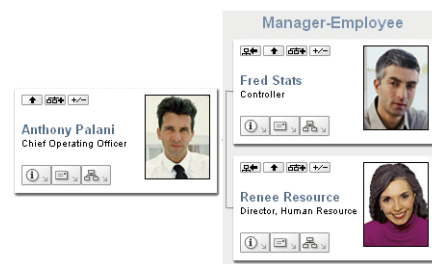
Preference	What to Do
Tree Presentation	<p>Defines the Org Chart orientation (horizontal or vertical) and whether the chart displays as business cards or text. Values range between 0 and 5. Values of 0, 2, and 4 display business cards. Values of 1, 3, and 5 display text.</p> <p>Tree Presentation Values of 0, 2, and 4 display business cards.</p> <p>Specify 0, to place a card above a vertical list of items.</p>



Specify 2, to place a business card above a horizontal list of items.

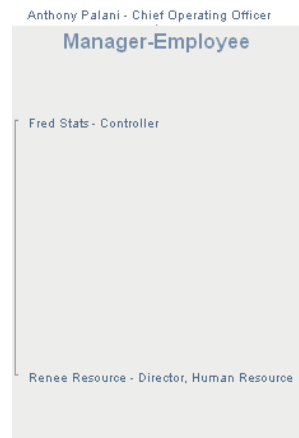


Specify 4, to place card before a vertical list of items



Preference**What to Do**

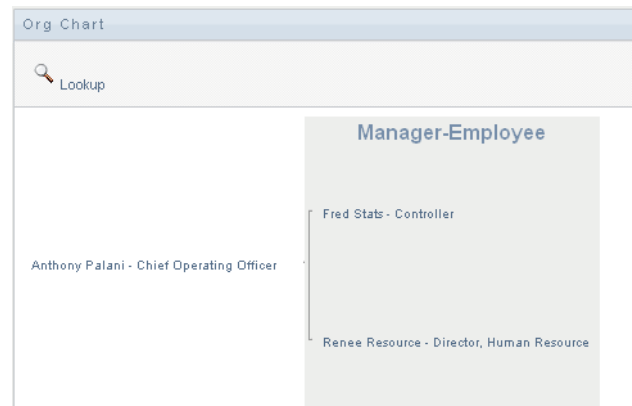
Tree Presentation Values of 1, 3, and 5 display the org chart using lines.
Specify 1, to display a line above a vertical list of items



Specify 3, to display a line above a horizontal list of items



Specify 5, to display a line before a vertical list of items

**Leaf Presentation**

Defines the appearance (orientation and distribution) of the org chart's entity at the maximum depth allowed. For example, if you defined the maximum depth as 10, the leaf presentation controls the display of the entity at the 10th level of the org chart. If you define the maximum depth as 1, this controls the layout of the entity at the 1st level.

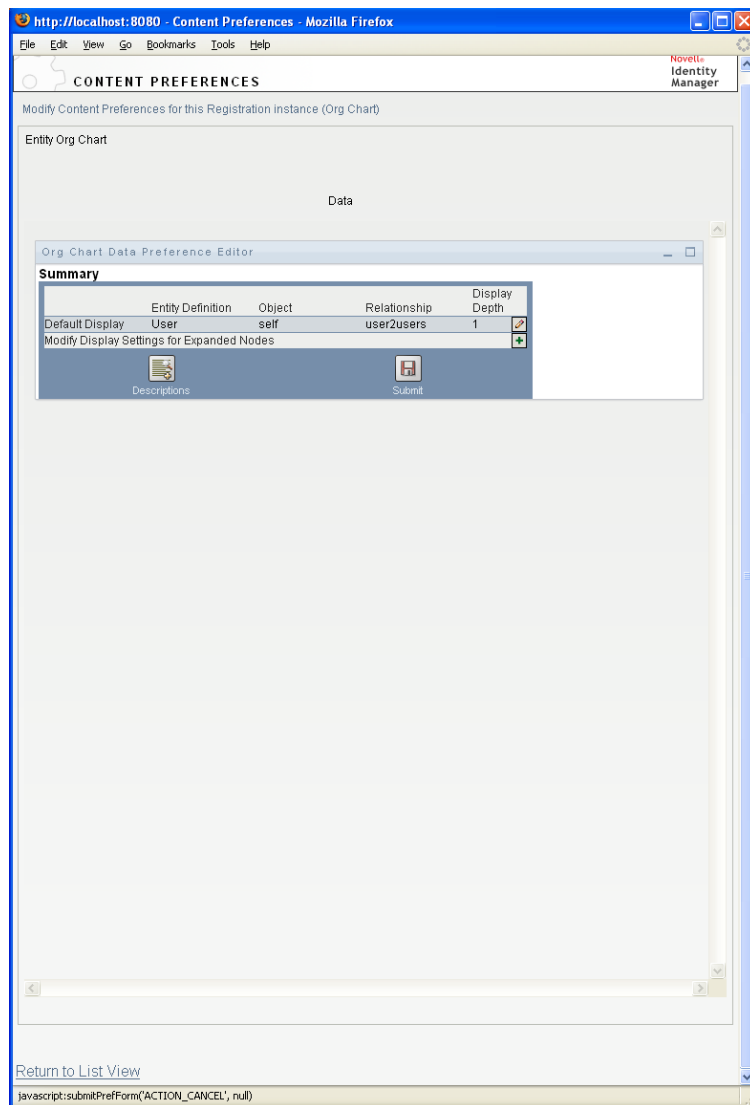
You can display any Leaf Presentation within any Tree Presentation.



Preference	What to Do
Minimum item width	The minimum width (in pixels) of the business card display (in HTML mode). This value should equal to round ('item min height' * 1.618).
Minimum item height	The minimum height (in pixels) of the business card display. 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 Data/Relationship Preferences

You access the Org Chart relationship preferences by clicking the *View/Edit Custom Preferences* link of the *Data* preference. The initial preference page is shown below. It displays the default relationship used in the default Org Chart.

Figure 13-7 Org Chart Data/Relationship Preferences



To edit the entity and relationships available to the org chart, click edit button . See [Editing Data/Relationship Preferences \(page 249\)](#). To modify the display settings for the expanded nodes, click the modify button . See [Modifying Expanded Nodes \(page 250\)](#).

Editing Data/Relationship Preferences


This set of preferences affects the initial display of the org chart and the relationships displayed when users click the expand/collapse relationship button. . You can define any number of relationship levels.

Figure 13-8 Edit Default Data/Relationship Preferences

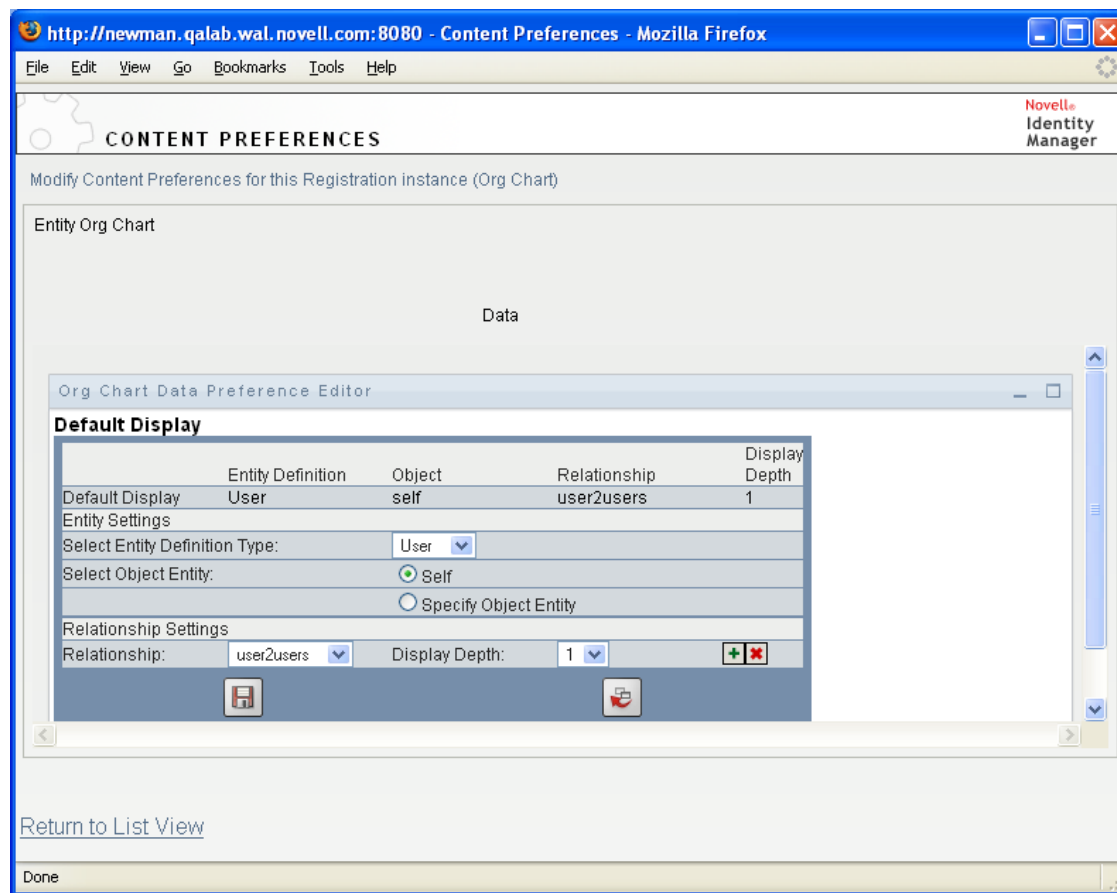



Table 13-4 Org Chart Data/Relationship Preference

Preference	Description
<i>Entity Settings</i>	<p>The <i>Select Entity Definition Type</i> preference lets you choose the entity whose relationships you want to display. Only entities defined in the directory abstraction layer are available in this drop-down list.</p> <p>The <i>Select Object Entity</i> preference lets you choose the chart's root entity. Click the object selector button to search for an object. If the selected entity type definition is a user, then you can select Self instead of an object. Choosing Self means that the org chart root is the logged-on user.</p>

Preference	Description
<i>Relationship Settings</i>	<p>The settings in this category let you specify the details about the relationships displayed by the default chart.</p> <p>The <i>Relationship</i> preference lets you choose a relationship from the drop-down list. Only the relationships that make sense for the selected entity are included in this list.</p> <p>The <i>Display Depth</i> preference controls how many levels of the relationship are displayed. Only display depths allowed for the selected relationship are displayed.</p>

The expanded node preferences are the same, except that they control the relationships displayed after the user clicks the expand/collapse button .

Modifying Expanded Nodes


Expanded nodes preferences let you control what is displayed when the user clicks the expand/collapse button of the org chart. .

Figure 13-9 *Preferences for Modifying Expanded Nodes*

http://localhost:8080 - Content Preferences - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

Novell Identity Manager

CONTENT PREFERENCES

Modify Content Preferences for this Registration instance (Org Chart)

Entity Org Chart

Data

Org Chart Data Preference Editor

Expanded Node

Entity Definition	Object	Relationship	Display Depth
Expanded Node			
Entity Settings			
Select Entity Definition Type:	Group		
Select Object Entity:			
Relationship Settings			
Relationship:		Display Depth:	

Submit Cancel

[Return to List View](#)

javascript:submitPrefForm("ACTION_ADD_EXPANDED_NODE", "");

Org Chart Presentation Layout Preferences

The *Org Chart Presentation Layout* preferences let you define the HTML layout for the display of the org chart entries. You can use the HTML editor available from the preferences sheet, or you can

use the HTML editor of your choice for more precise editing. See [“Using an External HTML Editor” on page 261](#).

The HTML editor, available from the preferences page, 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, specifying anchors and images, and so on. Use the *Keywords* drop-down list to place attributes, commands, and navigation URLs within the layout area. When you choose a keyword from the drop-down list, it is inserted with the proper syntax, but you can also add HTML within the layout area.

Figure 13-10 Org Chart Presentation Layouts Preferences

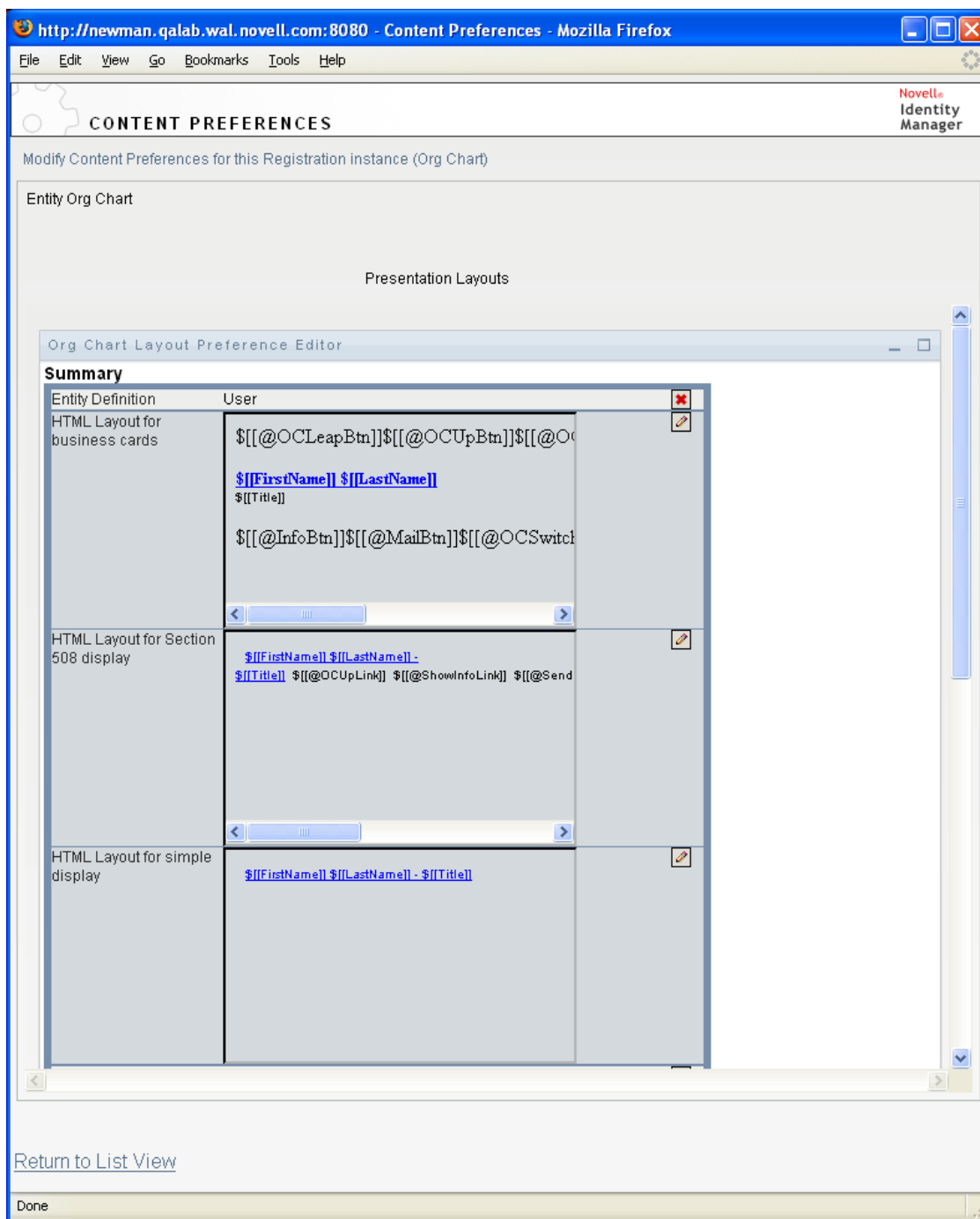


Table 13-5 HTML Layout Definitions

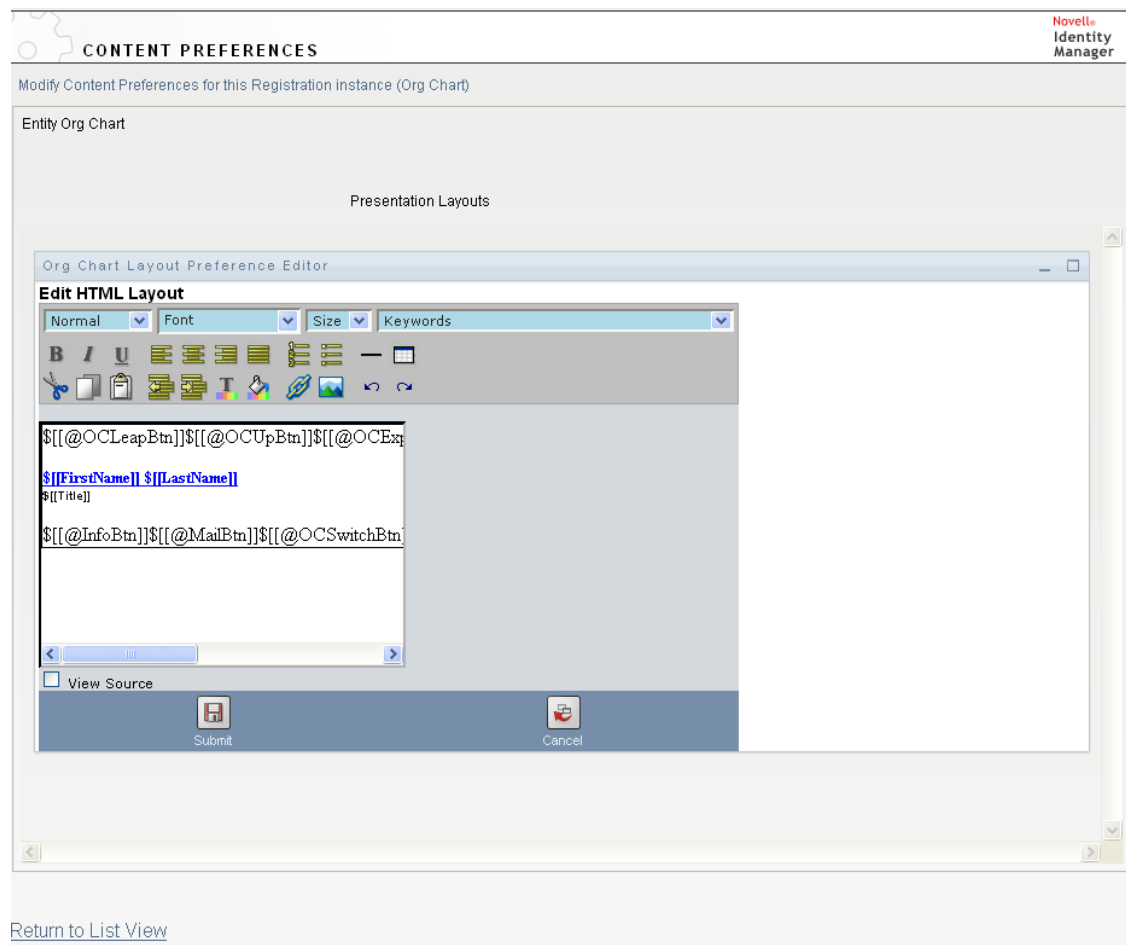
Layout Section	Description
HTML Layout Section Business Cards	The default layout. The layout displayed when Tree Presentation is set to 0, 2, or 4.

Layout Section	Description
HTML Layout Section for Section 508 Display	The default layout for the Accessible Pane.
HTML Layout Section for Simple Layout	The layout when the Tree Presentation is set to 1, 3, or 5.

Using the HTML Editor

You access the HTML editor by clicking the *Edit* button. The HTML editor is shown in [Figure 13-11](#).

Figure 13-11 HTML Editor



HTML Editor Features and Keywords

[Table 13-6](#) describes the HTML editor features and *Keywords* drop-down list. To save your layouts, click *Submit*.

Table 13-6 *HTML Editor Features*

Feature	Tip
Insert Link button	<p>In Mozilla:</p> <ol style="list-style-type: none"> 1. Select the text you want to link, then click <i>Insert Link</i>. 2. Type the URL and click <i>Create Link</i>. 3. Save the preferences. <p>In IE:</p> <ol style="list-style-type: none"> 1. Click <i>Insert Link</i>. 2. Type the URL in the pop-up window. 3. Select the text you want to link, then click <i>Create Link</i> in the pop-up window. 4. Save the preferences. <hr/> <p>NOTE: If your image or URL is located in the upper-left quadrant of the HTML editor, the pop-up window overlaps it. Because the pop-up cannot be moved, you must create the text you want elsewhere in the editor and cut and paste it to the correct location.</p> <hr/>
Add Image button	<p>In Mozilla:</p> <ol style="list-style-type: none"> 1. Place the cursor where you want to insert an image, then click <i>Add Image</i>. 2. Type the URL and text, then click <i>Create Image</i> in the pop-up window. 3. Save the preferences. <p>In IE:</p> <ol style="list-style-type: none"> 1. Click <i>Add Image</i>. 2. Type the URL and text in the pop-up window, place the cursor where you want to insert an image, then click <i>Create Image</i> in the pop-up window. 3. Save the preferences. <hr/> <p>NOTE: If your image or URL is located in the upper-left quadrant of the HTML editor, the pop-up window overlaps it. Because the pop-up cannot be moved, you must create the text you want elsewhere in the editor and cut and paste it to the correct location.</p> <hr/>

Feature	Tip
Keyword drop-down List: Attributes	<p>The set of attributes available for this entity. When designing your layout, you can use the Keywords drop-down list to insert variables that are replaced at runtime with specific attribute values. You can also type the attributes directly in the editor using the following syntax:</p> <pre>\$ [[keyword]]</pre> <p>where <i>keyword</i> is the value of an entity attribute such as <code>LastName</code>.</p> <p>You can concatenate attributes using this syntax:</p> <pre>\$ [[keyword+keyword]]</pre> <pre>\$ [[FirstName+LastName]]</pre> <p>For example, you can concatenate as many attributes as you want and can also include quoted strings like this:</p> <pre>\$ [[keyword+"sample text"+keyword]]</pre> <p>This renders the values of the keywords and the quoted text.</p> <hr/> <p>NOTE: When a keyword is mistyped in a layout, it is rendered as-is in the org chart (including the <code>\$[]</code>).</p> <hr/>
Keyword drop-down List: Commands	<p>These commands allow the Org Chart portlet to display the links or buttons for the built-in links described in "Built-in links." on page 236.</p> <p>The keyword commands generate:</p> <ul style="list-style-type: none"> ♦ Navigation URLs. See Table 13-7, "Org Chart Keywords: Built-in Action URLs," on page 257. ♦ Action Links. See Table 13-8, "Org Chart Keywords: Built-in Action Links," on page 259. ♦ Navigation Buttons. Table 13-9, "Org Chart Buttons Built-in Action Buttons," on page 260. <p>There is a set of commands that generate buttons for the HTML display and a set of commands that generate links for the accessible view. The links do not display with link attributes. See Table 13-8 on page 259.</p>

Table 13-7 *Org Chart Keywords: Built-in Action URLs*

Menu Item	Source Created	Usage
<i>OrgChart Navigation Click (Link)</i>	@OCNavClick	<p>Use this keyword for an onClick event. It makes the clicked entity the new org chart root.</p> <p>To use this keyword:</p> <ol style="list-style-type: none"> 1. Click <i>View Source</i>. 2. Type the @NavClick keyword using this syntax: <pre>\$ [[SomeAttribute]]</pre> <p>where <i>SomeAttribute</i> is an entity attribute that becomes a clickable link.</p> <p>The "javascript:return false;" is required. Omitting it will cause an error.</p>
<i>OrgChart Up Navigation (Link)</i>	@OCUpClick	<p>Use this keyword for an onClick event. It navigates to the current entity's parent. If there is more than one parent, it displays a popup menu with selectable options.</p> <p>To use this keyword, you must:</p> <ol style="list-style-type: none"> 1. Click <i>View Source</i>. 2. Type @OCUpClick using this syntax: <pre>\$ [[SomeAttribute]]</pre> <p>where <i>SomeAttribute</i> is an entity attribute that becomes a clickable link.</p> <p>The "javascript:return false;" is required. Omitting it will cause an error.</p>

Menu Item	Source Created	Usage
	@OCExpCollClick	<p>Use this keyword on an onClick event. It allows the user to Expand/Collapse existing relationships from the clicked entity. To use this keyword, you must:</p> <ol style="list-style-type: none"> 1. Click <i>View Source</i>. 2. Type @OCExpCollClick using this syntax: <pre>\$ [[SomeAttribute]] </pre> <p>where <i>SomeAttribute</i> is an entity attribute that becomes a clickable link.</p> <p>The "javascript:return false;" is required. Omitting it will cause an error.</p>

Menu Item	Source Created	Usage
OrgChart Navigation Url (Link)	@OCNavURL	<p>Specify a URL or entity attribute to display as a link. When clicked, the org chart displays with the clicked entity becoming the root node. This is only valid when the Source and Target entities are the same object type. For example, in the Manager-Employee relationship, both are users.</p> <p>Use this keyword as follows:</p> <ol style="list-style-type: none"> 1. Click <i>View Source</i>. 2. Type the @NavUrl keyword using this syntax: <pre>someText</pre> <p>where <i>someText</i> is the text or an entity attribute. In the following example, <i>Click here</i> becomes a clickable link:</p> <pre>Click here</pre> <p>Here, the <i>FirstName</i> attribute is the clickable link:</p> <pre>\$ [[FirstName]]</pre> <p>With Internet Explorer, do not use the following syntax. IE adds a context before the @NavURL; it will not display correctly.</p> <pre>someText</pre>

The keywords in [Table 13-8](#) generate localized text links for use on the HTML pane.





Table 13-8 Org Chart Keywords: Built-in Action Links





Menu Item	Source Created	Renders as a Localized Link of This Text
Expand/Collapse Current Relationship (Link)	@OCLazyExpCollLink	<p><i>Expand/Collapse current relationship</i></p> <p>Finds the first reentrant relationship and collapses it.</p>

Menu Item	Source Created	Renders as a Localized Link of This Text
<i>Org Chart Up Button (Link)</i>	@OCUpLink	<i>Go up a level</i> Goes to the current entity's parent. If there is more than one parent, it displays a popup that allows the user to select the parent.
<i>Show Info (Link)</i>	@ShowInfoLink	<i>Show info</i> Launches the Detail portlet for the selected entity.
<i>Email Info (Link):</i>	@SendInfoLink	<i>Email Info</i> Launches an e-mail that contains the clicked entity's information.
<i>Email to team (Link)</i>	@MailTeamLink	<i>Email to team</i> Launches an e-mail to the selected entity's team.

The keywords in [Table 13-9](#) generate image buttons for use with the HTML pane.

Table 13-9 *Org Chart Buttons Built-in Action Buttons*

Menu Item	Syntax	Renders As
<i>OrgChart Leap (Action Button)</i>	@OCLeapBtn	 The button makes the clicked entity the new root.
<i>OrgChart Up Button (Action Button)</i>	@OCUpButton	 The button goes to the current entity's parent. If there is more than one parent, it displays a popup that allows the user to select the parent.
<i>Choose relationship to Expand/Collapse (Action Button)</i>	@OCExpColBtn	 This buttons expands/collapses existing relationships from the clicked entity.
<i>Expand/Collapse current relationship (Action Button)</i>	@OCLazyExpColBtn	 This button finds the first reentrant relationship and collapses it.

Menu Item	Syntax	Renders As
<i>OrgChart (Action Button)</i>	@OCSwitchBtn	 <p>This buttons shows the available relationships from the clicked entity. When the user picks one, the clicked entity becomes the new root and the selected relationship is expanded.</p>
<i>Info (Action Button)</i>	@InfoBtn	 <p>Displays the detail portlet for the selected entity.</p>
<i>IM (Action Button)</i>	@IMBtn	 <p>Allows the user to send instant messages and add contacts. The entity must include the appropriate attributes or the org chart displays a message indicating that no data is available.</p>
<i>Mail (Action Button)</i>	@MailBtn	 <p>Launches an e-mail that contains the clicked entity's information.</p>

Using an External HTML Editor

Use the following process to work in an external HTML editor:

- 1 Create the HTML source for the entity attributes, commands, and keywords using *HTML Layout Editor*, available in the preferences.
- 2 Copy the HTML source to the editor of your choice.
- 3 Make the changes that you want.
- 4 Copy the HTML source back to the HTML Layout Editor preference when you have finished editing it.

13.2.3 Dynamically Loading Images

To display images that are stored in your Identity Vault (such as user photos), 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 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 would 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.

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 + ...]]
```

13.3 Configuring Org Chart for Guest Access

To configure the org chart portlet for anonymous access:

- 1 Go to *Administration > Portlet Admin*.
- 2 Register and name a new instance of the OrgChartPortlet, for example, Public OrgChart.
- 3 Select the new instance, then go to the *Settings* tab.
- 4 Set *Requires Authentication* to false, then click *Save Settings*.
- 5 Go to the *Preferences* tab and modify the preferences as needed.
- 6 Reference this instance of Org Chart from the Create or Detail portlets defined for anonymous access.

This section describes how to set up and customize the Resource Request portlet for use with the User Application. It includes these topics:

- ♦ [Section 14.1, “About the Resource Request Portlet,” on page 263](#)
- ♦ [Section 14.2, “Configuring the Resource Request Portlet,” on page 263](#)
- ♦ [Section 14.2.1, “Setting Preferences,” on page 264](#)

14.1 About the Resource Request Portlet

The Resource Request portlet allows the guest or anonymous user to execute resource requests. For example, you could set up a resource request that allows a user to register (which adds an Identity Vault) upon a completed and approved workflow.

14.2 Configuring the Resource Request Portlet

Follow these steps to configure the Resource Request portlet:

Table 14-1 *Resource Request Configuration Steps*

Step	Task	Description
1	Define the anonymous user for your system.	<p>Are you using the LDAP guest, or a special user defined in the Identity Vault? What privileges will this user need in order to execute the workflow? Does the workflow have the correct property attributes set?</p> <p>For more information about the anonymous user, see Chapter 1, “Introduction to the User Application,” on page 21.</p>
2	Specify the resource request to be executed from this portlet.	<p>For more information, see Section 14.2.1, “Setting Preferences,” on page 264.</p>
3	Create a new page to contain the resource request. The security on this page should allow guest or anonymous access.	<p>For more information, see Section 6.3, “Creating and Maintaining Shared Pages,” on page 155.</p> <p>After you create the new shared page, make sure that you specify the Guest Category and deselect the page's <i>View permission Set to Admin only</i>.</p>
4	Test the resource request as the anonymous user.	<p>Verify that the workflow completes as expected.</p>

TIP: When you create the workflows to use with the Resource Request portlet and you define the To token in the e-mail notification as `_default_`, the addressee expression must be an IDVault expression.

14.2.1 Setting Preferences

Preferences include:

Table 14-2 *Resource Request Portlet: General and Custom Preferences*

Preference	Description
Resource Request	<p>Click <i>View/Edit Custom Preference</i> to access the list of resource requests to add to the page. This list is populated with any resource requests deployed to the User Application driver.</p> <p>Choose a single resource request. The list is populated with the resource requests that are deployed to the User Application driver.</p>

This section describes how to set up and customize the Search List portlet for use with the Identity Manager User Application. Topics include:

- ♦ [Section 15.1, “About Search List,” on page 265](#)
- ♦ [Section 15.2, “Configuring the Search List portlet,” on page 269](#)
- ♦ [Section 15.2.2, “Setting Search List preferences,” on page 271](#)
- ♦ [Section 15.3, “Configuring Search List for Anonymous Access,” on page 276](#)

15.1 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 and groups, but you can modify it to change the scope of searchable objects and attributes.

[Figure 15-1 on page 265](#) shows how the Directory Search action allows users to define search criteria.

Figure 15-1 Basic Search

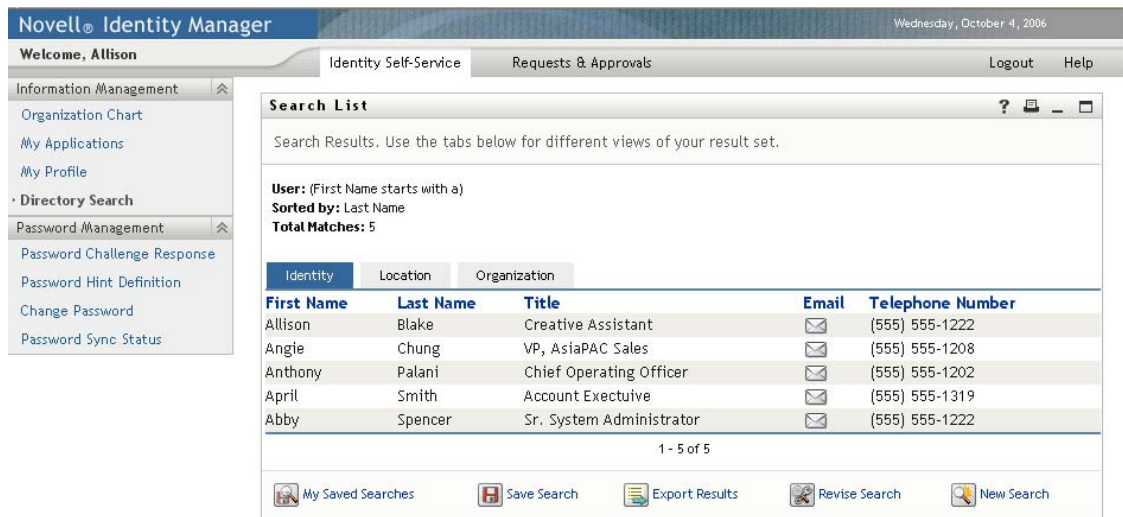
Table 15-1 Directory Search Criteria

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 15.2.2, “Setting Search List preferences,” on page 271 .

User Interface Element	Description
With this criteria	<p>Users define the search criteria by selecting attributes and search operators from the drop-down list.</p> <p>When users select <i>Advanced Search</i>, they are able to specify multiple rows and multiple blocks of search criteria groupings that can be made inclusive (AND) or exclusive (OR).</p> <p>For more information on defining the searchable attributes, see “Setting Search List preferences” on page 271.</p>
Search	<p>Runs the specified search criteria.</p> <p>For more information on defining the default search, see “Setting Search List preferences” on page 271.</p>
My Saved Searches	Allows the user to run, edit, or delete a previously saved search.
 My Saved Searches	
Advanced Search	<p>Lets users 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).</p> <p>For more information on defining the searchable attributes, see “Setting Search List preferences” on page 271.</p>
 Advanced Search	

This example shows how the portlet displays (using sample data) after the search criteria *First name starts with A* is entered:

Figure 15-2 Sample Search List Results








The screenshot shows the Novell Identity Manager interface. The top navigation bar includes 'Welcome, Allison', 'Identity Self-Service', 'Requests & Approvals', 'Logout', and 'Help'. The left sidebar contains a menu with options like 'Information Management', 'Organization Chart', 'My Applications', 'My Profile', 'Directory Search', 'Password Management', 'Password Challenge Response', 'Password Hint Definition', 'Change Password', and 'Password Sync Status'. The main content area displays the 'Search List' portlet. It shows search results for the criteria 'User: (First Name starts with a)' and 'Sorted by: Last Name'. The results are displayed in a table with columns: First Name, Last Name, Title, Email, and Telephone Number. The table lists five users: Allison Blake, Angie Chung, Anthony Palani, April Smith, and Abby Spencer. The bottom of the portlet shows a pagination indicator '1 - 5 of 5' and buttons for 'My Saved Searches', 'Save Search', 'Export Results', 'Revise Search', and 'New Search'.

First Name	Last Name	Title	Email	Telephone Number
Allison	Blake	Creative Assistant	(555) 555-1222	
Angie	Chung	VP, AsiaPAC Sales	(555) 555-1208	
Anthony	Palani	Chief Operating Officer	(555) 555-1202	
April	Smith	Account Executive	(555) 555-1319	
Abby	Spencer	Sr. System Administrator	(555) 555-1222	

You can configure the Search List portlet to use any of the features listed in [Table 15-2 on page 267](#).

Table 15-2 Search List Portlet 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 “About Results List Display Formats” on page 267 .
My Saved Searches	Allows users to select a previously saved search.
 My Saved Searches	
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 <code>srvprvQueryList</code> attribute.
 Save Search	
Export Results	Lets users export the search results to a different format.
 Export Results	
Revise Search	Lets users change the search criteria.
 Revise Search	
New Search	Lets users define a new search.
 New Search	

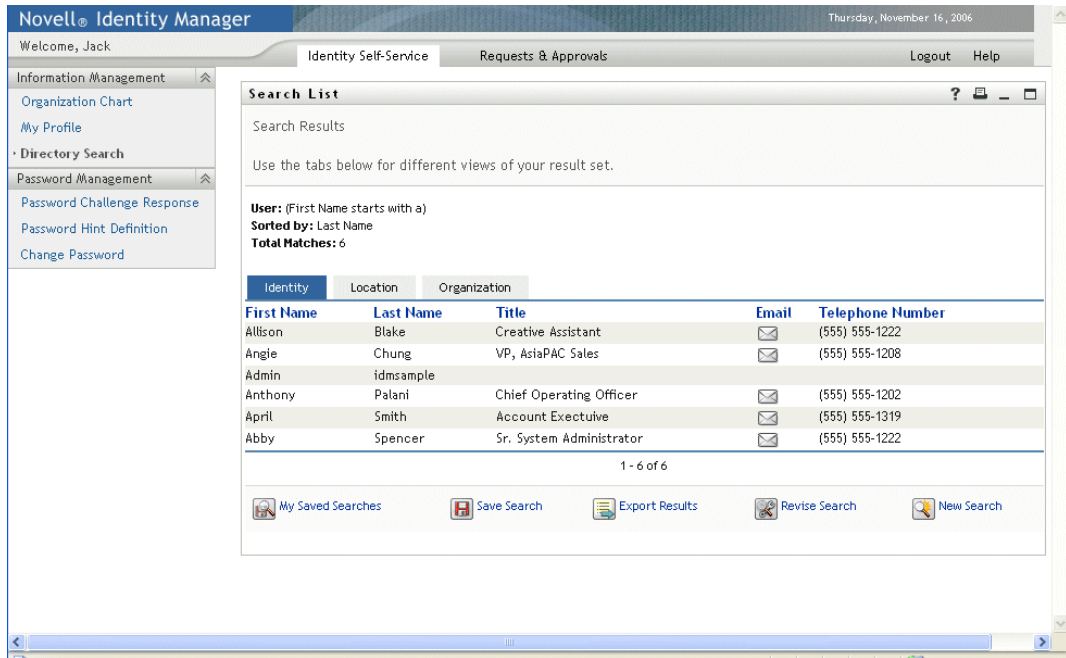
By default, Search List also allows users to:

- ♦ Print the search results
- ♦ Launch e-mail from the results list
- ♦ Launch the Detail portlet from the results list

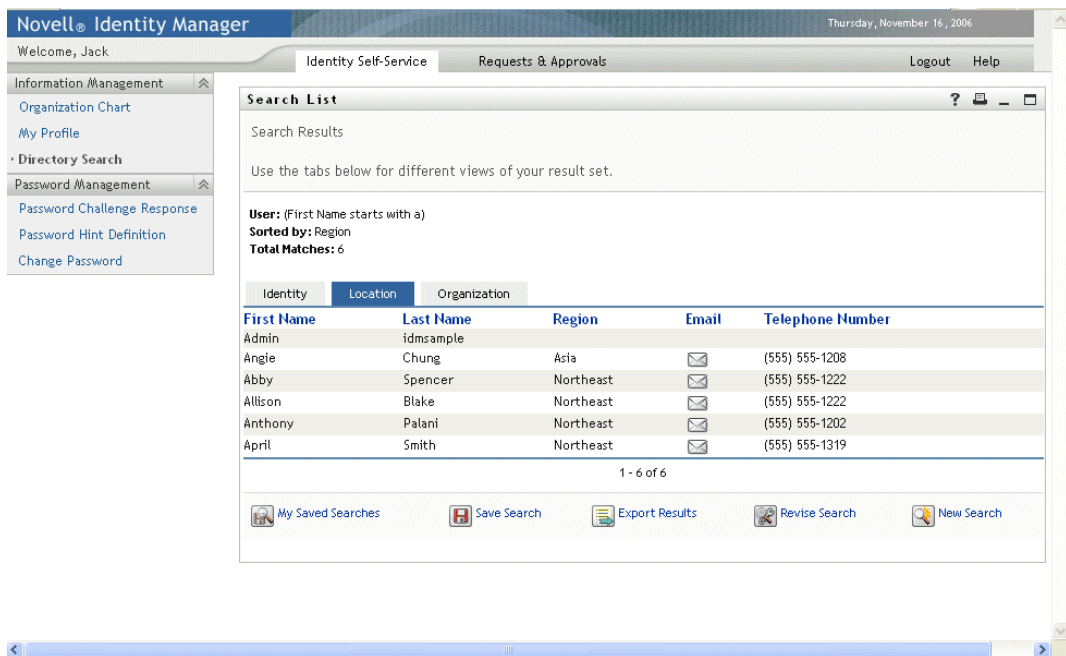
15.1.1 About Results List Display Formats

You can define how data that is returned from the Identity Vault search is displayed to users. The data can be organized in one or more of these page types:

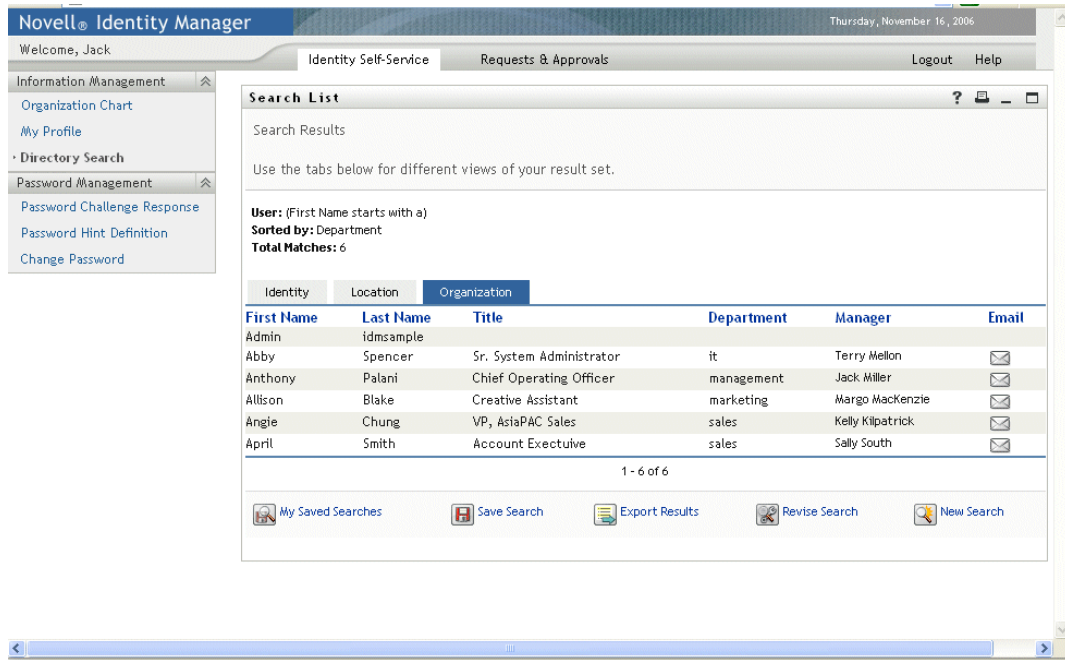
- ♦ Identity Pages typically include contact information, as shown here:



- ◆ Location Pages typically include location information, as shown here:



- ◆ Organization Pages typically include 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 includes information about employee skills, you can set up a results list to display this information.

Depending on how you configure the portlet, 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

15.2 Configuring the Search List portlet

To configure the Search List portlet, follow the steps in [Table 15-3](#).

Table 15-3 Search List Portlet Configuration Steps

Step	Task	Description
1	Define: <ul style="list-style-type: none"> • The entities and attributes you allow users to search. • How you display the results list. 	<p>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.</p> <p>For more information, see “Setting Search List preferences” on page 271.</p>
2	Verify that the set of entities and attributes for searching are defined in the directory abstraction layer.	For more information, see Section 1.2.2, “Directory Abstraction Layer,” on page 24 .

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 6, “Page Administration,” on page 139 .
4	Set preferences for the portlet.	Preferences for the search list portlet let you define: <ul style="list-style-type: none"> ♦ The attributes displayed for each results list format. ♦ The results list display format that a search produces. ♦ The default sort order for the results list formats. For more information, see Section 15.2.2, “Setting Search List preferences,” on page 271 .
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: <ul style="list-style-type: none"> ♦ The user performing the search needs Browse rights to any users or objects being searched. To save a search (for non-Administrative users): <ul style="list-style-type: none"> ♦ <i>Trustee</i> of the organizational unit and the organization where they will be executing the search. ♦ <i>User</i> 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 15.2.2, “Setting Search List preferences,” on page 271](#).

15.2.1 Directory Abstraction Layer Setup

The entities and attributes that can be selected from the search criteria drop-down list and data returned from the Identity Vault searches must be defined in the directory abstraction layer. [Table 15-4](#) shows the properties that you should set for the entities and attributes used by search list.

Table 15-4 *Search List Entities and Attributes*

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). Any attribute that you want to appear in the list of available search criteria must have search=true. When false, you cannot define a search on this attribute or include it in a results list format.
	hide	Unselected (false). Any attribute that you want to include in the results list must have hide=false.

Other Directory abstraction layer settings. The directory abstraction layer data type, format type, filters, and search scope also impact the Search List portlet. The data type and format type affect the appearance; the filter and search scope affect how much data is returned.

For more information, see *Identity Manager User Application: Design Guide*.

15.2.2 Setting Search List preferences

You can define two types of preferences:

- ♦ [“Search preferences” on page 271](#)
- ♦ [“Results List format preferences” on page 273](#)

Search preferences

The search preferences are contained in a single preference page:

Modify Content Preferences for this Registration Instance (Search List)

Search List

Preference	Preference Value	Req.	Read only	Hide								
Reset Default Mode:	<div>My Saved Searches</div> <div> <div>Choices</div> <table border="1"> <thead> <tr> <th>Value</th> <th>Display</th> </tr> </thead> <tbody> <tr> <td>MODE_SIMP</td> <td>Basic Search</td> </tr> <tr> <td>MODE_ADV</td> <td>Advanced Se</td> </tr> <tr> <td>MODE_SAVE</td> <td>My Saved Se</td> </tr> </tbody> </table> <div>Add</div> </div>	Value	Display	MODE_SIMP	Basic Search	MODE_ADV	Advanced Se	MODE_SAVE	My Saved Se	Detail <input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Value	Display											
MODE_SIMP	Basic Search											
MODE_ADV	Advanced Se											
MODE_SAVE	My Saved Se											
Reset Pagination:	<div>10</div> <div> <div>Range</div> <table border="1"> <thead> <tr> <th>Min</th> <th>Max</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table> </div>	Min	Max			Detail <input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Min	Max											
Reset Results Limit:	<div>0</div> <div> <div>Range</div> <table border="1"> <thead> <tr> <th>Min</th> <th>Max</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table> </div>	Min	Max			Detail <input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Min	Max											
Reset Search and List complex preference:	View/Edit Custom Preference	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>								

The search preferences are defined in [Table 15-5 on page 273](#).

Table 15-5 Search List Portlet Preferences

Preference	What to Do
Default Mode	<p>Specify how you want the portlet to display when a user first accesses it. Values are:</p> <p><i>Basic Search.</i> Allows users to enter a single search criteria. For example:</p> <p>First Name starts with A</p> <p><i>Advanced search.</i> Allows users to define multiple search criteria in one or more search blocks. Users can use the <code>and</code> and <code>or</code> logical operators within the search criteria or among the search blocks. For example, users can create a search like this:</p> <p>(First Name starts with A or First Name starts with B) and (Region = Northeast or Region = Southeast)</p> <p>OR</p> <p>(First Name starts with A and Last Name starts with B) or (First Name starts with B and Last Name starts with A)</p> <p><i>My Saved Searches.</i> Displays a list of searches saved by the currently logged in user. The searches are saved in the user's <code>srprvQueryList</code> attribute.</p> <hr/> <p>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.</p>
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	<p>Click to refine the</p> <ul style="list-style-type: none"> ◆ Entities to search ◆ Result set type ◆ Attributes to include in the pages and the order in which they appear

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:

CONTENT PREFERENCES

Modify Content Preferences for this Registration instance (Search List)

Search List

Search and List complex preference

Search List

Summary

Entity Definition	User		
Show Email as icon	<input checked="" type="radio"/> true <input type="radio"/> false		
Result List Types	default		
Identity	<input checked="" type="radio"/> sort		
Attributes	First Name	<input type="radio"/>	
	Last Name	<input checked="" type="radio"/>	
	Title	<input type="radio"/>	
	Email	<input type="radio"/>	
	Telephone Number	<input type="radio"/>	
Location	<input type="radio"/> sort		
Attributes	First Name	<input type="radio"/>	
	Last Name	<input type="radio"/>	
	Region	<input checked="" type="radio"/>	
	Email	<input type="radio"/>	
	Telephone Number	<input type="radio"/>	
Organization	<input type="radio"/> sort		

[Return to List View](#)

The complex preferences are listed in [Table 15-6 on page 275](#).

Table 15-6 Search List Portlet: Complex Preferences

Preference	What to Do
Entity Definition	<p>Each object that is valid for searching (view=true) has a corresponding Entity Definition block on this preferences page. Use these preferences to:</p> <ul style="list-style-type: none"> ◆ 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 <i>Delete</i>, shown on the Entity Definition line. This deletes the entire entity definition block. <p>You can add the object back to the search later by clicking <i>Add Entity Definition</i> (located at the bottom of the page) and completing the wizard selection panels.</p> <hr/> <p>TIP: If an object does not appear in this list, but is listed in the directory abstraction layer, check the <i>view</i> modifier (on the entity object). If it is set to false, then the entity cannot be used by the identity portlets.</p>
Show email as Icon	<p>When set to <i>True</i> and an e-mail attribute is specified in the results list, it displays as an icon. When set to <i>False</i>, the e-mail attribute displays the full e-mail address. The e-mail attribute (whether text or icon) is a clickable <i>mailto:</i> link.</p>
Results List Types (default)	<p>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.</p>
Results List display format block	<p>Specifies the display format (such as Identity, Location, or Organizational pages) and includes the set of attributes to include for the type.</p> <p>To remove a Results List Type:</p> <ul style="list-style-type: none"> ◆ Click <i>Delete</i> next to the <i>Results List Type</i>. <p>This deletes the page type and all of its associated attributes from the search.</p> <p>To add a result set page:</p> <ul style="list-style-type: none"> ◆ Click <i>Expand</i> and select the result set format from the list of choices.

Preference	What to Do
Attributes	<p>Specifies the set of attributes that will be displayed for the particular display format.</p> <p>To add or remove an attribute:</p> <ul style="list-style-type: none"> Click the <i>Modify attributes</i> 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 <i>Submit</i>. <p>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 drop-down 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 are 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.</p> <p>For more information, see Section 1.2.2, "Directory Abstraction Layer," on page 24.</p>
Results List display format block Sort	<p>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.</p> <p>Multi-valued attributes and single-valued attributes. The number of records displayed in a results list varies depending on whether the sort attribute is single- or multi-valued. Sorting on multi-value attributes generally appears 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.</p>

Completing the Preferences Panel

To verify that you have submitted valid entries, click *Submit*. If an entry is invalid, you will see an error message displayed at the top of the preferences page. When you are able to resolve all of the errors, click *Return to List View*, then click *Save Preferences*.

15.3 Configuring Search List for Anonymous Access

To set up the Search List portlet for anonymous access:

- 1 Go to *Administration > Portlet Admin*.
- 2 Register and name a new instance of the Search List portlet, for example, Public Search.
- 3 Select the new instance and go to *Settings*.

- 4 Set *Requires Authentication* to false, then click *Save Settings*.
- 5 Go to *Preferences*, then
 - ♦ Change *Default Search Mode* to *Basic* or *Advanced* (Saved Search mode is not valid for an anonymous user).
 - ♦ Consider specifying a Detail Portlet instance that is also set up for public access (*Requires Authentication* is set to false). If you use the default DetailPortlet, the user will be forced to log in when viewing the detail of any result list link.
 - ♦ Go to *View/Edit Custom preferences* and remove any entities or attributes that you do not want the guest user to see.

To create a new shared page for the anonymous Search List:

- 1 Go to *Administration > Page Admin*.
- 2 Create a new Page and add it to the Guest Pages category (and any other categories for logged-in users.)
- 3 Click *Add Permissions*. Deselect *View Permissions set to admin only*.
- 4 Save the page.

If the Search List portlet instance requires a DNLookup attribute, you need to change the ParamListPortlet setting *Requires Authentication* to false.

Configuring and Managing Provisioning Workflows



These sections describe how to configure and manage provisioning requests, provisioning workflows, and provisioning teams:

- ♦ [Chapter 16, “Configuring the User Application Driver to Start Workflows,” on page 281](#)
- ♦ [Chapter 17, “Configuring Provisioning Request Definitions,” on page 295](#)
- ♦ [Chapter 18, “Managing Provisioning Workflows,” on page 325](#)
- ♦ [Chapter 19, “Configuring Provisioning Teams,” on page 347](#)

Configuring the User Application Driver to Start Workflows

16

This section describes the User Application driver and how to configure it to automatically trigger a workflow based on an event in the Identity Vault.

- ♦ [Section 16.1, “About the User Application Driver,” on page 281](#)
- ♦ [Section 16.2, “Setting Up Workflows to Start Automatically,” on page 282](#)

16.1 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. Although 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 configuration data for setting up the new driver. It is installed in the `DirXML.Drivers` directory, which is `\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. Before you can run the Identity Manager 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 might be required, or you might 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.

16.2 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.

16.2.1 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.

This section provides an example of starting a workflow based on events in the Identity Vault. Although any of the policies can be used to trigger a workflow, the example presented in this section 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 is 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 might cause the workflow process to fail.

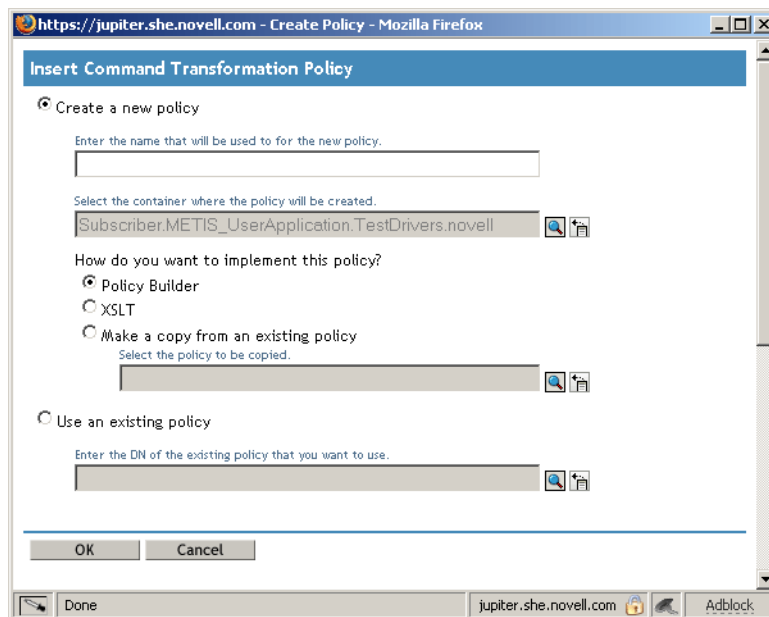
An empty Schema Mapping Policy is also created. You can use this policy as a starting point for triggering a workflow, based on events in the Identity Vault.

16.2.2 Using the Policy Builder

The easiest way to automatically start a workflow based on an Identity Vault event is to use the Policy Builder. The Policy Builder provides a Start Workflow action that simplifies the process of setting up a workflow to start automatically.

- 1 In iManager, expand the *Identity Manager Role*, then click *Identity Manager Overview*.
- 2 Specify a driver set.
- 3 Click the driver for which you want to manage policies. The *Identity Manager Driver Overview* opens.
- 4 Click the policy that you want to edit.

- 5 Click *Insert* to open the Policy Builder.



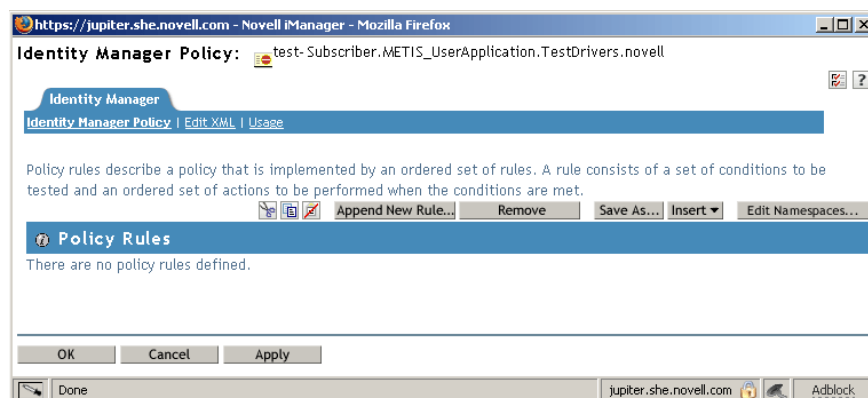
- 6 Click *Create a new policy*.

- 7 Type a name for the policy.

- 8 Click *Policy Builder*.

- 9 Click *OK*.

iManager displays a screen that lists defined policy rules.



- 10 Click *Append New Rule*.

iManager displays the *Rule Builder*.

- 11 Type a *Description* for the rule.
- 12 Select *operation attribute* for the *If* condition in *Condition Group 1*.

- 13 Use the *Browse attributes* button for the *Enter name* field to specify the Identity Vault attribute that you want to use to start the workflow.
For example, to start a workflow when a telephone number changes, select the *Telephone Number* attribute.

- 14 Use the *Select Operator* list to select the operator to use to test the specified attribute.
For example, to start a workflow when a telephone number changes, select *changing*.

- 15 Select *start workflow* from the *Action* list.

- 16 Use the Object Selector in the *Enter provisioning request DN* field to select the provisioning request definition that you want to be executed when the *if* condition is true.

The *Enter user application URL* and *Enter authorized user DN* fields are filled in automatically.

- 17 Type the password for the User Application administrator in the *Enter authorized user password* field.

We recommend using a named password, because typing a password in clear text is a security risk. See “Named Password” in the *Policies in iManager in Identity Manager 3.5* (http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_imanager/data/bookinfo.html) guide.

- 18 In the *Enter recipient DN* field, specify the DN of the recipient of the workflow in LDAP format.

The expression for the recipient DN must evaluate to a DN that conforms to RFC 2253 format (in other words, cn=user,ou=organizational unit,o=organization). For example, you can click the *Argument Builder* button in the *Enter recipient DN* field to create the following expression to pass the recipient’s DN to the workflow:

```
Parse DN("qualified-slash", "ldap", XPath("@qualified-src-dn"))
```

The screenshot shows the 'Append Condition Group' section with 'Condition Group 1' selected. Below it, a condition is defined: 'If operation attribute' with 'Enter name:' set to 'Telephone Number' and 'Select operator:' set to 'changing'. The 'Actions' section shows an 'Action List' with a single action 'Do start workflow'. The 'Enter provisioning request DN:' field contains a long DN string: 'CN=TestSingleApprovalTA,CN=RequestDefs,CN=AppConfig,CN=METIS_UserApplic'. Other fields include 'Enter user application URL:' (http://164.99.26.207:8080/IDMPROV), 'Enter authorized user DN:' (CN=admin,OU=idm-metis,O=novell), 'Enter authorized user password:' (test), and 'Enter recipient DN:' (Parse DN("qualified-slash", "ldap", XPath("@qualified-src-dn"))).

- 19 Specify the arguments for the workflow in the *Enter additional arguments* field.

You must use this field to specify the *reason* attribute, which is required by the workflow. You can click the *String Builder* button in the *Enter additional arguments* field to specify the *reason* attribute and create a value for the attribute (for example, “the recipient’s telephone number has changed”).

The screenshot shows the 'String Builder' dialog box. It has a title bar 'https://jupiter.she.novell.com - Z:String Builder - Mozilla Firefox'. The main area is titled 'String Builder' and contains the text 'String elements provide values for arguments.' Below this is a 'Strings' section with a list of strings. One string is defined: 'Name: reason' and 'String value: The recipient's telephone number changed'. There are buttons for 'Append New String' and 'Remove'. At the bottom are 'OK' and 'Cancel' buttons. The status bar at the very bottom shows 'Done', 'jupiter.she.novell.com', and 'AdBlock'.

- 20 Click *OK* to close the Rule Builder.
- 21 Click *OK* to close the Policy Builder.
- 22 Click *OK* to close the Policies screen.
- 23 Make sure that you add any attributes needed by the workflow to the filter.

In the example described in this procedure, you would need to add *Telephone Number* and *CN* to the filter. For information about adding objects to the filter, see “Controlling the Flow of Objects with the Filter” in the [Policies in iManager in Identity Manager 3.5](http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_imanager/data/bookinfo.html) (http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_imanager/data/bookinfo.html) guide.

16.2.3 Using the Schema Mapping Policy Editor

The Schema Mapping Policy Editor provides an alternative method of starting a workflow automatically, by mapping Identity Vault attributes to workflow runtime data. To get you started, the User Application driver provides an empty policy to edit. Workflow runtime data is available from the workflow definition template described in [Chapter 17, “Configuring Provisioning Request Definitions,” on page 295](#).

When a workflow is created, the following global attributes are created in the Identity Vault:

- ♦ `<workflowName>_StartWorkflow`. This attribute starts a workflow.

- ♦ `<workflowName>_recipient`. This attribute accepts runtime data needed by the workflow from the Identity Vault.
- ♦ `<workflowName>_reason`. This attribute accepts runtime data needed by the workflow from the Identity Vault.

Two other attributes always exist and accept runtime data needed by the workflow from the Identity Vault:

- ♦ AllWorkflows:reason
- ♦ AllWorkflows:recipient

Ensure you have the following information before you set up a workflow to start based on an event in the Identity Vault:

- ♦ The name of the Identity Vault attribute that you want to use as a trigger for the workflow
- ♦ 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.



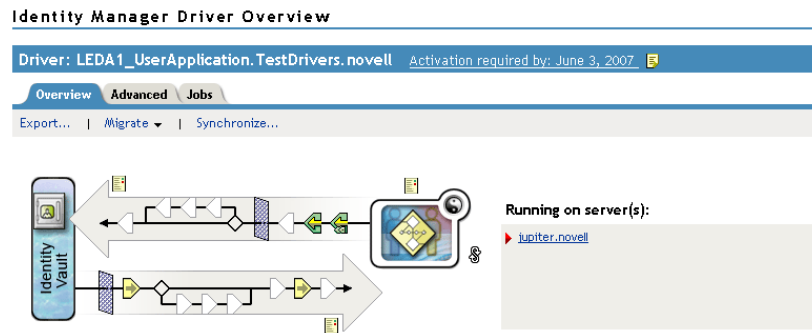
The Identity Manager Overview page displays. This page prompts you to select a driver set.

- 2 Click *Search Entire Tree*; then click *Search*. The Identity Manager Overview page displays, with a graphic that depicts the drivers in the currently selected driver set.
- 3 Click the large driver icon for the User Application driver:

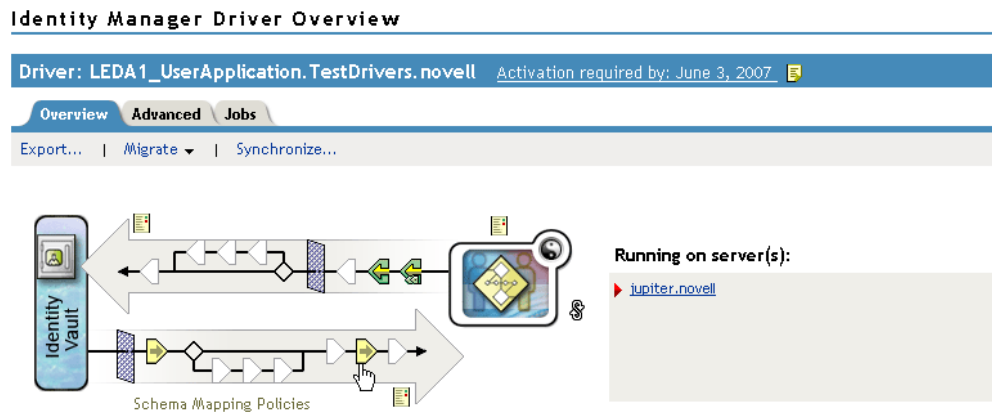


UserApplication

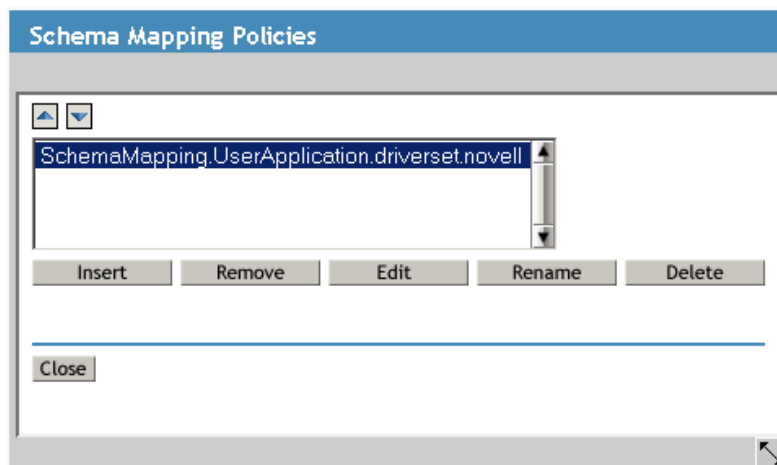
The Identity Manager Driver Overview displays:



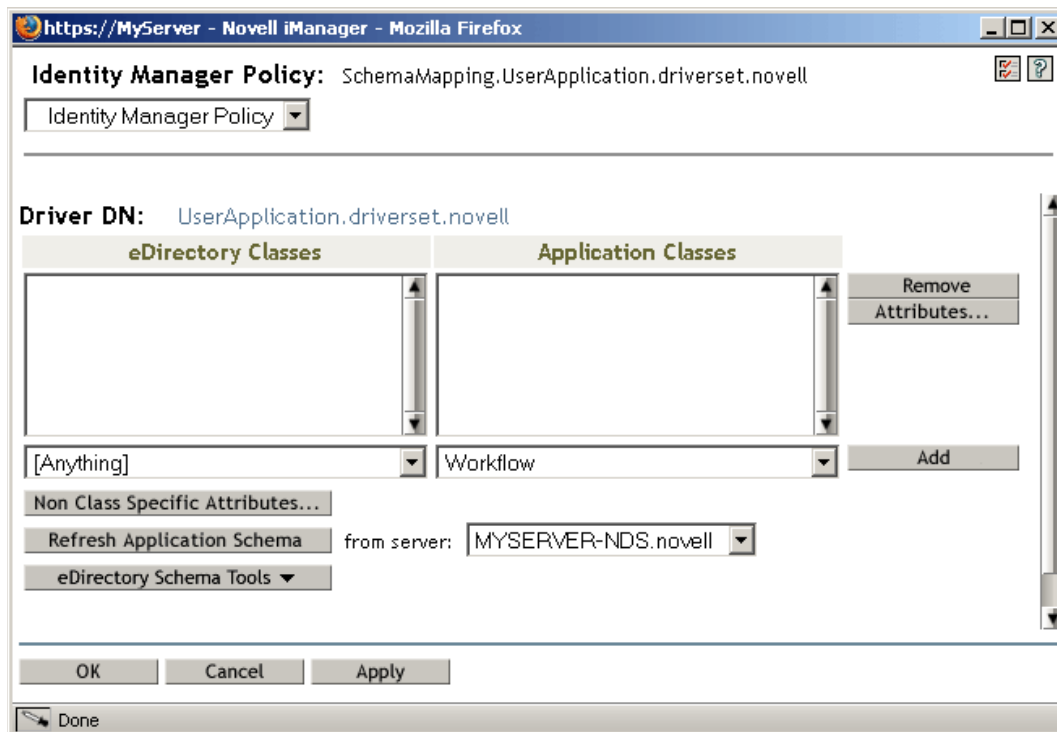
The top horizontal arrow represents the Publisher channel (which is not used in the User Application driver) and the bottom horizontal arrow represents the Subscriber channel. As you pass the mouse pointer over an object in the graphic, a description of the object displays:



- 4 Click the *Schema Mapping Policies* icon. The *Schema Mapping Policies* dialog box displays:

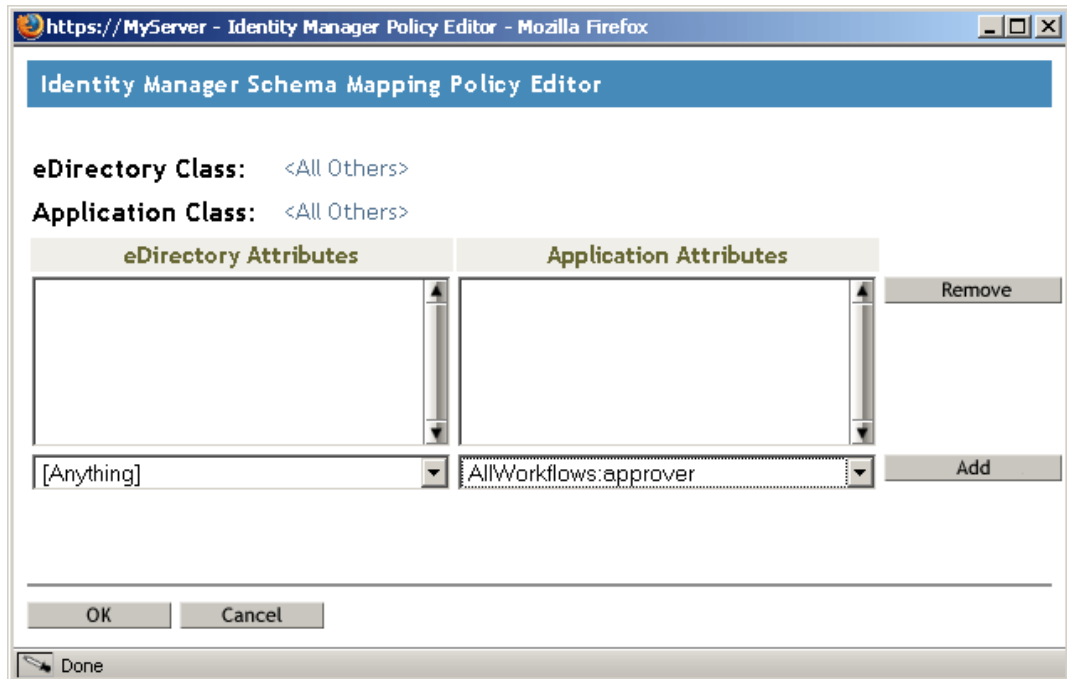


- 5 Click *Edit*. The Identity Manager Policy dialog box displays. (This dialog box maps Identity Vault classes to application classes, but this procedure uses it to map eDirectory attributes to global User Application attributes.)



- 6 Click *Refresh Application Schema*. A message displays informing you that the driver must be stopped in order to read the schema, then restarted. It might 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 displays 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 displays.



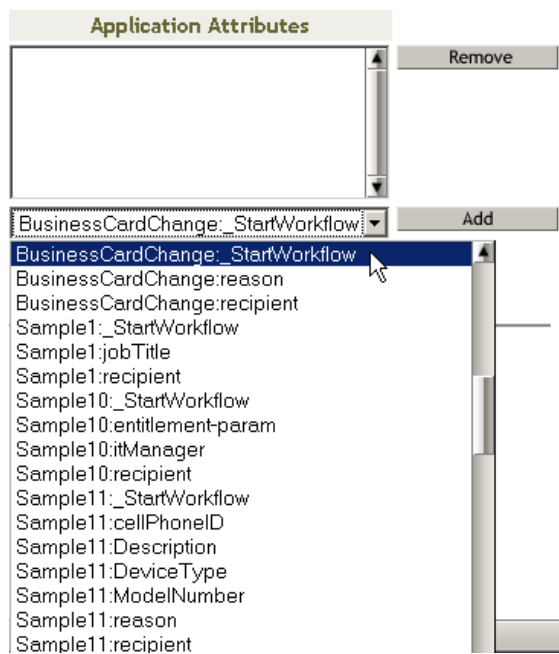
The *eDirectory Attributes* drop-down list contains all eDirectory attributes.

The *Application Attributes* drop-down 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, map *manager* to *Allworkflows:manager*. If you want a different eDirectory attribute (for example, *HRmanager*) to be used for a specific workflow, 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, map the eDirectory attribute that you 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 can also require attributes like *City* and *State*. Alternatively, these attributes can 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 starts whenever an employee's telephone number changes.

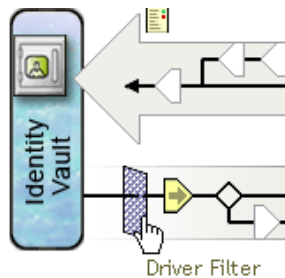
eDirectory Attributes	Application Attributes
<div> <div>Telephone Number</div> <div> <div>Telephone Number</div> <div>teletexTerminalIdentifier</div> <div>telexNumber</div> <div>Timezone</div> <div>Title</div> <div>tollFreePhoneNumber</div> <div>transitionGroupDN</div> <div>Transitive Vector</div> <div>treeReferral</div> <div>Trustees Of New Object</div> <div>Type Creator Map</div> <div>UID</div> <div>uniqueID</div> <div>Unknown</div> <div>Unknown Auxiliary Class</div> <div>Unknown Base Class</div> <div>Used By</div> <div>User</div> <div>userCertificate</div> <div>userPKCS12</div> </div> </div>	<div>BusinessCardChange:_StartWorkflow</div>

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

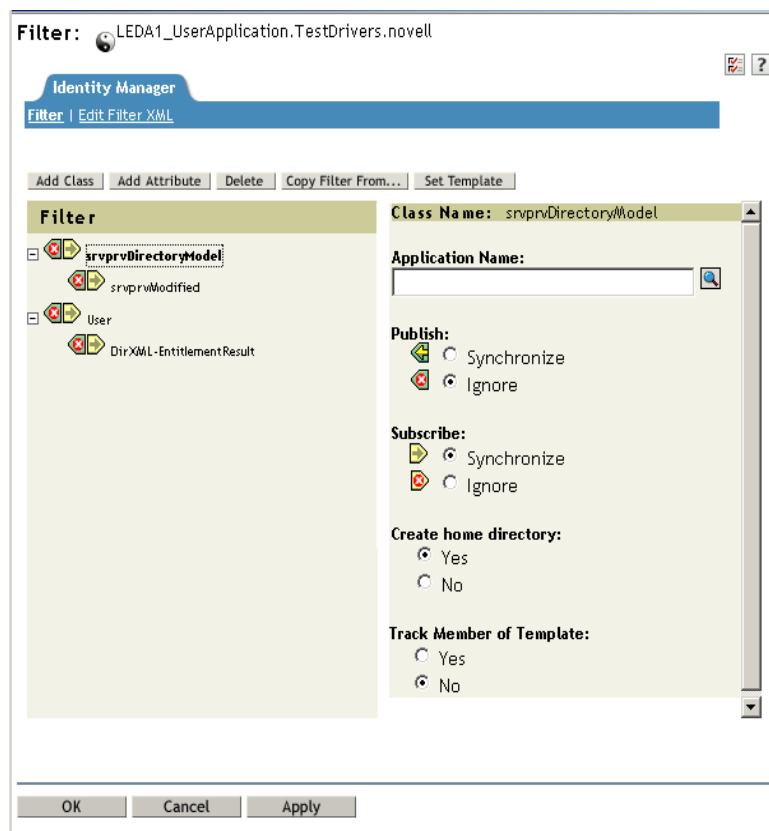
eDirectory Attributes	Application Attributes
<div> <div>Telephone Number</div> </div>	<div>BusinessCardChange:_StartWorkflow</div>
<div>[Anything]</div>	<div>AllWorkflows:approver</div>

- 13** Repeat **Step 10** through **Step 12** to map eDirectory attributes to the workflow `_reason` and `_recipient` attributes.
- 14** If additional eDirectory attributes 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 starts automatically when a change occurs in the eDirectory attribute that is mapped to an application `_StartApprovalFlow` attribute. However, the eDirectory attribute only reaches the Schema Mapping policy if the eDirectory attribute is included in the Driver Filter. In the following steps, add the eDirectory attribute to the Driver Filter.
- 15** Click *OK* to close the Schema Mapping Policy Editor.

- 16 Click *OK* to close the Identity Manager Policy dialog box.
- 17 Click *Close* to close the Schema Mapping Policies dialog box.
- 18 Click the *Driver Filter* icon.



The filter window displays:



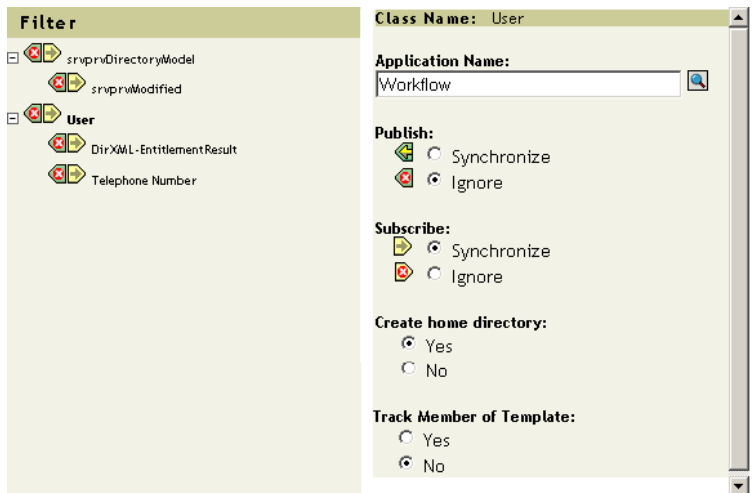
Event filters specify the object classes and the attributes for which the Identity Manager engine processes events. The read-only *Filter* list on the left shows the attributes of the class. The *Class Name* list on the right displays options associated with the target object.

- 19 Click the name of the class to which the attribute that you want to add to the filter belongs (for example, *User*).
- 20 Click *Add Attribute*. A list of attributes displays.

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



- 22** Click the attribute name. The synchronization options for the attribute are displayed on the panel on the right.



- 23** Under *Subscribe*, click *Synchronize*.



- 24** 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.
- 25** Click *OK*. A message displays asking you if you would like the driver to be restarted to put the changes into effect.
- 26** Click *OK*. You are returned to the Identity Manager Driver Overview page.

Configuring Provisioning Request Definitions

17

This section provides instructions for configuring provisioning request definitions. Topics include:

- ♦ [Section 17.1, “About the Provisioning Request Configuration Plug-in,” on page 295](#)
- ♦ [Section 17.2, “Working with the Installed Templates,” on page 296](#)
- ♦ [Section 17.3, “Configuring a Provisioning Request Definition,” on page 298](#)

17.1 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.

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 Configuration role. The Provisioning Requests task consists of the panels described in [Table 17-1](#).

Table 17-1 *Provisioning Requests Task: 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	<p>Provides tools that let you:</p> <ul style="list-style-type: none">♦ 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 <p>When you choose to create a new provisioning request or edit an existing one, the plug-in runs the Provisioning Request Configuration Wizard.</p>

17.2 Working with the Installed Templates

You can define provisioning request definitions from scratch in the Designer for Identity Manager. Alternatively, you can define provisioning requests by modeling them after the provisioning request templates that ship with the product. To use the templates, 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.

Identity Manager ships with the templates listed in [Table 17-2](#).

Table 17-2 *Templates for Provisioning Requests*

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 activity approves the request and the work item 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 activity approves the request and the work item 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 activity approves the request and the work item 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 activity approves the request and the work item forwards to the next activity. This template supports sequential processing.

Template	Description
Five Step Sequential Approval (Timeout Approves)	<p>Requires five approvals for the provisioning request to be fulfilled. If an activity times out, the activity approves the request and the work item forwards to the next activity.</p> <p>This template supports sequential processing.</p>
One Step Approval (Timeout Denies)	<p>Requires a single approval for the provisioning request to be fulfilled. If an activity times out, the workflow denies the request.</p> <p>This template supports sequential processing.</p>
Two Step Sequential Approval (Timeout Denies)	<p>Requires two approvals for the provisioning request to be fulfilled. If an activity times out, the workflow denies the request.</p> <p>This template supports sequential processing.</p>
Three Step Sequential Approval (Timeout Denies)	<p>Requires three approvals for the provisioning request to be fulfilled. If an activity times out, the workflow denies the request.</p> <p>This template supports sequential processing.</p>
Four Step Sequential Approval (Timeout Denies)	<p>Requires four approvals for the provisioning request to be fulfilled. If an activity times out, the workflow denies the request.</p> <p>This template supports sequential processing.</p>
Five Step Sequential Approval (Timeout Denies)	<p>Requires five approvals for the provisioning request to be fulfilled. If an activity times out, the workflow denies the request.</p> <p>This template supports sequential processing.</p>
Two Step Parallel Approval (Timeout Approves)	<p>Requires two approvals for the provisioning request to be fulfilled. If an activity times out, the activity approves the request and the work item forwards to the next activity.</p> <p>This template supports parallel processing.</p>
Three Step Parallel Approval (Timeout Approves)	<p>Requires three approvals for the provisioning request to be fulfilled. If an activity times out, the activity approves the request and the work item forwards to the next activity.</p> <p>This template supports parallel processing.</p>
Four Step Parallel Approval (Timeout Approves)	<p>Requires four approvals for the provisioning request to be fulfilled. If an activity times out, the activity approves the request and the work item forwards to the next activity.</p> <p>This template supports parallel processing.</p>

Template	Description
Five Step Parallel Approval (Timeout Approves)	<p>Requires five approvals for the provisioning request to be fulfilled. If an activity times out, the activity approves the request and the work item forwards to the next activity.</p> <p>This template supports parallel processing.</p>
Two Step Parallel Approval (Timeout Denies)	<p>Requires two approvals for the provisioning request to be fulfilled. If an activity times out, the workflow denies the request.</p> <p>This template supports parallel processing.</p>
Three Step Parallel Approval (Timeout Denies)	<p>Requires three approvals for the provisioning request to be fulfilled. If an activity times out, the workflow denies the request.</p> <p>This template supports parallel processing.</p>
Four Step Parallel Approval (Timeout Denies)	<p>Requires four approvals for the provisioning request to be fulfilled. If an activity times out, the workflow denies the request.</p> <p>This template supports parallel processing.</p>
Five Step Parallel Approval (Timeout Denies)	<p>Requires five approvals for the provisioning request to be fulfilled. If an activity times out, the workflow denies the request.</p> <p>This template supports parallel processing.</p>

Workflows and provisioned resources. When you create a new provisioning request definition, you bind it to 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* (lowercase).

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 lowercase. This is necessary to ensure that categories work properly in the Identity Manager User Application.

For details on creating provisioning categories, see the *Identity Manager User Application: Design Guide*.

17.3 Configuring a Provisioning Request Definition

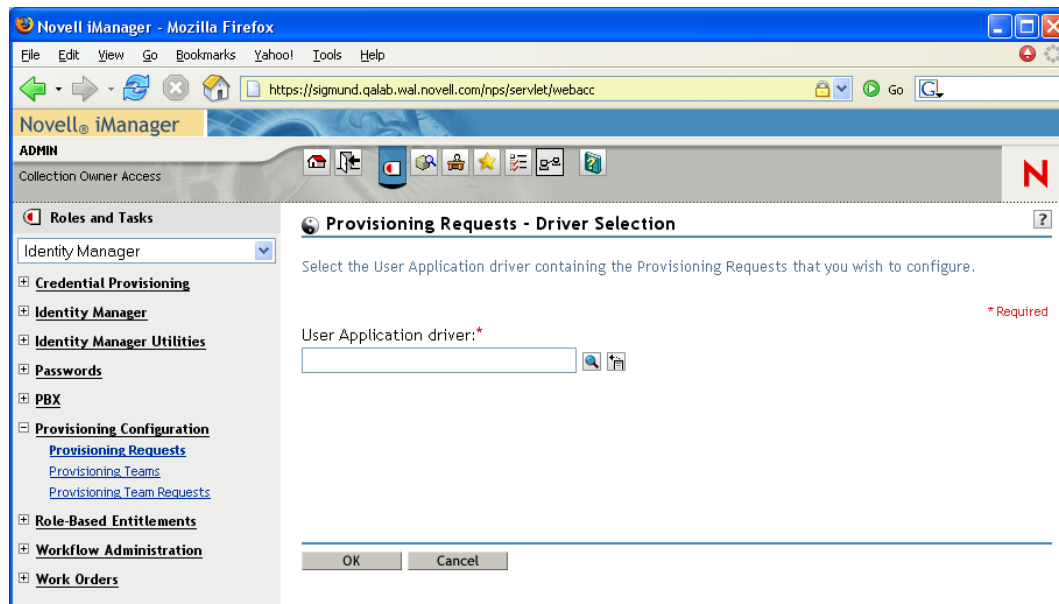
Before configuring a provisioning request definition, you need to select the Identity Manager User Application driver that contains the definition. Having selected the driver, you can create a new provisioning request definition or edit an existing definition. You can also delete provisioning request definitions, change the status of a request definition, or define rights for a request definition.

17.3.1 Selecting the Driver

To select an Identity Manager User Application driver:

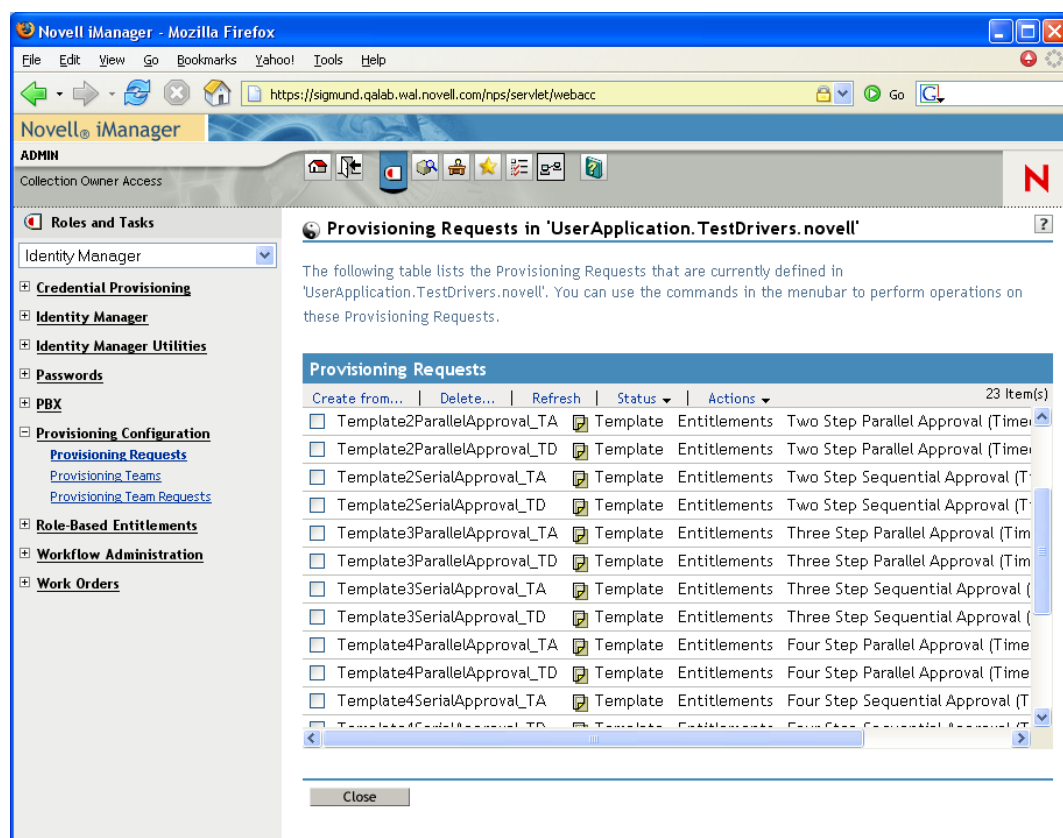
- 1 Select the *Identity Manager* category in iManager.
- 2 Open the *Provisioning Request Configuration* role.
- 3 Click the *Provisioning Requests* task.

iManager displays the User Application Driver panel.



- 4 Specify the driver name in the *User Application Driver* field, then 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.

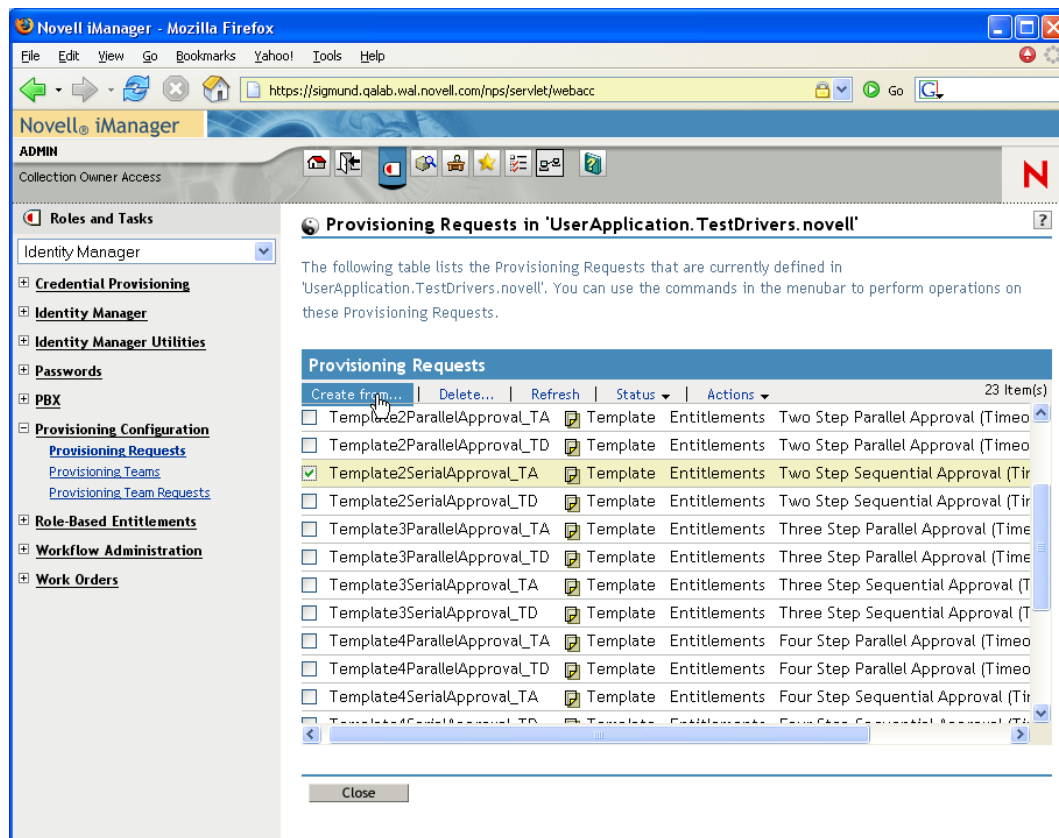
Changing the driver. When you have selected a driver, the driver selection remains in effect for the duration of your iManager session, unless you select a new driver. To select a new driver, click the *Actions* command, then choose *Select User Application Driver* from the *Actions* menu.

17.3.2 Creating or Editing a Provisioning Request

To create a new provisioning request:

- 1 Click 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.

https://sigmund.qalab.wal.novell.com - Provisioning Request Configuration Wizard - FrameSet - Mozilla Firefox

Create New Provisioning Request

Step 1 of 6: Edit general Provisioning Request information.

Enter the name for the new Provisioning Request. Enter the display names and descriptions for the defined languages. English will be displayed for undefined languages.

Name (CN):

Provisioning Request Localized Strings

[Add...](#) | [Delete...](#)

Language	Display name	Description
<input type="checkbox"/> English	<input type="text"/>	<input type="text"/>

<< Back Next >> Cancel Finish

- 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 is used to identify the provisioning request throughout the User Application.
- 5 To add a new language to the list, click *Add*, then select the desired language.
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 305.
- 8 Configure the activities for the workflow associated with the request definition, as described in “[Configuring the Workflow Activities](#)” on page 308.
- 9 Specify the access rights for the request definition, as described in “[Specifying the Access Rights for the Provisioning Request](#)” on page 318.
- 10 Specify the initial status for the request definition, as described in “[Specifying the Initial Status of the Provisioning Request](#)” on page 319.

11 Review your settings, then click *Finish*.

Create New Provisioning Request

Step 6 of 6: Summary

The following information was collected and will be used to create a new Provisioning Request:

Provisioning Request to Create:

- Name: **test**
- Create from: **Template2SerialApproval_TA**
- Context: **RequestDefs.AppConfig.UserApplication.TestDrivers.novell**
- Display name: **test**
- Description: **test**
- Status: **Active**
- Category: **Entitlements**
- Grant: **true**
- Revoke: **false**
- Notify addressee: **true**
- Trustee assignments: **users.idmsample.novell**
- User activities:
 - First approval (Id: approval_1)**
 - Addressee: **IDVault.get(recipient,'user','manager')**
 - Timeout: **<System default>**
 - Retry attempts: **3**
 - Retry addressee: **IDVault.get(approval_1.getAddressee(),'user','manager')**
 - Second approval (Id: approval_2)**
 - Addressee: **IDVault.get(approval_1.getAddressee(),'user','manager')**
 - Timeout: **<System default>**
 - Retry attempts: **3**
 - Retry addressee: **IDVault.get(approval_2.getAddressee(),'user','manager')**

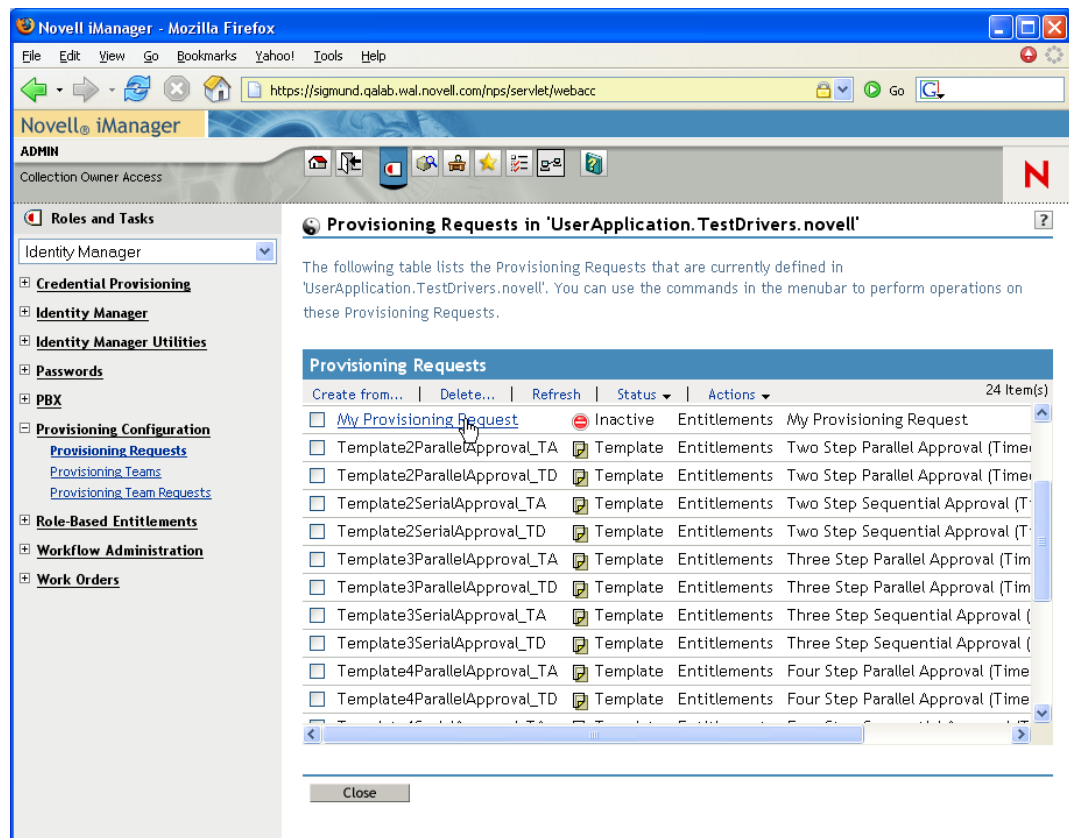
Resource to be provisioned:

- Name: **testresource**
- Context: **ResourceDefs.AppConfig.UserApplication.TestDrivers.novell**
- Entitlement based: **true**

<< Back Next >> Cancel Finish

To edit an existing provisioning request:

- 1 Click 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.

If you have a large number of request definitions, you might want to sort the list by a particular column, such as the *Name* or *Description*. To sort by a particular column, click 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*, then type the localized text in the *Display Name* and *Description* fields. This text is used to identify the provisioning request throughout the User Application.
- 3 To add a new language to the list, click *Add*, then select the desired language.
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 305.
- 6 Configure the activities for the workflow associated with the request definition, as described in “[Configuring the Workflow Activities](#)” on page 308.
- 7 Specify the access rights for the request definition, as described in “[Specifying the Access Rights for the Provisioning Request](#)” on page 318.
- 8 Specify the initial status for the request definition, as described in “[Specifying the Initial Status of the Provisioning Request](#)” on page 319.

9 Review your settings, then 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 *Novell Identity Manager: Administration Guide*.

To specify the provisioned resource:

- 1 To use the target that is currently associated with the request definition, select *Provisioned resource*.

Provisioned resource 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 option is not selected.

- 2 To bind the request definition to another resource that was previously defined within the currently selected driver, select *Available provisioned resources*, then pick a target from the drop-down list.

The screenshot shows a web-based configuration wizard titled "Edit Existing Provisioning Request" in a Mozilla Firefox browser window. The address bar shows "http://your.server.address:8080". The wizard is at "Step 2 of 6: Define the Provisioned Resource and Category for the Provisioning Request." A text box explains: "The Provisioned Resource shown is the one currently referenced by this Provisioning Request. The Provisioned Resource can also be specified by selecting from other available Provisioned Resources in eDirectory or by creating a new Provisioned Resource based on an Identity Manager entitlement." There are two radio buttons: "Provisioned Resource:" (unselected) and "Available Provisioned Resources:" (selected). Below the second radio button is a dropdown menu with two options: "<Select an existing Provisioned Resource>" and "testresource". A mouse cursor is pointing at "testresource". Below the dropdown are three small icons: a plus sign, a pencil, and a minus sign. A text box below the dropdown states: "The category selected will be used for this Provisioning Request. If you change the Provisioned Resource, its category will be selected." Below this is a "Category:" label and a dropdown menu showing "Entitlements". At the bottom are four buttons: "<< Back", "Next >>", "Cancel", and "Finish". A "Done" button is visible in the bottom right corner of the wizard frame.

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* drop-down 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 also changed 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* drop-down list.

- 4 To create a new resource based on an Identity Manager entitlement, click the + icon.



To edit an existing resource, click the pen icon.



To define the characteristics of the resource:

- 4a Specify the name for the resource in the *Name (CN)* field.
 - 4b Select a category for the resource in the *Category* drop-down list.
 - 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*, then type the localized text in the *Display Name* and *Description* fields. This text is used to identify the provisioning resource throughout the User Application.
 - 4e To add a new language to the list, click *Add*, then select the desired language.
- By default, a newly created provisioning resource supports only English.

https://sigmund.qalab.wal.novell.com - Provisioned Resource Wizard - FrameSet - Mozilla Firefox

Create New Provisioned Resource

Step 1 of 3: Edit general Provisioned Resource information.

Enter the name for the new Provisioned Resource, select its category and select its associated Identity Management entitlement. Enter the display names and descriptions for the defined languages. English will be displayed for undefined languages.

Name (CN):

Category:

Entitlement:

Provisioned Resource Localized Strings

Add... | Delete...

Language	Display name	Description
<input type="checkbox"/> English	<input type="text" value="MyResource"/>	<input type="text" value="This is my resource."/>

<< Back Next >> Cancel Finish

- 5 Click *Next*.

The Provisioned Resource wizard displays a page to allow you to provide data for any parameters required for the entitlement.

https://sigmund.qalab.wal.novell.com - Provisioned Resource Wizard - FrameSet - Mozilla Firefox

Create New Provisioned Resource

Step 2 of 3: Provide the necessary data to configure the Provisioned Resource.

Identity Manager Entitlement:

Name: **IDVAULT**

Display name: **ID Vault value set**

Description: **manage id vault attributes**

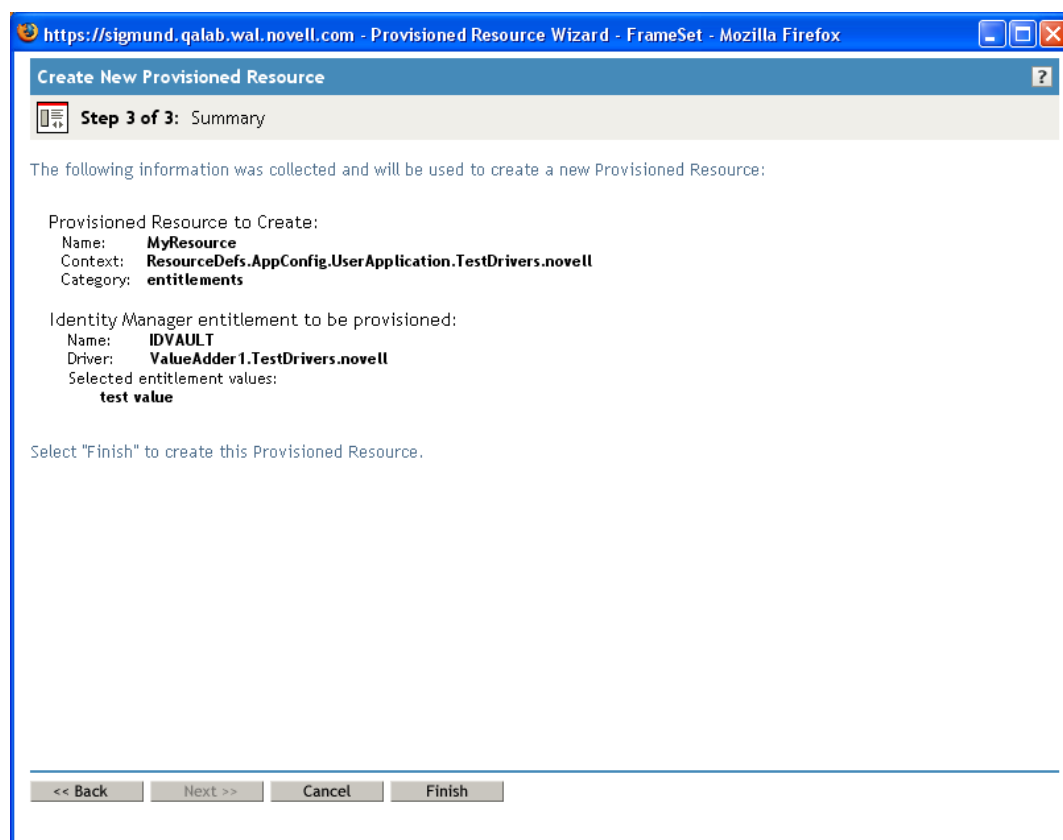
This entitlement requires that a single value be specified. The value must be entered below.

Value:

<< Back Next >> Cancel Finish

- 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.



https://sigmund.qalab.wal.novell.com - Provisioned Resource Wizard - FrameSet - Mozilla Firefox

Create New Provisioned Resource

Step 3 of 3: Summary

The following information was collected and will be used to create a new Provisioned Resource:

Provisioned Resource to Create:

- Name: **MyResource**
- Context: **ResourceDefs.AppConfig.UserApplication.TestDrivers.novell**
- Category: **entitlements**

Identity Manager entitlement to be provisioned:

- Name: **IDVAULT**
- Driver: **ValueAdder 1.TestDrivers.novell**
- Selected entitlement values:
test value

Select "Finish" to create this Provisioned Resource.

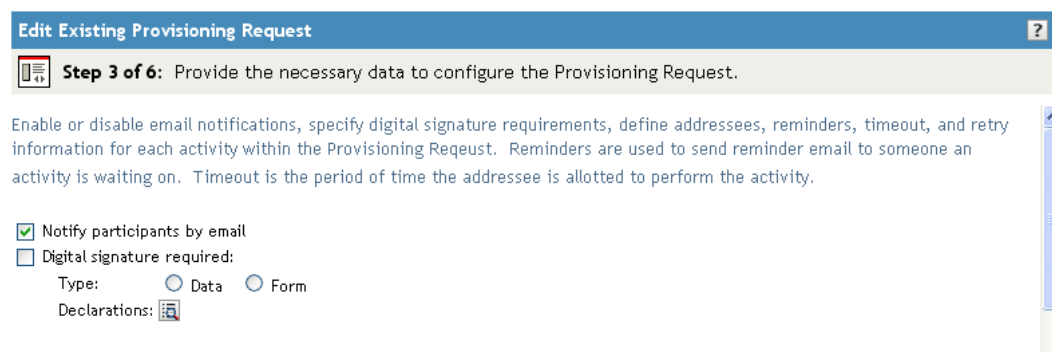
<< Back Next >> Cancel Finish

7 Click *Finish*.

Configuring the Workflow Activities

To configure the activities for the associated workflow:

- 1 Specify whether you want to enable e-mail notifications for the workflow by selecting or deselecting the *Notify participants by e-mail* check box.



Edit Existing Provisioning Request


Step 3 of 6: Provide the necessary data to configure the Provisioning Request.

Enable or disable email notifications, specify digital signature requirements, define addressees, reminders, timeout, and retry information for each activity within the Provisioning Request. Reminders are used to send reminder email to someone an activity is waiting on. Timeout is the period of time the addressee is allotted to perform the activity.

☒ Notify participants by email

☐ Digital signature required:

Type: ☐ Data ☐ Form

Declarations: 

- 2 Specify whether a digital signature is required to initiate the provisioning request by selecting or deselecting the *Digital signature required* check box.

Edit Existing Provisioning Request ?

Step 3 of 6: Provide the necessary data to configure the Provisioning Request.

Enable or disable email notifications, specify digital signature requirements, define addressees, reminders, timeout, and retry information for each activity within the Provisioning Request. Reminders are used to send reminder email to someone an activity is waiting on. Timeout is the period of time the addressee is allotted to perform the activity.

☐ Notify participants by email

☒ Digital signature required:

Type: ☒ Data ☐ Form

Declarations:

- 2a** If you enable the *Digital signature required* check box, specify whether the digital signature will use data or form as its type:
- ♦ *Data* specifies that the XML signature serve as the user agreement. When Data is selected, the XML data is written to the audit log.
 - ♦ *Form* specifies that a PDF document that includes the digital signature declaration be generated. This document serves as the user agreement. The user can preview the generated PDF document before submitting a request or approval. When Form is selected, the PDF document (encapsulated in XML) is written to the audit log.
- 2b** If you enable the *Digital signature required* check box, you also need to specify a digital signature confirmation string. To do this, click the *Declarations* icon.

Edit Existing Provisioning Request ?

Step 3 of 6: Provide the necessary data to configure the Provisioning Request.

Enable or disable email notifications, specify digital signature requirements, define addressees, reminders, timeout, and retry information for each activity within the Provisioning Request. Reminders are used to send reminder email to someone an activity is waiting on. Timeout is the period of time the addressee is allotted to perform the activity.

☐ Notify participants by email

☒ Digital signature required:

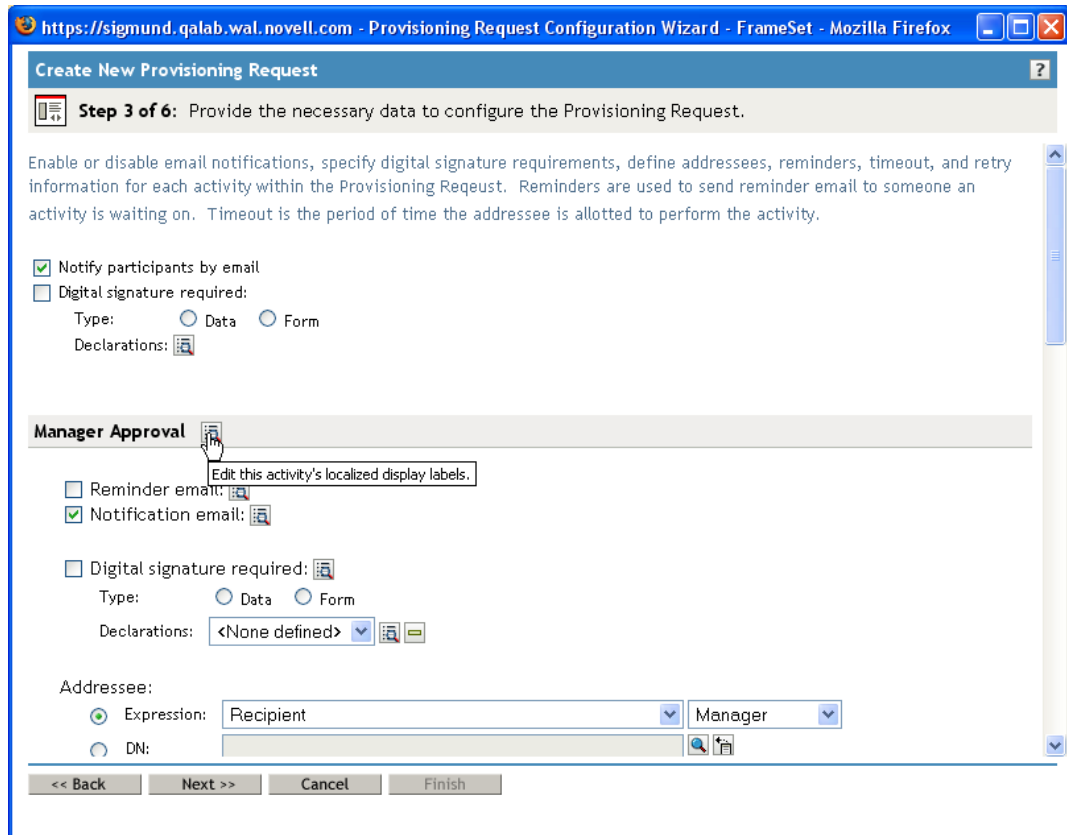
Type: ☒ Data ☐ Form

Declarations:

Edit the request's localized declaration strings.

Type the signature confirmation string, then click *OK*.

- 3 (Optional) For each workflow activity, change the display label by clicking the icon beside the name of the activity (in this case, *Manager Approval*).



https://sigmund.qalab.wal.novell.com - Provisioning Request Configuration Wizard - FrameSet - Mozilla Firefox

Create New Provisioning Request


Step 3 of 6: Provide the necessary data to configure the Provisioning Request.


Enable or disable email notifications, specify digital signature requirements, define addressees, reminders, timeout, and retry information for each activity within the Provisioning Request. Reminders are used to send reminder email to someone an activity is waiting on. Timeout is the period of time the addressee is allotted to perform the activity.


☒ Notify participants by email


☐ Digital signature required:


Type: ☐ Data ☐ Form

Declarations: 



Manager Approval  Edit this activity's localized display labels.

☐ Reminder email: 

☒ Notification email: 

☐ Digital signature required: 

Type: ☐ Data ☐ Form

Declarations:  

Addressee:

☒ Expression:

☐ DN:

<< Back Next >> Cancel Finish

Type the display label in the *Display Label* field, then click *OK*.

https://your.server.address - Provisioning Request Display Label Editor - FrameSet - Mozilla Firefox

User Activity - Display Label Editor

Enter the User Activity's display labels for the defined languages. English will be displayed for undefined languages.

User Activity Localized Display Labels

Add | Remove

Language	Display Label
<input type="checkbox"/> English	Manager approval

OK Cancel

Done oldschool.qalab.wal.novell.com

The default display labels (First approval, Second approval, and so on) suggest that approvals are processed sequentially. For parallel flows, you might 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.

- 4 (Optional) For each workflow activity that supports quorums or multiple addressees, add additional addressees by clicking the *Add another addressee to this user activity* icon beside the name of the activity.


Manager Approval [List Icon] [Plus Icon]



This user activity will require [Add another addressee to this user activity]s approve it. The portion required is specified by the approver condition. The approver condition can be the specific number or the percentage of the addressees required for approval.

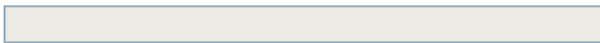


Approver condition: 25%

When you click this button, a new Addressee section is presented on the page. You can use the controls in this section to define an expression or DN for the addressee (as described in the next

step in this procedure). To delete the addressee, you can click the minus sign next to the addressee:

Addressee: 

☒ Expression: Recipient  Manager 

☐ DN:   

(e.g., CN=Admin,O=Novell)

5 For each workflow activity, also provide the following information:

Field	Description
Reminder email	<p>Indicates whether reminder e-mail messages should be sent for this activity.</p> <p>Click the <i>Edit this activity's reminder email</i> icon to configure reminder notifications. Specify these settings:</p> <ul style="list-style-type: none"> ♦ <i>Start</i> specifies when to send the first reminder. The start value is an offset from the time of the first assignment associated with the activity. ♦ <i>Interval</i> specifies how often to send reminders after the first reminder has been sent. ♦ <i>Email Template</i> specifies the language-independent name for the template to use for reminder e-mail messages. After the template name has been specified, the notification engine can determine which language-specific template to use at runtime. <p>The language-independent template can have any name you like. The default template for reminder e-mail messages is called:</p> <p>Provisioning Reminder</p> <p>Each language-specific version of a template must have a suffix that provides a language code (for example, <i>_fr</i> for French, <i>_es</i> for Spanish, and so forth).</p>

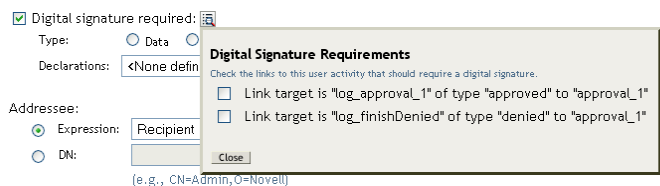
Field	Description
Notification email	<p>Indicates whether notification e-mail messages should be sent for this activity.</p> <p>Click the <i>Edit this activity's notification email</i> icon to configure notification e-mail messages. Specify these settings:</p> <ul style="list-style-type: none"> ♦ <i>Email Template</i> specifies the language-independent name for the template to use for notification e-mail messages. After the template name has been specified, the notification engine can determine which language-specific template to use at runtime. <p>The language-independent template can have any name you like. The default template for notification e-mail messages is called:</p> <p>Provisioning Notification</p> <p>Each language-specific version of a template must have a suffix that provides a language code (for example, <code>_fr</code> for French, <code>_es</code> for Spanish, and so forth).</p> <ul style="list-style-type: none"> ♦ <i>Replacement Parameters Map</i> specifies one or more substitution values for the replacement parameters used in the e-mail template. To edit an existing value, click the replacement parameter, then specify an ECMAScript expression or fixed value. To add a new substitution value, click Add, select the target parameter, then specify an expression or fixed value.

Field	Description
-------	-------------

Digital signature required

Indicates whether a digital signature is required to approve the request. Because each approval step might have more than one outgoing link, you need to specify whether a digital signature is required for each link.

When you enable this check box, you are prompted to select the link for which a digital signature is required. Select the link, then click *Close*.



If you enable the *Digital signature required* check box, specify whether the digital signature will use data or form as its type:

- ♦ *Data* specifies that the XML signature serve as the user agreement. When Data is selected, the XML data is written to the audit log.
- ♦ *Form* specifies that a PDF document that includes the digital signature declaration be generated. This document serves as the user agreement. The user can preview the generated PDF document before submitting a request or approval. When Form is selected, the PDF document (encapsulated in XML) is written to the audit log.

If you enable the *Digital signature required* check box, you also need to specify a digital signature confirmation string. First, create an identifier for the string by selecting *Create one* in the Declarations list box. Then select the ID and click the *Declarations* icon.



Type the signature confirmation string, then click *OK*.

Field	Description
Approver Condition	<p>Specifies the approver condition for quorums.</p> <p>When the approver type for an activity is configured to support quorums, you can set the approver condition to define the number of approvers required to approve the activity. You can specify the condition as a numeric constant or a percentage of addressees.</p> <p>For example, if you wanted to require a 25-percent majority, you would specify an approver condition of 25%, as shown below.</p> <p>This user activity will require that a portion of the addressees approve it. The portion required is specified by the approver condition. The approver condition can be the specific number or the percentage of the addressees required for approval.</p> <p>Approver condition: <input type="text" value="25%"/></p> <p>Alternatively, if you wanted to require approvals from two of the addressees, you would set the approver condition to 2.</p> <hr/> <p>NOTE: When quorum support is enabled for an activity, you cannot specify retry settings for the activity. Therefore, the Retry Escalation Reminder Email, Retry Attempts, Retry Interval, and Retry Addressee fields are not displayed.</p> <hr/>

Field	Description
Addressee Expression	<p>Specifies a dynamic expression that identifies the addressee for the activity. The addressee is determined at runtime, based on how the expression is evaluated.</p> <p>The first term of an addressee expression can be any of the following values:</p> <ul style="list-style-type: none"> ♦ Initiator ♦ Recipient ♦ Addressee of <i>activity-name</i> <p>A separate Addressee of <i>activity-name</i> term is listed in the Expression drop-down list for each activity in the workflow (except the activity you are currently configuring). The <i>activity-name</i> is the display label you specified for the activity, or the default name, if you did not specify a display label.</p> <p>The second term of an addressee expression can be either of the following values:</p> <ul style="list-style-type: none"> ♦ Manager ♦ <No attribute> <p>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) are available for selection if they meet the following requirements:</p> <ul style="list-style-type: none"> ♦ Must be defined on the User entity in the abstraction layer ♦ Must be single-valued ♦ Must have a DN data type
Addressee DN	<p>Specifies the distinguished name for a user, group, or task group.</p> <hr/> <p>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.</p> <hr/>
Timeout	<p>Specifies the period of time allotted for the activity to complete its processing. The timeout interval is the total time allowed for the activity, not the time allowed for each retry.</p> <p>The Timeout setting for the activity takes precedence over the Retry Attempts and Retry Interval values. Therefore, if the Timeout setting for the activity is reached before one or more of the retries have been attempted, the activity finishes processing without executing these retries. For example, if you set the timeout to 10 minutes, and define three retries with a retry interval of 5 minutes, the activity will finish after 10 minutes without attempting all of the retries. In this example, the second retry will be canceled. At the conclusion of the activity, the workflow engine will follow the link defined by the final timeout action in Designer.</p> <p>Specify a value in milliseconds, seconds, minutes, hours, or days.</p>

Field	Description
Retry Escalation Reminder Email	<p>Specifies whether e-mail messages should be sent to remind the current addressee of the activity that an action is required. Check this box to enable this feature.</p> <p>To change the retry escalation reminder notification settings for this activity, click the <i>Edit this activity's retry reminder email</i> icon to configure escalation reminder notifications. Specify these settings:</p> <ul style="list-style-type: none"> ♦ <i>Start</i> specifies when to send the first reminder. The start value is an offset from the time of the retry assignment. ♦ <i>Interval</i> specifies how often to send reminders after the first reminder has been sent. ♦ <i>Email Template</i> specifies the language-independent name for the template to use for reminder e-mail messages. After the template name has been specified, the notification engine can determine which language-specific template to use at runtime. <p>The language-independent template can have any name you like. The default template for reminder e-mail messages is called:</p> <p>Provisioning Reminder</p> <p>Each language-specific version of a template must have a suffix that provides a language code (for example, <code>_fr</code> for French, <code>_es</code> for Spanish, and so forth).</p>
Retry Attempts	<p>Specifies the number of times to retry the activity in the event that the retry interval has been reached.</p> <p>When an activity reaches the retry interval, the workflow process tries to complete the activity again, depending on the retry count specified for the activity. With each retry, the workflow process can escalate the activity 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 is executed, the activity might be marked as approved or denied, depending on how the workflow was configured.</p> <p>The Timeout setting for the activity takes precedence over the Retry Attempts and Retry Interval values. Therefore, if the Timeout setting for the activity is reached before one or more of the retries have been attempted, the activity finishes processing without executing these retries. For example, if you set the timeout to 10 minutes, and define three retries with a retry interval of 5 minutes, the activity will finish after 10 minutes without attempting all of the retries. In this example, the second retry will be canceled. At the conclusion of the activity, the workflow engine will follow the link defined by the final timeout action in Designer.</p>
Retry Interval	<p>Defines the period of time allotted for the addressee to complete the task. When the Retry Interval is reached, the workflow can optionally reassign the activity to a new addressee or try again with the original addressee. The Addressee Expression gives you control over the reassignment.</p>

Field	Description
Retry Addressee Expression	<p>Specifies a dynamic expression that identifies the user who should get this task in the event that the timeout limit has been reached.</p> <p>The retry addressee is determined at runtime, based on how the expression is evaluated.</p> <p>The first term of an addressee expression can be any of the following values:</p> <ul style="list-style-type: none"> ♦ Initiator ♦ Recipient ♦ Addressee of <i>activity-name</i> <p>A separate Addressee of <i>activity-name</i> term is listed in the Expression drop-down for each activity in the workflow (including the activity you are currently configuring). The <i>activity-name</i> is the display label you specified for the activity, or the default name, if you did not specify a display label.</p> <p>The second term of an addressee expression depends on how the data abstraction layer has been defined. For example, you might see the following values:</p> <ul style="list-style-type: none"> ♦ Manager ♦ Group ♦ Direct Reports ♦ <No attribute> <p>If you select <i>Manager</i>, each retry will escalate to a new manager at a higher level within the organization. Therefore, 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.</p>
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.

6 When you finish configuring an activity, you might need to scroll down to see the other activities for the flow.

7 Click *Next*.

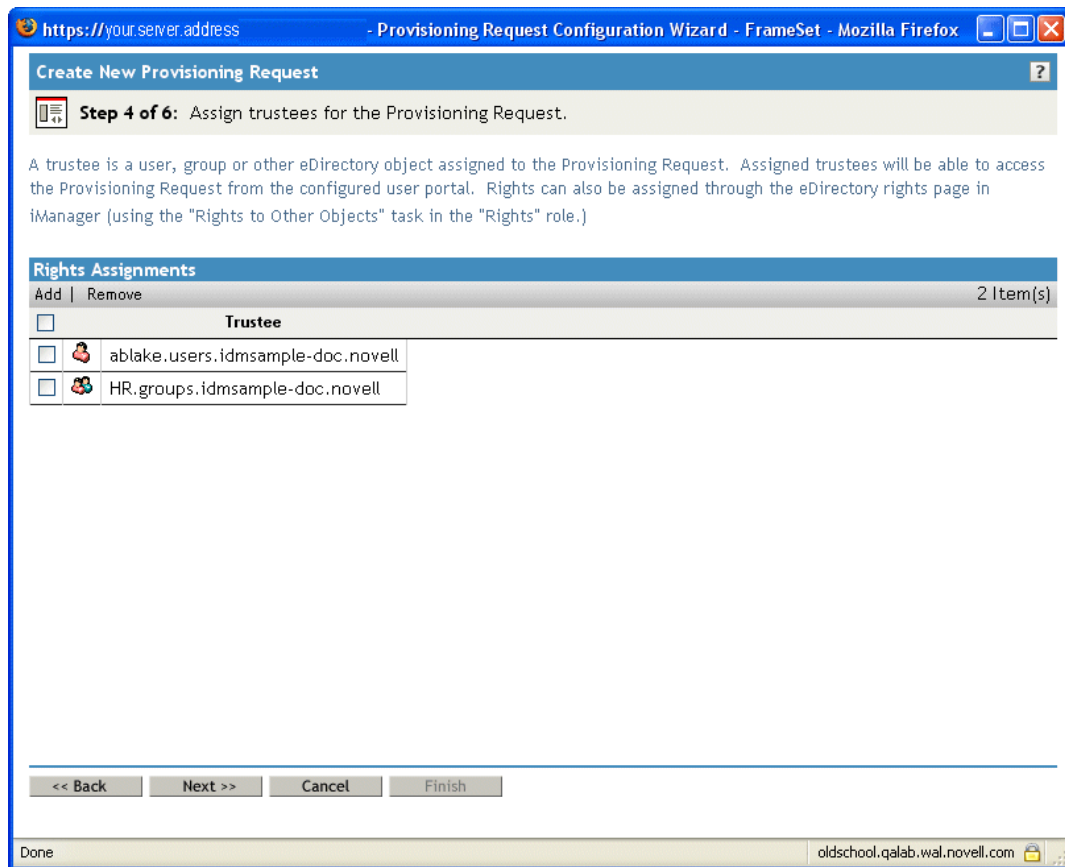
NOTE: The number of activities you can configure varies, depending on which provisioning request definition was used as the basis for creating this definition. The number and type of entitlement parameters varies, depending on the provisioned resource associated with the request.

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*, then select the object.

After you have 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, then 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 button for the desired status:

Status	Description
Active	Available for use.
Inactive	Temporarily unavailable for use. This is the default.
Retired	Permanently disabled.

https://your.server.address - Provisioning Request Configuration Wizard - FrameSet - Mozilla Firefox

Create New Provisioning Request

Step 5 of 6: Determine the status of the Provisioning Request.

The status will determine the availability of the Provisioning Request to users in the system. An active Provisioning Request will be available to users within the system. The status of a Provisioning Request can be changed at any time.

☒ Active
☐ Inactive
☐ Retired

Disposition of Identity Manager Entitlement

Select the appropriate radio button based on whether this Provisioning Request is to "Grant" or "Revoke" the Provisioned Resource's Identity Manager entitlement.

☒ Grant the Identity Manager entitlement
☐ Revoke the Identity Manager entitlement

<< Back Next >> Cancel Finish

Done oldschool.qalab.wal.novell.com

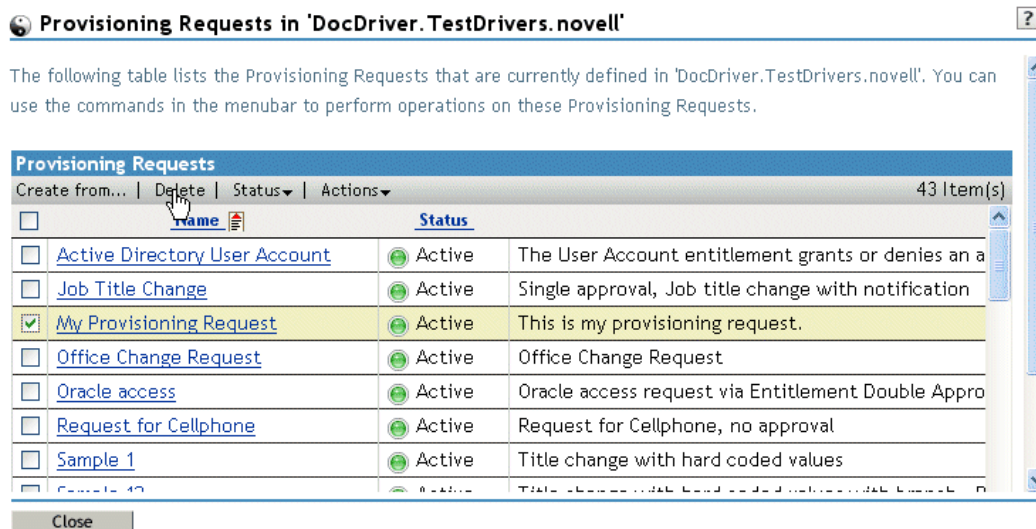
- 2 Click the button for the correct action (*Grant* or *Revoke*).
- 3 Click *Next*.

17.3.3 Deleting a Provisioning Request

To delete a provisioning request:

- 1 Select the provisioning request you want to delete by clicking the check box next to the name.
You are not permitted to delete a provisioning request that is a template.

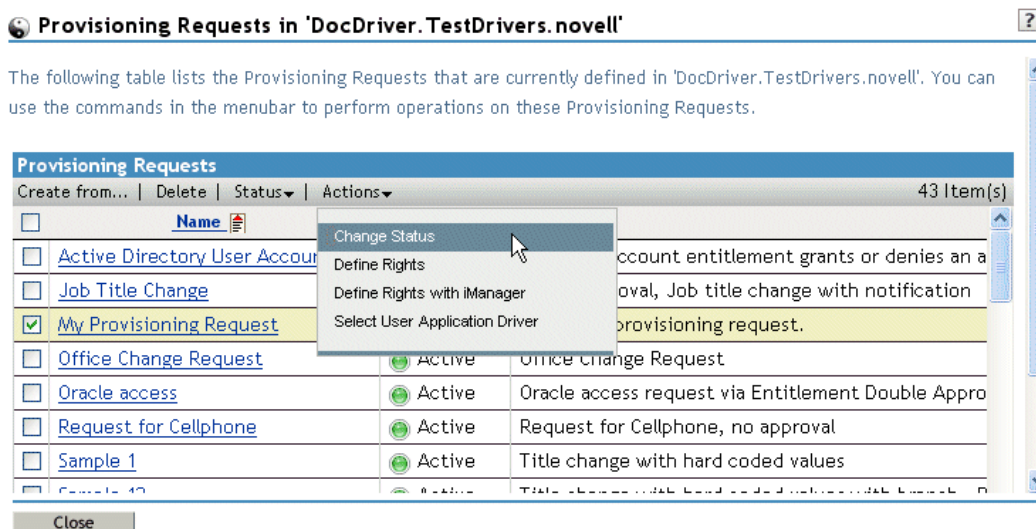
- 2 Click the *Delete* command in the Provisioning Request Configuration panel.



17.3.4 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:

- 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.

This section provides instructions for managing provisioning workflows at runtime. It also provides instructions for configuring e-mail notification for provisioning workflows.

Topics include:

- ♦ [Section 18.1, “About the Workflow Administration Plug-in,” on page 325](#)
- ♦ [Section 18.2, “Managing Workflows,” on page 325](#)
- ♦ [Section 18.3, “Configuring the E-Mail Server,” on page 333](#)
- ♦ [Section 18.4, “Working with E-Mail Templates,” on page 334](#)

18.1 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 stopped and cannot be restarted.

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.

The Workflows task comprises the panels listed in [Table 18-1](#).

Table 18-1 *Workflows Task: Panels*

Panel	Description
Workflows	<p>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.</p> <p>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 select a driver before you can log in to the server and begin workflow administration.</p> <p>When you have selected a driver, you can specify search criteria for selecting the workflows to manage.</p>
Workflow Detail	<p>Provides a read-only user interface for viewing the details about a specific workflow.</p>

18.2 Managing Workflows

This section includes procedures for managing provisioning workflows using the Workflow Administration plug-in:

- ♦ [Section 18.2.1, “Connecting to a Workflow Server,” on page 326](#)

- ♦ Section 18.2.2, “Finding Workflows that Match Search Criteria,” on page 328
- ♦ Section 18.2.3, “Controlling the Active Workflows Display,” on page 329
- ♦ Section 18.2.4, “Terminating a Workflow Instance,” on page 330
- ♦ Section 18.2.5, “Viewing Details about a Workflow Instance,” on page 330
- ♦ Section 18.2.6, “Reassigning a Workflow Instance,” on page 330
- ♦ Section 18.2.7, “Managing Workflow Processes in a Cluster,” on page 331

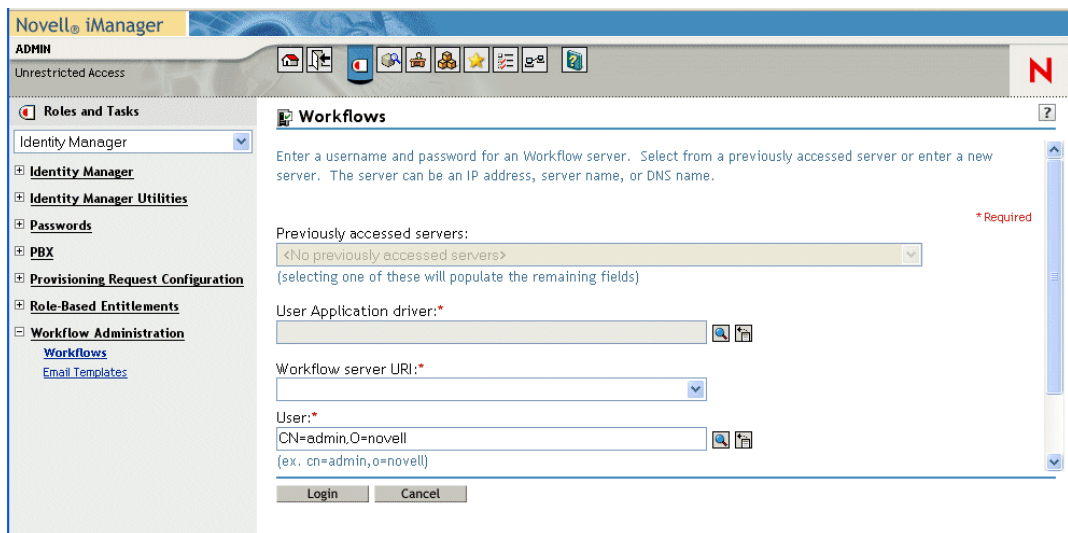
18.2.1 Connecting to a Workflow Server

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 the *Workflows* task.

iManager displays the Workflows panel.



- 4 If you accessed the target workflow server previously, you can select the server from the *Previously accessed servers* drop-down list.
iManager fills in the remaining fields on the panel.
- 5 If you have not yet accessed a workflow server, specify the driver name in the *User Application Driver* field, then click *OK*.

iManager fills in the *Workflow server URI* and *User* fields.

The screenshot shows the 'Workflows' configuration window in Novell iManager. The left sidebar contains a tree view with 'Workflows' selected under 'Workflow Administration'. The main area has a title bar 'Workflow - Server Query Frame'. Below the title bar, there's a section 'Workflows' with a help icon. The text says: 'Enter a username and password for an Workflow server. Select from a previously accessed server or enter a new server. The server can be an IP address, server name, or DNS name.' Below this, there's a dropdown for 'Previously accessed servers:' with the value '<No previously accessed servers>' and a note '(selecting one of these will populate the remaining fields)'. There's a red asterisk and the word 'Required' to the right. Below the dropdown is a text field for 'User Application driver:' with the value 'UserApplication.JKlobucher.TestDrivers.novell'. Below that is a text field for 'Workflow server URI:' with a red asterisk and 'Required' to the right. Below that is a text field for 'User:' with the value 'CN=admin,O=novell' and a note '(ex. cn=admin,o=novell)'. Below that is a text field for 'Password:' with a red asterisk and 'Required' to the right. At the bottom are 'Login' and 'Cancel' buttons.

6 Type the password for the user in the *Password* field.

7 Click *Login*.

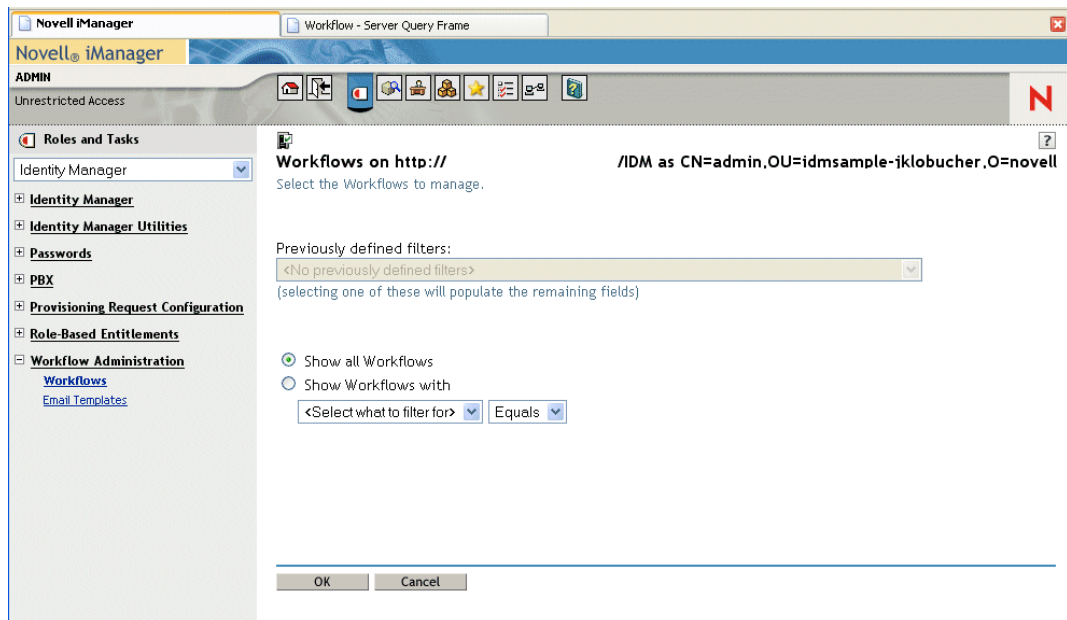
The Workflow Administration plug-in displays a page that allows you to specify a filter for finding workflows:

The screenshot shows the 'Workflows on http://' configuration window in Novell iManager. The left sidebar contains a tree view with 'Workflows' selected under 'Workflow Administration'. The main area has a title bar 'Workflow - Server Query Frame'. Below the title bar, there's a section 'Workflows on http://' with a help icon. The text says: 'Select the Workflows to manage.' Below this, there's a dropdown for 'Previously defined filters:' with the value '<No previously defined filters>' and a note '(selecting one of these will populate the remaining fields)'. Below the dropdown are two radio buttons: 'Show all Workflows' (selected) and 'Show Workflows with'. Below the radio buttons is a text field for 'Select what to filter for' with the value '<Select what to filter for>' and a dropdown for 'Equals'. At the bottom are 'OK' and 'Cancel' buttons.

18.2.2 Finding Workflows that Match Search Criteria

If the target workflow server is running a large number of workflow processes, you might want to filter the list of workflows you see in iManager. To do this, you can specify search criteria.

- 1 Select *Show Workflows with*.



By default, *Show all Workflows* 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	Username of the requestor.
Recipient	Username 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.

Operator	Comment
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 *Object History* or *Object Selector* to specify a value. For all other attributes, select the value from the drop-down list.

- 5 Click *OK*.

iManager displays the workflows you have selected on the Workflows panel.

Changing the target server and filter. When you have 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, then choose *Select Server* from the *Actions* menu.

To specify different search criteria, choose *Define Filter* on the *Actions* menu.

18.2.3 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 should refresh the list of active workflows running on the server.

- 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:
 - ♦ Refresh Off
 - ♦ Refresh Now
 - ♦ 10 seconds
 - ♦ 30 seconds
 - ♦ 60 seconds
 - ♦ 5 minutes
- 3 Click *OK*.

Sorting the List of Workflows

If you have a large number of request definitions, you might want to sort the list by a particular column, such as *Name* or *Description*.

- 1 Click the heading for the sort column.

18.2.4 Terminating a Workflow Instance

If you do not want a workflow instance to continue its processing, you can terminate the workflow.

- 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.

18.2.5 Viewing Details about a Workflow Instance

When you have 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 shows a single activity as current because only one user can act on the work item at any point in time. However, if the workflow handles parallel processing and branching, there might 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.

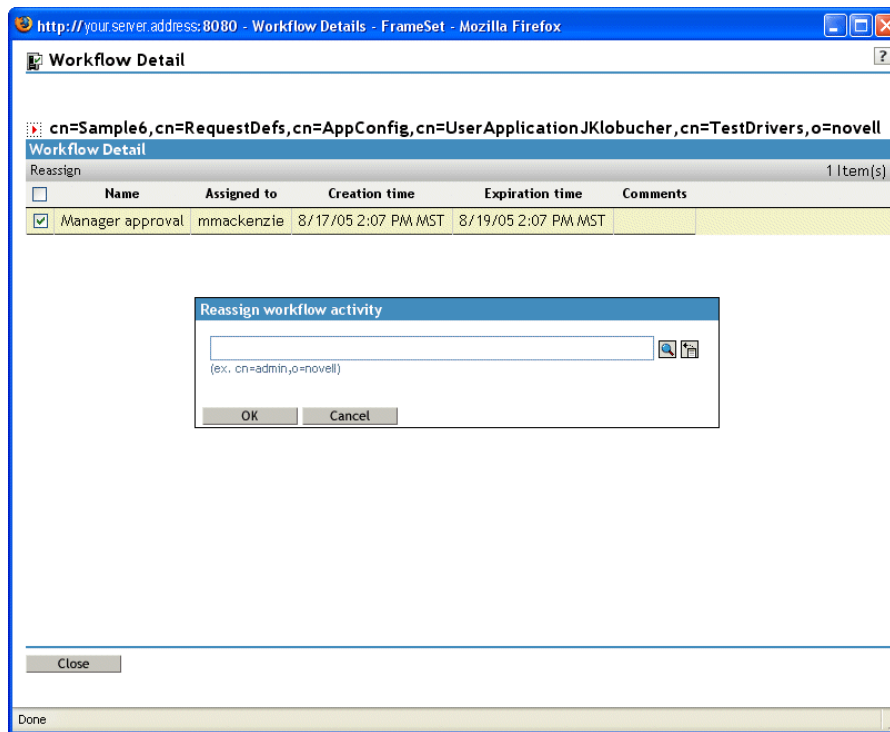
Workflow Detail					
Reassign					
	Name	Assigned to	Creation time	Expiration time	Comments
<input type="checkbox"/>	Second approval	kkilpatrick	3/6/07 10:38 AM EST	3/8/07 10:38 AM EST	Assigned
<input type="checkbox"/>	First approval	kkilpatrick	3/6/07 10:38 AM EST	3/8/07 10:38 AM EST	User task assigned to reviewer Kelly Kilpatrick

18.2.6 Reassigning a Workflow Instance

If a workflow instance has stopped and cannot be restarted, you can reassign the work item to another user or group.

- 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 work item.

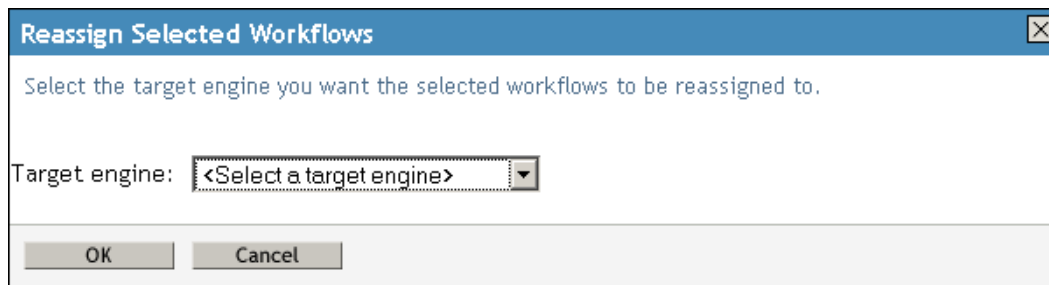
18.2.7 Managing Workflow Processes in a Cluster

You can use the Workflows screen to reassign processes from one workflow engine to another. For example, you could use this feature to reassign processes back to a failed workflow engine when the workflow engine is brought back online, or you could redistribute processes to other engines when an engine is permanently removed from the cluster.

The source engine(s) must be in a SHUTDOWN or TIMEDOUT state. The target engine must be restarted in order to restart the processes that were reassigned to that engine.

Reassigning a Process from One Workflow Engine to Another

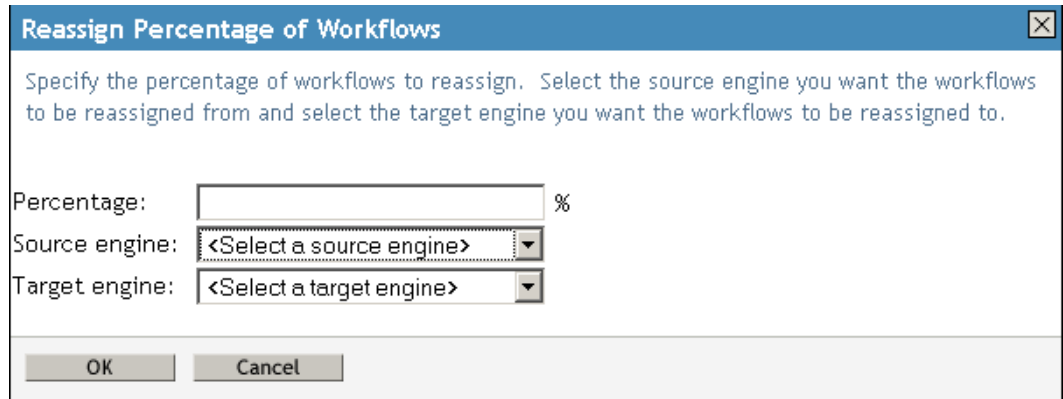
- 1 In the Workflows panel, select the workflow that you would like to reassign by clicking the check box next to the workflow name.
- 2 Select *Actions > Reassign*.



- 3 Select the workflow engine to which you want to reassign the workflow process from the *Target Engine* list.
- 4 Click *OK*.

Reassigning a Percentage of Processes from One Workflow Engine to Another

- 1 In the Workflows panel, select the workflow that you would like to reassign by clicking the check box next to the workflow name.
- 2 Select *Actions > Reassign Percentage*.



The dialog box is titled "Reassign Percentage of Workflows" and contains the following elements:

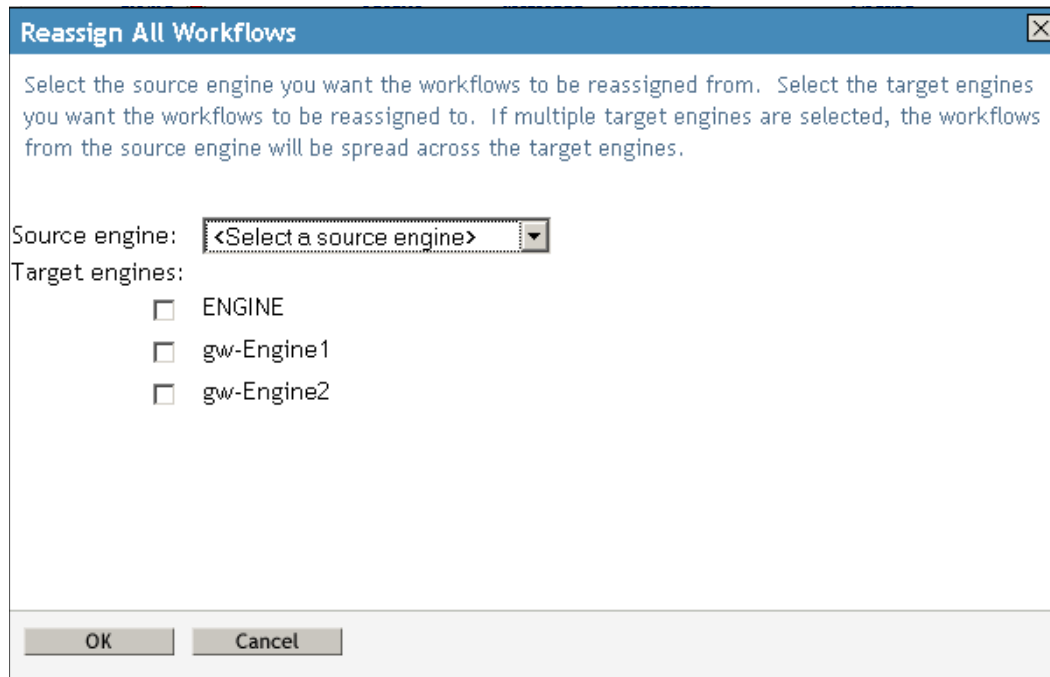
- A text instruction: "Specify the percentage of workflows to reassign. Select the source engine you want the workflows to be reassigned from and select the target engine you want the workflows to be reassigned to."
- A "Percentage:" label followed by a text input field and a "%" symbol.
- A "Source engine:" label followed by a dropdown menu with the text "<Select a source engine>".
- A "Target engine:" label followed by a dropdown menu with the text "<Select a target engine>".
- At the bottom, there are two buttons: "OK" and "Cancel".

- 3 In the *Percentage* field, type the percentage of workflow processes that you would like to reassign from one workflow engine to another.
- 4 Use the *Source engine* list to select the workflow engine from which you want to reassign processes.
- 5 Use the *Target engine* field to select the workflow engine to which you want to reassign processes.
- 6 Click *OK*.

Reassigning All Processes from One Workflow Engine to Another

- 1 In the Workflows panel, select the workflow that you would like to reassign by clicking the check box next to the workflow name.

2 Select *Actions > Reassign All*.



The dialog box is titled "Reassign All Workflows" and contains the following elements:

- Instructional text: "Select the source engine you want the workflows to be reassigned from. Select the target engines you want the workflows to be reassigned to. If multiple target engines are selected, the workflows from the source engine will be spread across the target engines."
- Source engine: A dropdown menu with the text "<Select a source engine>".
- Target engines: A list of three items, each with a checkbox:
 - ☐ ENGINE
 - ☐ gw-Engine1
 - ☐ gw-Engine2
- Buttons: "OK" and "Cancel" at the bottom.

3 Use the *Source engine* list to select the workflow engine from which you want to reassign processes.

4 Select the workflow engines to which you would like to reassign processes by clicking the check box next to the name of the workflow engine.

If you select multiple target engines, the processes from the source engine will be evenly distributed to the target engine.

5 Click *OK*.

18.3 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 panel.

The screenshot shows the Novell iManager interface. On the left, a navigation pane lists various administrative tasks, with 'Email Server Options' highlighted under the 'Workflow Administration' section. The main window displays the 'Email Server Options' configuration panel. It includes a title bar, a toolbar, and a main content area. The content area has a heading 'Email Server Options' and a sub-heading 'Enter the settings for your e-mail notification server.' Below this, there are input fields for 'Host Name' (containing 'mail.novell.com') and 'From' (containing 'spitfire@novell.com'). There is also a checkbox for 'Authenticate to server using credentials' which is currently unchecked. Below the checkbox are three input fields for 'User Name', 'Password', and 'Retype password'. At the bottom of the panel are 'OK' and 'Cancel' buttons.

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 in order 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 username and password.

7 When you are finished, click *OK*.

18.4 Working with E-Mail Templates

Identity Manager includes e-mail notification templates that are designed specifically for workflow-based provisioning. These e-mail templates include the following.

- ◆ *New Provisioning Request* (Provisioning Notification)
- ◆ *Availability Setting Notification* (Availability)
- ◆ *Delegate Assignment Notification* (Delegate)
- ◆ *Provisioning Approval Notification* (Provisioning Approval Completed Notification)
- ◆ *Reminder - A request is waiting on your approval* (Provisioning Reminder)
- ◆ *Proxy Assignment Notification* (Proxy)

The subject lines are listed first above. The template names (as they appear in iManager and Designer) are given in parentheses.

You can edit the templates to change the content and format of e-mail messages. You can also create new templates. If you create new templates, you need to follow these naming conventions.

- ♦ The language-independent version of the Provisioning Notification template can have any name you like. The default template for notification e-mail messages is called:

Provisioning Notification

- ♦ The language-independent version of the Provisioning Reminder template can have any name you like. The default template for reminder e-mail messages is called:

Provisioning Reminder

- ♦ Each delegation template must have a name that begins with the word:

delegate

The language-independent name can be followed by one or more characters that describe the purpose or content of the template.

- ♦ Each proxy template must have a name that begins with the word:

proxy

The language-independent name can be followed by one or more characters that describe the purpose or content of the template.

- ♦ Each availability template must have a name that begins with the word:

availability

The language-independent name can be followed by one or more characters that describe the purpose or content of the template.

Each language-specific version of a template must have a suffix that provides a language code (for example, `_fr` for French, `_es` for Spanish, and so forth).

To create or edit an e-mail template, 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.

You also can create and edit e-mail templates in Designer.

When you create a User Application driver in iManager or Designer, any e-mail notification templates that are missing from the standard set of e-mail notification templates are replaced. Existing e-mail notification templates are not updated. This is to prevent overwriting e-mail notification templates that you have customized. You can update existing e-mail notification templates manually using Designer (see “[About E-Mail Notification Templates](#)” in the *Identity Manager 3.5 User Application: Design Guide*). For more information about e-mail notification templates, see “Setting up E-Mail Notification Templates” in the *Novell Designer 2.0 for Identity Manager 3.5 Administration Guide*.

18.4.1 Default Content and Format

This section shows you what the content of the e-mail templates looks like after you install the product. It also describes the replacement tags that can be used in the e-mail template.

New Provisioning Request

This 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.

Hi,

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\$.

Table 18-2 *New Provisioning Request Template: Replacement Tags*

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. For information about setting the value for this parameter, see Section 18.4.3, “Modifying Default Values for the Template,” on page 345.
\$PORT\$	The port for the Identity Manager User Application. For information about setting the value for this parameter, see Section 18.4.3, “Modifying Default Values for the Template,” on page 345.
\$SECURE_PORT\$	The secure port for the Identity Manager User Application. For information about setting the value for this parameter, see Section 18.4.3, “Modifying Default Values for the Template,” on page 345.
\$TASKLIST_CONTEXT\$	The page that displays the list of all requests pending for the addressee.
\$TASK_DETAILS\$	The page that displays details for the request for which this e-mail message was generated.

Availability Setting Notification

This template identifies a user whose availability has been updated. It includes the start time and expiration time of the period for which the user is unavailable, and the resources for which the user is unavailable.

Hi,

\$submitterFirstName\$ \$submitterLastName\$ has updated availability settings for

\$userFirstName\$ \$userLastName\$.

This user has \$operation\$ an availability setting that applies to the following resources:

\$resources\$

This setting indicates that \$userFirstName\$ \$userLastName\$ is unavailable to work on these resources during the timeframe outlined below:

Start time: \$startTime\$

Expiration time: \$expirationTime\$

When a user is unavailable, any delegates assigned may handle resource requests for that user.

You can review a list of your availability settings at \$PROTOCOL\$://\$HOST\$: \$PORT\$/\$AVAILABILITY_CONTEXT\$.

Table 18-3 Availability Setting Notification Template: Replacement Tags

Tag	Description
\$submitterFirstName\$	The first name of the user who updated the availability setting.
\$PROTOCOL\$	The protocol for URLs included in the e-mail message.
\$PORT\$	The port for the Identity Manager User Application. For information about setting the value for this parameter, see Section 18.4.3, “Modifying Default Values for the Template,” on page 345.
\$startTime\$	The start time of the workflow for this provisioning request.
\$resources\$	The resources (provisioning requests) for which the addressee is unavailable.
\$SECURE_PROTOCOL\$	The secure protocol for URLs included in the e-mail message.

Tag	Description
\$expirationTime\$	The time at which the availability will expire.
\$submitterLastName\$	The last name of the user who updated the availability setting.
\$SECURE_PORT\$	The secure port for the Identity Manager User Application. For information about setting the value for this parameter, see Section 18.4.3, “Modifying Default Values for the Template,” on page 345 .
\$userFirstName\$	The first name of the user to whom this availability setting applies.
\$userLastName\$	The last name of the user to whom this availability setting applies.
\$HOST\$	The host for the JBoss application server that is running the Identity Manager User Application. For information about setting the value for this parameter, see Section 18.4.3, “Modifying Default Values for the Template,” on page 345 .
\$ASSIGNMENT_LIST_CONTEXT\$	The context or path of the URL to the provisioning User Application.

Delegate Assignment Notification

This template notifies a user when a provisioning request has been submitted that requires the user’s approval. It includes the name of the request, the user who submitted the request, and the full name of the recipient. It includes links for viewing the provisioning request and for viewing all provisioning requests awaiting the user’s approval.

Hi,

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\$.

_SUBJECT

Table 18-4 *Delegate Assignment Notification: Replacement Tags*

Tag	Description
<code>\$submitterFirstName\$</code>	The first name of the user who assigned the delegate.
<code>\$PROTOCOL\$</code>	The protocol for URLs included in the e-mail message.
<code>\$PORT\$</code>	The port for the Identity Manager User Application. For information about setting the value for this parameter, see Section 18.4.3, "Modifying Default Values for the Template," on page 345.
<code>\$resources\$</code>	The resources (provisioning requests) for which the delegate is available.
<code>\$SECURE_PROTOCOL\$</code>	The secure protocol for URLs included in the e-mail message.
<code>\$fromUsers\$</code>	The users for which the assigned delegate is authorized to handle resource requests.
<code>\$relationship\$</code>	The relationship defined in the directory abstraction layer that was selected for this delegate assignment.
<code>\$expirationTime\$</code>	The time at which the delegate assignment will expire.
<code>\$fromContainers\$</code>	The containers for which the assigned delegate is authorized to handle resource requests.
<code>\$fromGroups\$</code>	The groups for which the assigned delegate is authorized to handle resource requests.
<code>\$submitterLastName\$</code>	The last name of the user who assigned the delegate.
<code>\$SECURE_PORT\$</code>	The secure port for the Identity Manager User Application. For information about setting the value for this parameter, see Section 18.4.3, "Modifying Default Values for the Template," on page 345.
<code>\$userFirstName\$</code>	The first name of the user who has been assigned as a delegate.
<code>\$userLastName\$</code>	The last name of the user who has been assigned as a delegate.
<code>\$HOST\$</code>	The host for the JBoss application server that is running the Identity Manager User Application. For information about setting the value for this parameter, see Section 18.4.3, "Modifying Default Values for the Template," on page 345.
<code>\$ASSIGNMENT_LIST_CONTEXT\$</code>	The context or path of the URL to the provisioning User Application.

Provisioning Approval Notification

This template notifies a user when an approval process for a provisioning request submitted by the user has been completed.

Hi,

The approval process of your provisioning request has completed.

Request name: \$requestTitle\$
Request id: \$requestId\$
Submitted by: \$initiatorFullName\$
Submitted on: \$requestSubmissionTime\$
Recipient: \$recipientFullName\$

Status: \$requestStatus\$

Table 18-5 *Provisioning Approval Notification: Replacement Tags*

Tag	Description
\$initiatorFullName\$	The full name of the initiator.
\$requestSubmissionTime\$	The time at which the request was submitted.
\$requestTitle\$	The display name of the provisioning request definition.
\$requestId	The ID of the provisioning request.
\$recipientFullName\$	The full name of the recipient.

Reminder - A Request Is Waiting on Your Approval

This template reminds a user that a provisioning request that requires the user's approval is waiting in a queue for approval. It includes the name of the request, the user who submitted the request, and the recipient. It includes links for viewing the provisioning request and for viewing all provisioning requests awaiting the user's approval.

Hi,

This is a reminder that a provisioning request is sitting in your queue waiting on 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 `$PROTOCOL$://$HOST$: $PORT$/$TASKLIST_CONTEXT$`.

Table 18-6 *Reminder - A request is waiting on your approval: Replacement Tags*

Tag	Description
<code>\$TASKLIST_CONTEXT\$</code>	The page that displays the list of all requests pending for the addressee.
<code>\$PROTOCOL\$</code>	The protocol for URLs included in the e-mail message.
<code>\$PORT\$</code>	The port for the Identity Manager User Application. For information about setting the value for this parameter, see Section 18.4.3, “Modifying Default Values for the Template,” on page 345.
<code>\$SECURE_PROTOCOL\$</code>	The secure protocol for URLs included in the e-mail message.
<code>\$initiatorFullName\$</code>	The full name of the initiator.
<code>\$recipientFullName\$</code>	The full name of the recipient.
<code>\$TASK_DETAILS\$</code>	The page that displays details for the request for which this e-mail message was generated.
<code>\$SECURE_PORT\$</code>	The secure port for the Identity Manager User Application. For information about setting the value for this parameter, see Section 18.4.3, “Modifying Default Values for the Template,” on page 345.
<code>\$userFirstName\$</code>	The first name of the addressee.
<code>\$HOST\$</code>	The host for the JBoss application server that is running the Identity Manager User Application. For information about setting the value for this parameter, see Section 18.4.3, “Modifying Default Values for the Template,” on page 345.
<code>\$requestTitle\$</code>	The display name of the provisioning request definition.

Proxy Assignment Notification

This template notifies the recipient that a proxy has been assigned. The user who has been assigned as a proxy is identified, as are the users, groups, and containers for which the user is authorized to act as proxy. It includes links for viewing the recipient’s list of proxy assignments.

Hi,

A proxy assignment that authorizes a user to act as proxy for one or more users, groups, or containers was `$operation$` by: `$submitterFirstName$ $submitterLastName$`.

Unlike delegate assignments, proxy assignments are independent of resource requests, and therefore apply to all work and settings actions.

The user selected as proxy is:

`$userFirstName$ $userLastName$`

The assigned proxy is authorized to handle all work for these users, groups, and containers:

Users: `$fromUsers$`

Groups: `$fromGroups$`

Containers: `$fromContainers$`

This proxy assignment expires at:

`$expirationTime$`

You can review a list of your proxy assignments at `$PROTOCOL$://$HOST$: $PORT$/$PROXY_CONTEXT$`.

Table 18-7 Proxy Assignment Notification: Replacement Tags

Tag	Description
<code>\$submitterFirstName\$</code>	The first name of the user who assigned the proxy.
<code>\$PROTOCOL\$</code>	The protocol for URLs included in the e-mail message.
<code>\$PORT\$</code>	The port for the Identity Manager User Application. For information about setting the value for this parameter, see Section 18.4.3, “Modifying Default Values for the Template,” on page 345.
<code>\$resources\$</code>	The resources (provisioning requests) for which the proxy is available.
<code>\$SECURE_PROTOCOL\$</code>	The secure protocol for URLs included in the e-mail message.
<code>\$fromUsers\$</code>	The users for which the assigned proxy is authorized to handle resource requests.
<code>\$expirationTime\$</code>	The time at which the proxy assignment will expire.
<code>\$fromContainers\$</code>	The containers for which the assigned proxy is authorized to handle resource requests.
<code>\$fromGroups\$</code>	The groups for which the assigned proxy is authorized to handle resource requests.

Tag	Description
\$submitterLastName\$	The last name of the user who assigned the proxy.
\$SECURE_PORT\$	The secure port for the Identity Manager User Application. For information about setting the value for this parameter, see Section 18.4.3, “Modifying Default Values for the Template,” on page 345.
\$userFirstName\$	The first name of the user who has been assigned as a proxy.
\$userLastName\$	The last name of the user who has been assigned as a proxy.
\$HOST\$	The host for the JBoss application server that is running the Identity Manager User Application. For information about setting the value for this parameter, see Section 18.4.3, “Modifying Default Values for the Template,” on page 345.
\$ASSIGNMENT_LIST_CONTEXT\$	The context or path of the URL to the provisioning User Application.

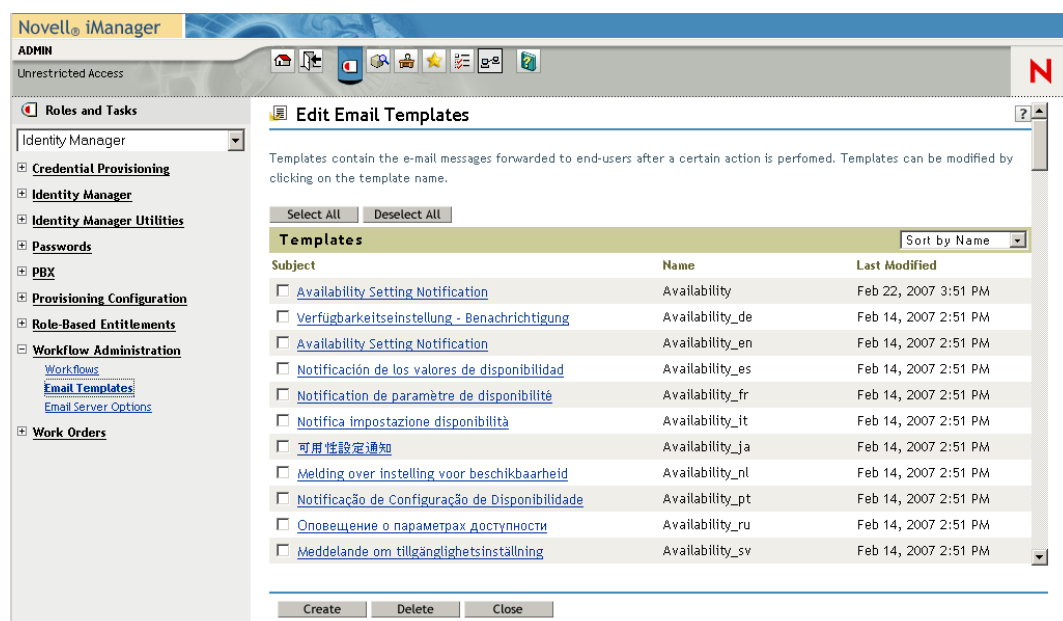
18.4.2 Editing E-mail Templates

You can change the content or format of the supplied e-mail templates. For information about creating e-mail templates, see “Configuring E-Mail Notification” in the *Novell Identity Manager Administration Guide*.

To edit a template:

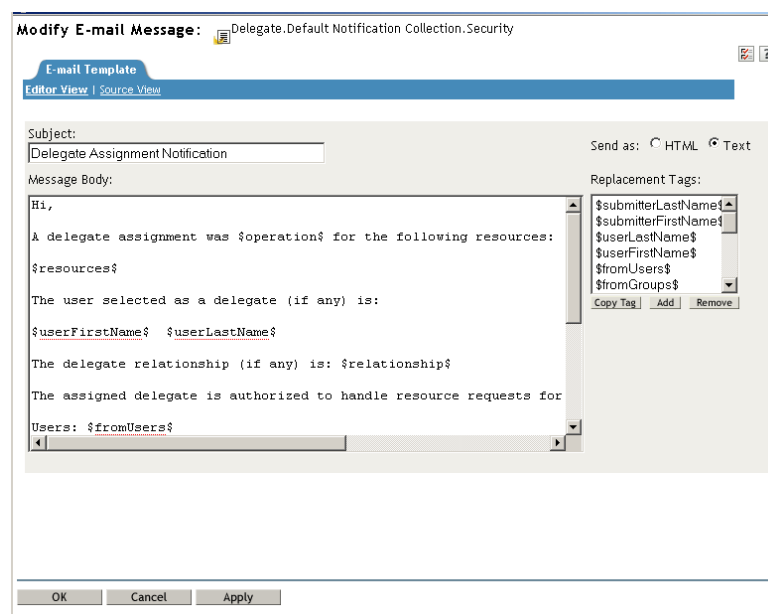
- 1 Select the *Identity Manager* category in iManager.
- 2 Open the *Workflow Administration* role.
- 3 Click the *Email Templates* task.

iManager displays the *Edit Email Templates* panel.



- Click the name of the e-mail template that you would like to edit.

iManager displays the *Modify E-mail Message* screen.



- Make your changes in the *Message Body* box.
- If necessary, copy one or more of the supplied tags in the *Replacement Tags* list to include dynamic text in the message body.

For a description of the replacement tags, see [Section 18.4.1, “Default Content and Format,” on page 335](#).

- When you are finished, click *OK*.

18.4.3 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 have completed the installation, you can also modify these values by using the User Application Configuration tool.

- 1 Run the `configupdate.sh` script in the `idm` folder.

```
./configupdate.sh
```

On Windows, run `configupdate.bat`.

The screenshot shows the 'User Application Configuration' window. It has a blue header bar with the title 'User Application Configuration'. Below the header are four main sections, each with a title bar and a list of configuration fields. The first section, 'eDirectory Connection Settings', contains fields for LDAP Host, LDAP Non-Secure Port, LDAP Secure Port, LDAP Administrator, LDAP Administrator Password, Use Public Anonymous Account, LDAP Guest, LDAP Guest Password, Secure Admin Connection, and Secure User Connection. The second section, 'eDirectory DNs', contains fields for Root Container DN, Provisioning Driver DN, User Application Admin, Provisioning Application Admin, User Container DN, and Group Container DN. The third section, 'eDirectory Certificates', contains fields for Keystore Path, Keystore Password, and Confirm Keystore Password. The fourth section, 'Email', contains a field for Notify Template Host Token. At the bottom of the window are three buttons: 'OK', 'Cancel', and 'Show Advanced Options'.

- 2 Make changes as necessary to any of the following fields:

Field	Description
Email Notify Host	Used to replace the <code>\$HOST\$</code> 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 <code>\$PORT\$</code> token in e-mail templates used in approval flows.
Email Notify Secure Port	Used to replace the <code>\$SECURE_PORT\$</code> token in e-mail templates used in approval flows.

- 3 Click *OK* to confirm your changes.

18.4.4 Adding Localized E-Mail Templates

To add localized e-mail templates:

- 1 Select the *Identity Manager* category in iManager.
- 2 Open the *Workflow Administration* role.
- 3 Click the *Email Templates* task.
iManager displays the *Edit Email Templates* panel.
- 4 Identify the e-mail template (without any locale in the name) that you want to copy.
 - 4a Write down the template name to use in [Step 5](#).
 - 4b Click the template subject to open the template and view its message subject, body, and replacement tags.
 - 4c Copy the message subject, body (to be translated), and replacement tags that you want to use in your new template.
 - 4d Click *Cancel*.
- 5 Click *Create*, then enter the template name with a locale extension. For example, to create a Forgot Hint template in German, enter the name Forgot Hint_de, where _de signifies Deutsch (German).
If you use a two-letter language and two-letter country code, this works fine. If you attempt to use a locale with a variant such as en_US_TX, only the variant and language are considered. Do not use locale variants when naming e-mail templates.
- 6 Click *OK*.
- 7 In the template list, click the newly created template, for example Forgot Hint_de, and enter the translated subject and message body. Be sure to preserve the replacement tags surrounded by the dollar (\$) sign in the message body.
- 8 If necessary, copy one or more of the supplied tags in the *Replacement Tags* list to include dynamic text in the message body.
For a description of the replacement tags, see [Section 18.4.1, “Default Content and Format,” on page 335](#).
- 9 Click *Apply*.
- 10 Click *OK*.

NOTE: E-mail templates only send localized content if the preferred locale is set for the user (to whom the mail is sent).

Configuring Provisioning Teams

19

This section describes how to use the iManager plug-ins to manage provisioning teams. Topics include:

- ♦ [Section 19.1, “About the Provisioning Teams Plug-Ins,” on page 347](#)
- ♦ [Section 19.2, “Managing Provisioning Teams,” on page 349](#)
- ♦ [Section 19.3, “Managing Provisioning Team Request Rights,” on page 358](#)
- ♦ [Section 19.4, “Creating a Team to Manage Direct Reports,” on page 366](#)

19.1 About the Provisioning Teams Plug-Ins

The *Requests & Approvals* tab in the Identity Manager User Application includes a group of actions called *My Team’s Work*. The *My Team’s Work* actions allow you to work with team member tasks and requests in a workflow.

To configure provisioning teams, you need to use the Provisioning Team and Provisioning Team Requests plug-ins to iManager. The Provisioning Team plug-in lets you define the characteristics of a team. The Provisioning Team Requests plug-in lets you specify the request rights for a team.

NOTE: A team requests object must be defined for each team definition. Any provisioning team without a team requests object will not be available for use within the User Application.

You can find the Provisioning Team and Provisioning Team Requests plug-ins in the Identity Manager category in iManager. These plug-ins are listed under the Provisioning Configuration role.

19.1.1 About Teams

A *team* identifies a group of users and determines who can manage provisioning requests and approval tasks associated with this team. The team definition consists of a list of team managers, team members, and team options, as described below:

- ♦ The *team managers* are those users who can administer requests and tasks for the team. Team managers can also be given permission to set proxies and delegates for team members. Team managers can be users or groups.
- ♦ The *team members* are those users who are allowed to participate on the team. Team members can be users, groups, or containers within the directory. Alternatively, they can be derived through directory relationships. For example, the list of members could be derived by the manager-employee relationship within the organization. In this case, the team members would be all users that report to the team manager.

NOTE: The Provisioning Application Administrator can configure the directory abstraction layer to support cascading relationships, in which case several levels within an organization can be included within a team. The number of levels to include is configurable by the administrator.

- ♦ The *team options* determine the provisioning request scope, which specifies whether the team can act on an individual provisioning request, one or more categories of requests, or all

requests. The team options also determine whether team managers can set proxies for team members or set the availability of team members for the purpose of delegation.

NOTE: The User Application only supports a single level for proxy assignments. Proxy assignments are not propagated to multiple levels.

The Provisioning Application Administrator can perform all team management functions.

The team definition itself is managed within iManager by one or more administrative managers.

Distinction between teams and groups Although a team can sometimes refer to a group in the Identity Vault, a team is not the same thing as a group. When you define a group in the Identity Vault, you identify a set of users that have something in common. However, the group does not automatically have the capabilities of a team within the User Application. To take advantage of the team capabilities within the User Application, you must define a team that points to the group.

19.1.2 About Team Request Rights

The *team requests object* specifies a list of requests that fall within the domain of a team, as well as the rights given to the team managers. The request rights specify the actions that team managers can perform on the provisioning requests and tasks.

The team definition has a *one-to-many relationship* with the team requests object. This means that each team must have at least one team requests object associated with it, but can have more than one team requests object. Each team requests object is associated with only one team definition.

The following task scope options are configurable for team managers:

- ♦ The ability to act on tasks where the team member is an addressee
- ♦ The ability to act on tasks where the team member is a recipient

WARNING: For security reasons, the recipient task scope option is disabled by default. Giving a team manager the ability to act on tasks where the recipient of the request is a team member can raise several security issues. First, the manager is then able to view data included on any of the forms that are displayed during the course of workflow execution, regardless of his or her trustee rights. Second, depending on the permission options (see below), a team manager could circumvent the approval process by claiming or approving the task, or by reassigning it to someone else.

If both of the task scope options described above are disabled, the team manager cannot view or act on any active requests. Therefore, team managers will typically want to have at least one of these options enabled.

The following permission options are configurable for team managers:

- ♦ The ability to initiate a provisioning request on behalf of a team member.
- ♦ The ability to retract a provisioning request on behalf of a team member.
- ♦ The ability to make a team member a delegate for other team members' provisioning requests.
- ♦ The ability to claim a task for a team member who is a recipient or addressee (based on the task scope).
- ♦ The ability to reassign a task for a team member who is a recipient or addressee (based on the task scope).

NOTE: The User Application only supports a single level for delegate assignments. Delegate assignments are not propagated to multiple levels.

The trustee rights defined for a provisioning request apply to team managers who want to initiate a request on behalf of their team members.

19.1.3 Using a Team to Manage Direct Reports

You can define a team that allows managers throughout an organization to control the provisioning environment for their direct reports. If defined properly, a single team definition can be used to allow *all* managers to control the activities of their direct reports, thereby removing the need to define a separate team for each reporting relationship.

Here are the basic requirements for a team that supports direct reports within an organization:

- ♦ The members of the team are defined by the Manager-Employee relationship.
- ♦ The managers of the team are defined by a dynamic group that searches subcontainers, using a search filter that retrieves only the managers.

After the team has been defined, the User Application allows all managers to use the team management actions within the navigation menu. This gives the managers the ability to control the provisioning activities that their direct reports can perform.

For details on how to define a team to manage direct reports, see [Section 19.4, “Creating a Team to Manage Direct Reports,” on page 366](#).

NOTE: This technique replaces the notion of an organizational team supported in earlier releases of the Identity Manager User Application.

19.2 Managing Provisioning Teams

Before configuring a provisioning team, you need to select the Identity Manager User Application driver that contains the definition. After selecting the driver, you can create a new team definition, edit an existing definition, or delete an existing definition.

The *Provisioning Teams* and *Provisioning Team Requests* tasks use the same driver as the *Provisioning Requests* task.

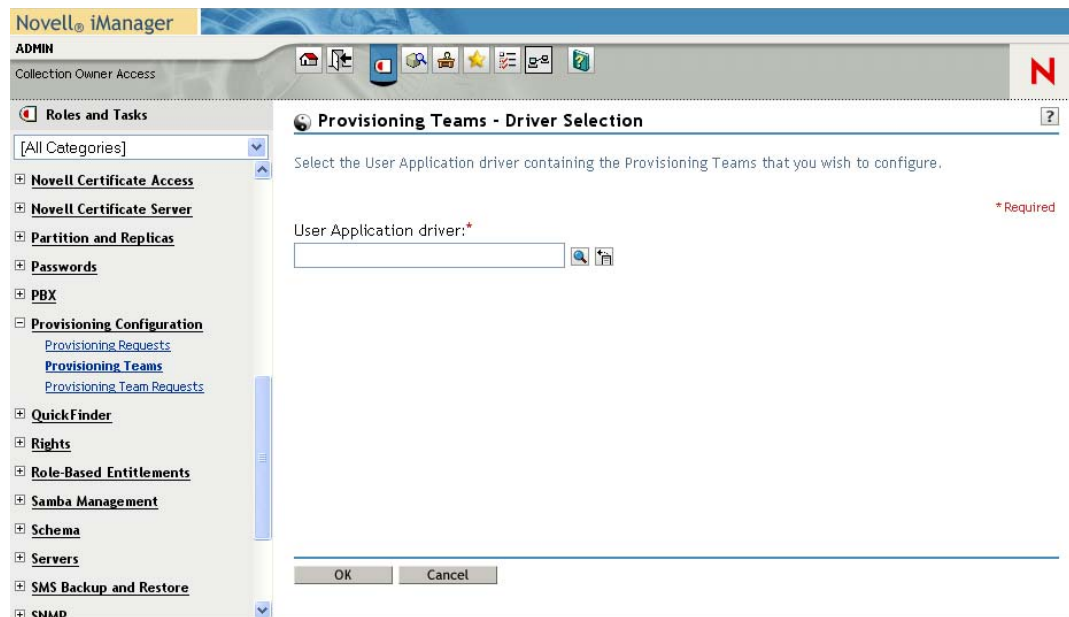
19.2.1 Selecting the Driver

After you have selected a User Application driver for the *Provisioning Requests*, *Provisioning Teams*, or *Provisioning Team Requests* task, you don't have to select the User Application driver again during this iManager session.

To select a User Application driver:

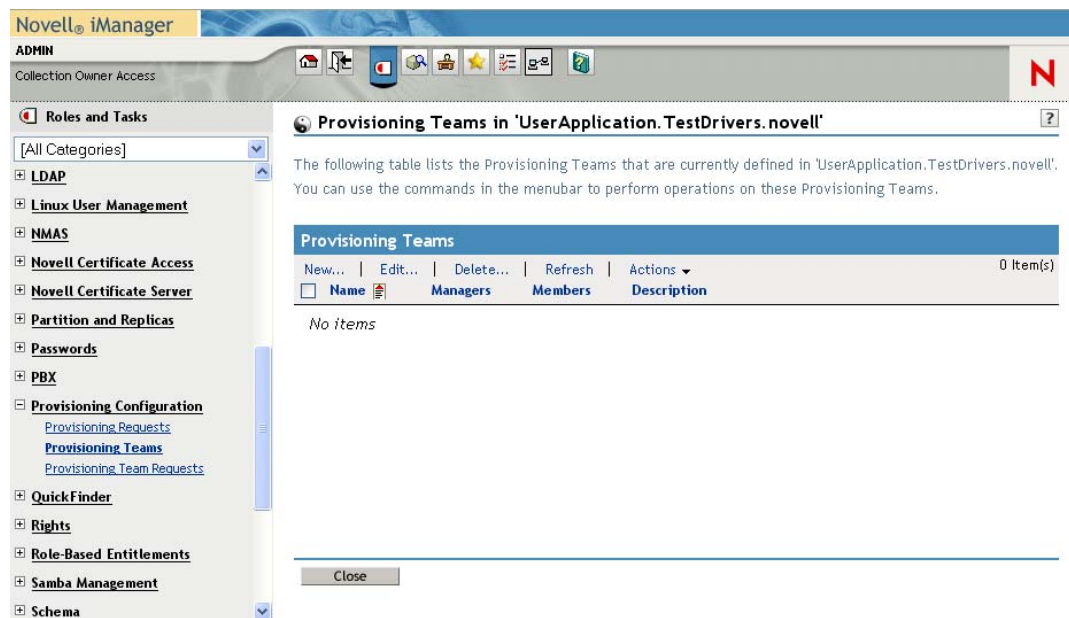
- 1 Select the *Identity Manager* category in iManager.
- 2 Open the *Provisioning Configuration* role.
- 3 Click the *Provisioning Teams* task.

iManager displays the User Application Driver panel.



- 4 Specify the driver name in the *User Application Driver* field, then click *OK*.

iManager displays the Provisioning Teams panel. The Provisioning Teams panel displays a list of existing team definitions.



Changing the driver. When you have 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.

19.2.2 Creating or Editing a Provisioning Team

To create a new provisioning team:

- 1 Click the *New* command in the Provisioning Teams panel.



The first page of the Create New Provisioning Team wizard displays.

https://sigmund.qalab.wal.novell.com - Provisioning Team Configuration Wizard - FrameSet - Mozilla Firefox

Create New Provisioning Team

Step 1 of 5: Edit general Provisioning Team information.

Enter the name for the new Provisioning Team. Enter the display names and descriptions for the defined languages. English will be displayed for undefined languages.

Name (CN):

Provisioning Team Localized Strings

Add... Delete...

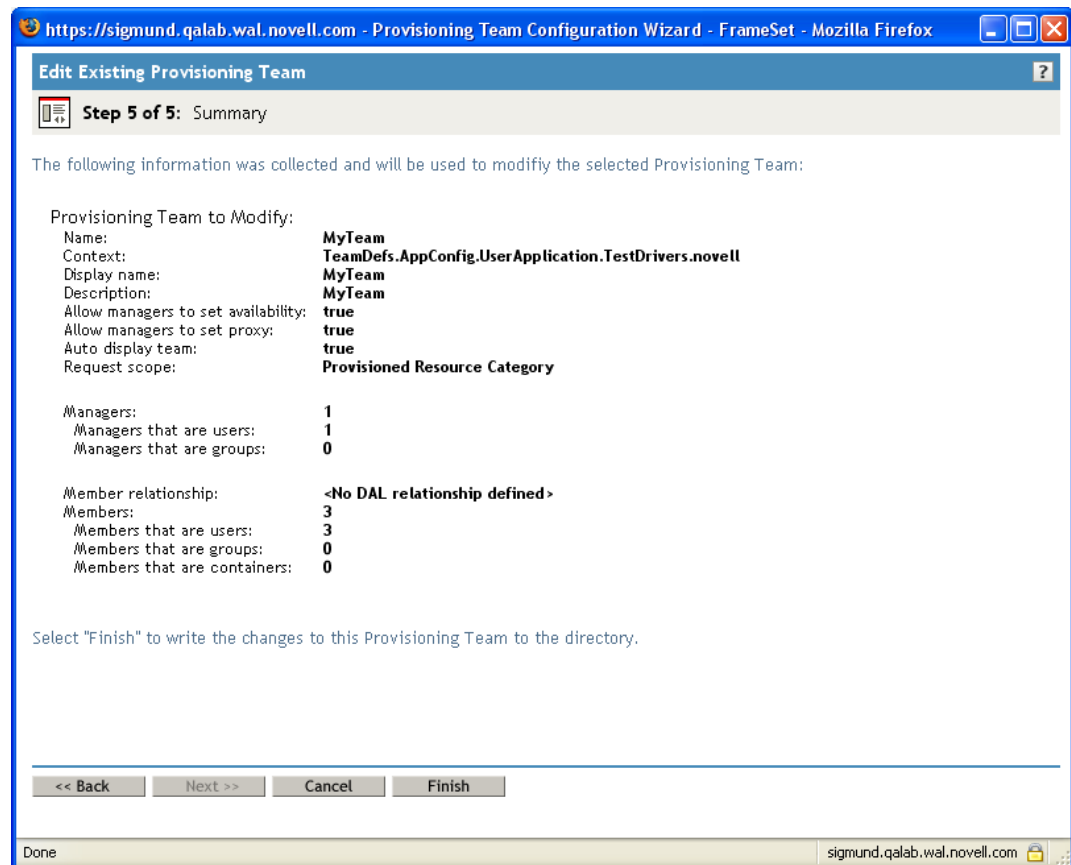
Language	Display name	Description
<input type="checkbox"/> English		

<< Back Next >> Cancel Finish

Done sigmund.qalab.wal.novell.com

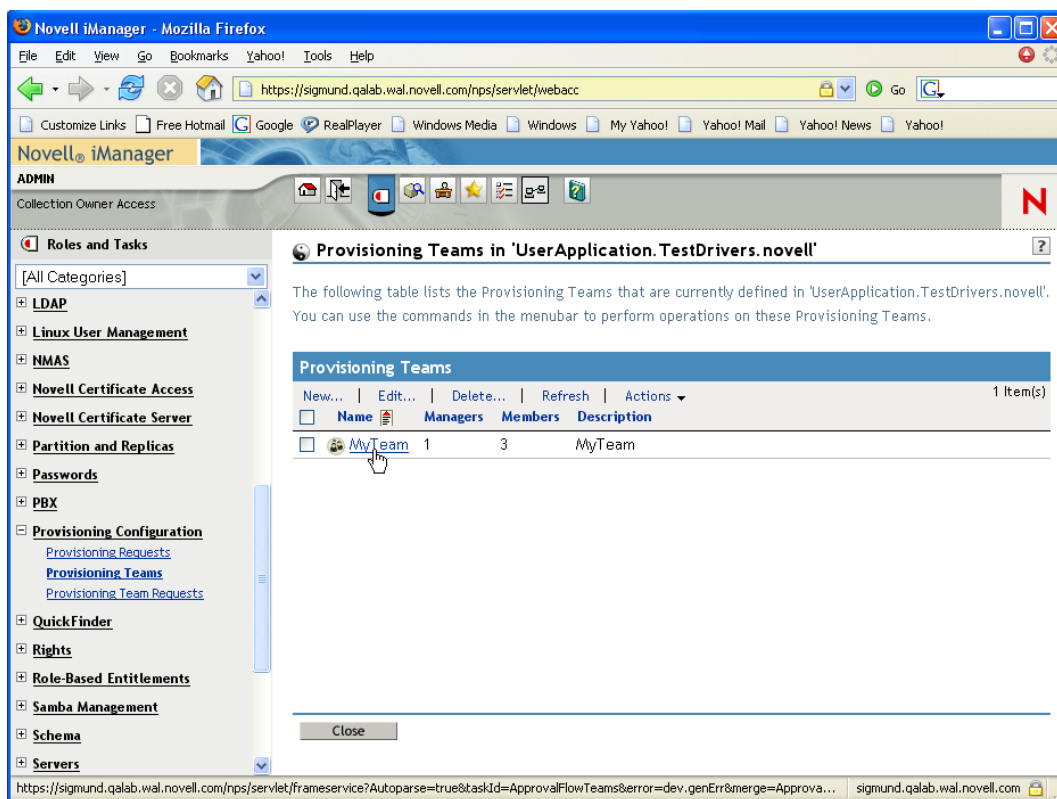
- 2 Type a common name for the new object in the *Name (CN)* field.
 - 3 For each language you want to support in your application, type the localized text in the *Display Name* and *Description* fields under *Provisioning Team Localized Strings*. This text is used to identify the provisioning team throughout the User Application.
 - 4 To add a new language to the list, click *Add*, then select the desired language.
- By default, a newly created provisioning team supports only English.

- 5 Click *Next*.
- 6 Specify the managers for the team, as described in “Specifying the Team Managers” on page 354.
- 7 Specify the members of the team, as described in “Specifying the Team Members” on page 355.
- 8 Specify the team options for the team, as described in “Specifying the Team Options” on page 356.
- 9 Review your settings, then click *Finish*.



To edit an existing provisioning team:

- 1 Click the name of the provisioning team in the Provisioning Teams panel.



If you have a large number of team definitions, you might want to sort the list by a particular column, such as the *Name* or *Description*. To sort by a particular column, click the column heading.

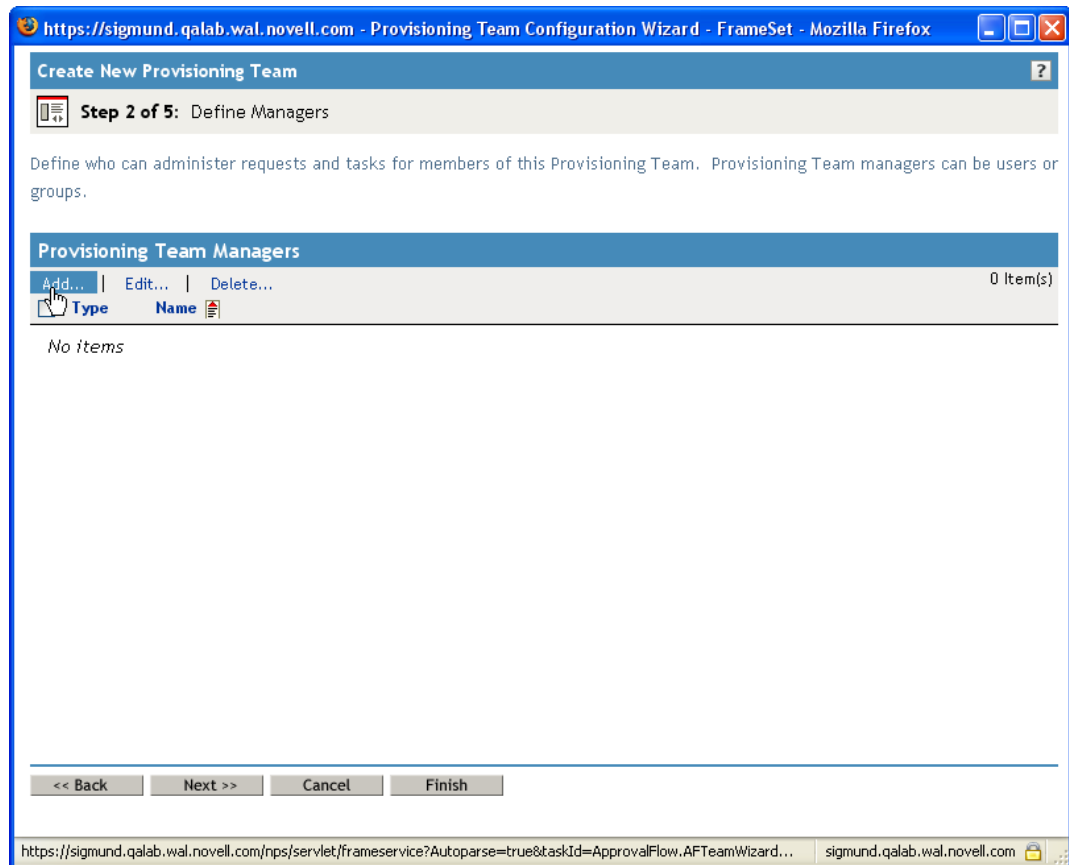
- 2 For each language you want to support in your application, click the check box beside the language in the list under *Provisioning Team Localized Strings*, and type the localized text in the *Display Name* and *Description* fields. This text is used to identify the provisioning team throughout the User Application.
- 3 To add a new language to the list, click *Add* and select the desired language.
By default, a newly created provisioning team supports only English.
- 4 Click *Next*.
- 5 Specify the managers for the team, as described in “[Specifying the Team Managers](#)” on [page 354](#).
- 6 Specify the members of the team, as described in “[Specifying the Team Members](#)” on [page 355](#).
- 7 Specify the team options for the team, as described in “[Specifying the Team Options](#)” on [page 356](#).
- 8 Review your settings, then click *Finish*.
iManager displays a message to remind you that you must define a team requests object for this team in order to make the team available for use within the User Application.
- 9 Click *OK*.

Specifying the Team Managers

This section provides instructions for specifying the managers for a team.

To specify the team managers:

- 1 Click *Add*.



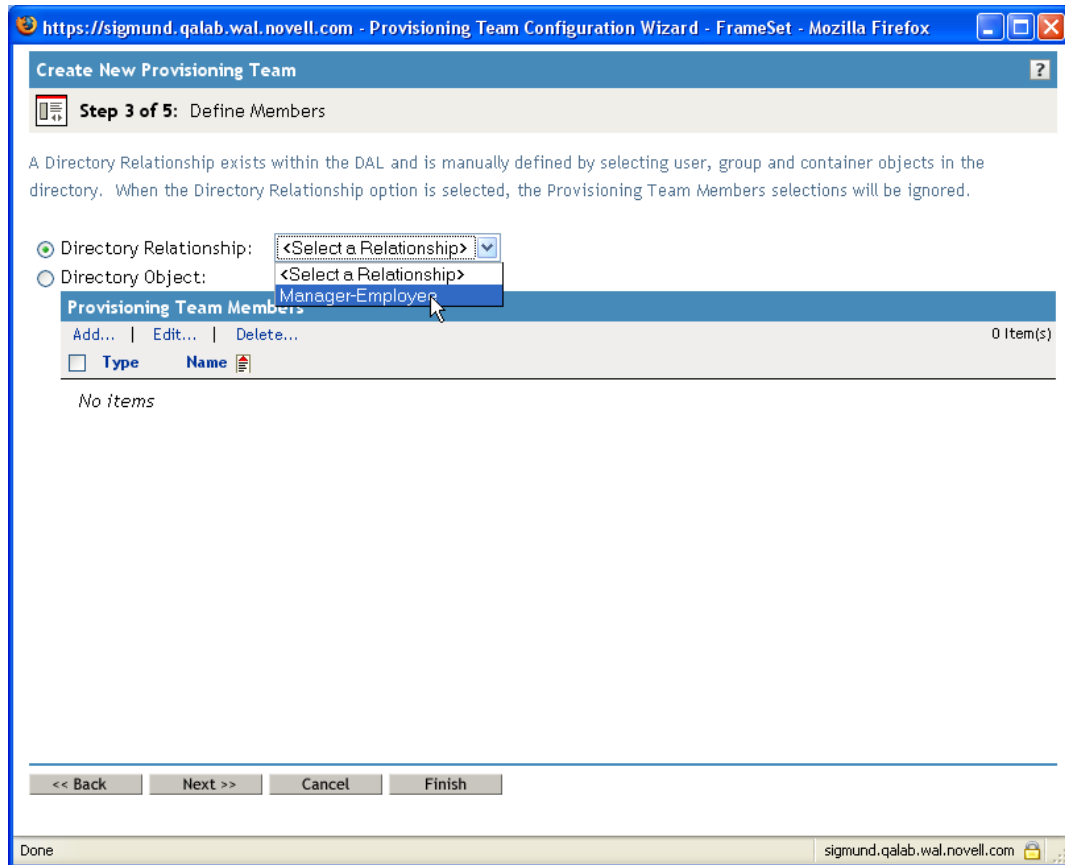
The interface displays the Object Selector.

- 2 Select one or more users or groups, then click *OK*.
- 3 Click *Next*.

Specifying the Team Members

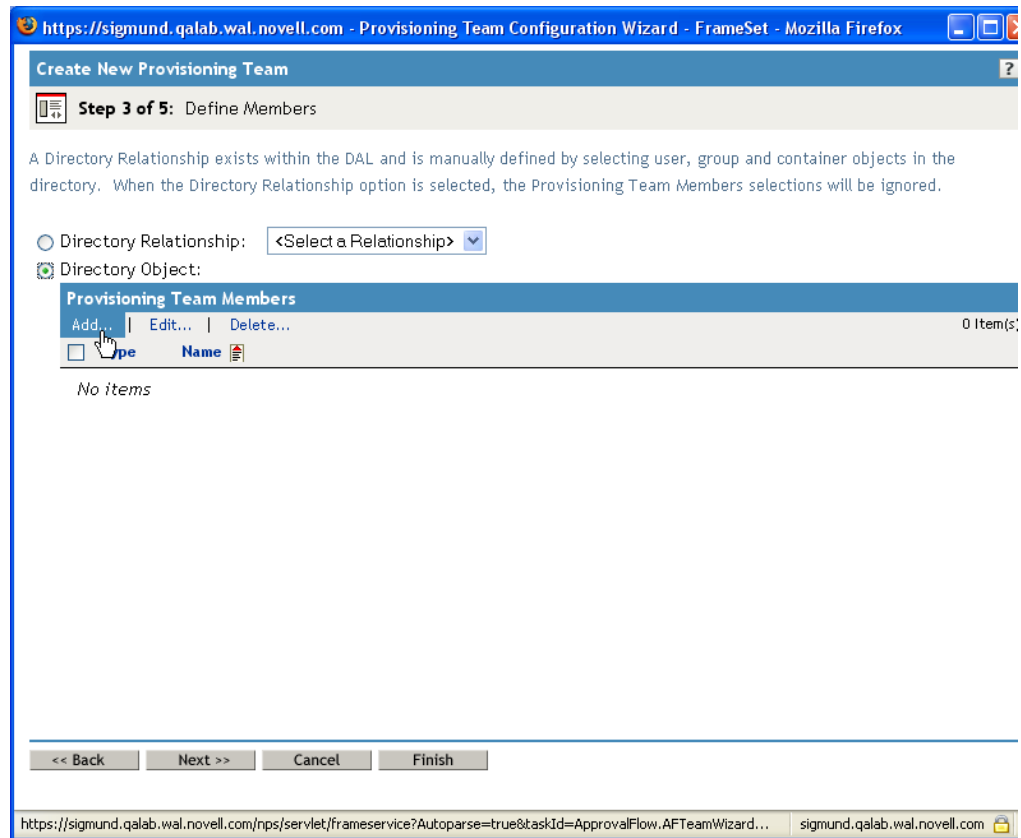
To specify the team members:

- 1 To define the members by using a directory relationship, click *Directory Relationship*, then select the relationship in the drop-down list.



- 2 To define the members by selecting them individually, click *Directory Object*, then follow these instructions:

2a Click *Add*.



The interface displays the Object Selector.

2b Select one or more users, groups, or containers, then click *OK*.

3 Click *Next*.

Specifying the Team Options

To specify the team options:

- 1** Define which request types the team manager can act on by selecting one of these options in the *Provisioning Request Scope* drop-down list:
 - ♦ *Individual Provisioning Request* indicates that this team definition applies to a single request type. You specify the request type when you define the team requests object.
 - ♦ *Provisioned Request Category* indicates that this team definition applies to all request types associated with a particular category. You specify the category when you define the team requests object.
 - ♦ *All Provisioning Requests* indicates that this team definition applies to all request types.
- 2** Define the team settings, as follows:

Setting	Description
<i>Allow managers to set team availability for team members</i>	When this setting is enabled, the team managers can access the <i>Team Availability</i> action in the navigation menu of the User Application.
<i>Allow managers to set proxies for team members</i>	When this setting is enabled, the team managers can access the <i>Team Proxy Assignments</i> action in the navigation menu of the User Application.
<i>All team members will display in a select list</i>	When this option is selected, the manager can select team members in a drop-down list box. Use this option when the team has only a few members.
<i>The manager will need to search for the user using a select-pick list</i>	When this option is selected, the manager must use the Object Selector to select team members. Use this option when the team has a large number of members.

If a particular team definition does not permit team managers to set proxies or team availability settings, the manager can still view the settings defined for the team members by the administrator or by a manager of another team to which these users belong. However, the team manager cannot edit these settings, view details for these settings, or create new proxy assignments or team availability settings.

https://sigmund.qalab.wal.novell.com - Provisioning Team Configuration Wizard - FrameSet - Mozilla Firefox

Create New Provisioning Team

Step 4 of 5: Define Team Options

Define which Provisioning Requests can be acted upon by this team's managers and the options for how they may control team member's interactions with those Provisioning Requests.

Provisioning Request Scope:

☒ Allow managers to set team availability for team members
☒ Allow managers to set proxies for team members
☒ All team members will display in a select list
☐ The manager will need to search for the user using a select-pick list

Done sigmund.qalab.wal.novell.com

- 3 If there are any existing team requests objects that refer to this team definition, you can navigate directly to one of these objects by clicking on the object name in the list, under the heading *Provisioning Team Requests Referring to this Provisioning Team*.

Provisioning Team Requests Referring to this Provisioning Team

[MyTeamRequests.TeamDefs.AppConfig.UserApplication.TestDrivers.novell](#)

When you click on a team requests object, iManager asks you commit your *Provisioning Request Scope* setting. If you click *OK* to commit this setting, the user interface takes you directly to the Provisioning Team Requests plug-in to allow you to make changes to the team requests object.

19.2.3 Deleting a Provisioning Team

To delete a provisioning team:

- 1 Select the provisioning team you want to delete by clicking the check box next to the name.
- 2 Click the *Delete* command in the Provisioning Teams panel.

19.3 Managing Provisioning Team Request Rights

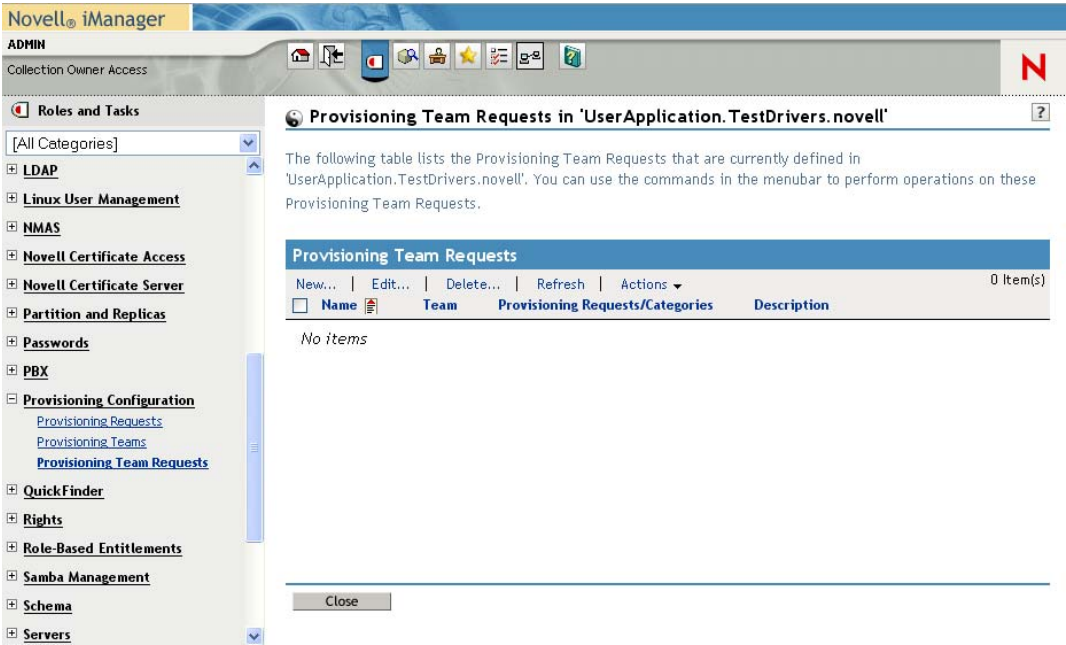
Before configuring a provisioning team requests object, you need to select the Identity Manager User Application driver that contains the definition. After selecting the driver, you can create a new team requests definition, edit an existing definition, or delete an existing definition.

19.3.1 Selecting the Driver

To select an Identity Manager User Application driver:

- 1 Select the *Identity Manager* category in iManager.
- 2 Open the *Provisioning Configuration* role.
- 3 Click the *Provisioning Team Requests* task.
iManager displays the User Application Driver panel.
- 4 Specify the driver name in the *User Application Driver* field, then click *OK*.

iManager displays the Provisioning Team Requests panel. The Provisioning Team Requests panel displays a list of existing team requests objects.

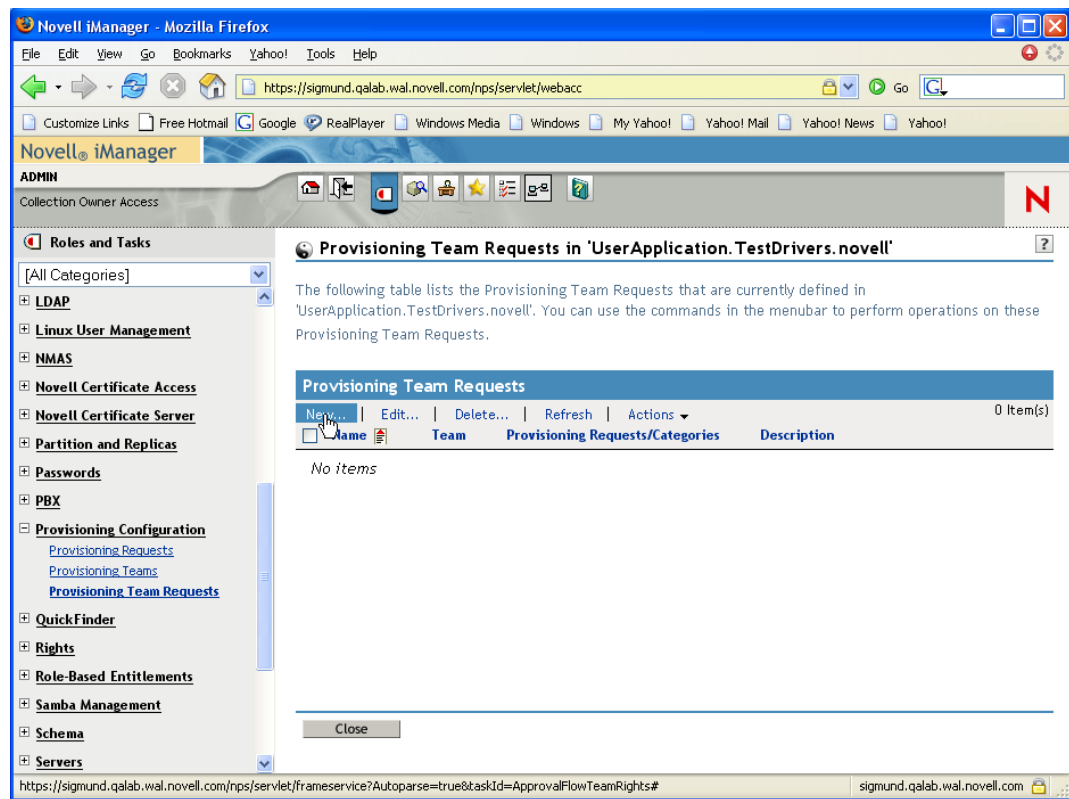


Changing the driver. When you have selected a driver, the driver selection remains in effect for the duration of your iManager session, unless you select a new driver. To select a new driver, click the *Actions* command, then choose *Select User Application Driver* from the *Actions* menu.

19.3.2 Creating or Editing a Provisioning Team Requests Object

To create a new provisioning team requests object:

- 1 Click the *New* command in the Provisioning Team Requests panel.



The first page of the Create New Provisioning Team Request wizard displays.

https://sigmund.qalab.wal.novell.com - Provisioning Team Request Configuration Wizard - FrameSet - Mozilla F...

Create New Provisioning Team Request

Step 1 of 4: Edit general Provisioning Team Request information.

Enter the name and descriptions for the new Provisioning Team Request.

Name (CN):

Provisioning Team Request Descriptions

Add... | Edit... | Delete... | Refresh 0 Item(s)

Description
No items

<< Back Next >> Cancel Finish

Done sigmund.qalab.wal.novell.com

- 2 Type a common name for the new object in the *Name (CN)* field.
- 3 For each description you want to add for the team requests object, type the description text in the *Description* fields under *Provisioning Team Request Descriptions*. This text is used to identify the provisioning team requests object in iManager.
- 4 To add a new description for the team requests object, click *Add*, type the description text, then click *OK*.
The text is then added to the *Description* field under *Provisioning Team Request Descriptions*. This text is used to describe the team requests object on the Provisioning Team Requests panel.
- 5 Click *Next*.
- 6 Select the team definition to which this team requests object applies, as described in “[Selecting the Team Definition for the Team Requests Object](#)” on page 362.
- 7 Specify the task scope and permission options for the team requests object, as described in “[Specifying the Team Requests Options](#)” on page 363.

8 Review your settings, then click *Finish*.

Create New Provisioning Team Request

Step 4 of 4: Summary

The following information was collected and will be used to create a new Provisioning Team Request:

Provisioning Team Request to Create:

Name: **MyTeamRequests**
Context: **TeamDefs.AppConfig.UserApplication.TestDrivers.novell**
Description: **MyTeam.TeamDefs.AppConfig.UserApplication.TestDrivers.novell**
Provisioning Team: **MyTeam.TeamDefs.AppConfig.UserApplication.TestDrivers.novell**
Provisioning Team's Request scope: **Provisioned Resource Category**
Provisioning Categories: **1**

Entitlements

Task Scope Options:

Task scope is addressee: **true**
Task scope is recipient: **false**

Permission Options:

Allow managers to initiate: **true**
Allow managers to retract: **true**
Allow managers to set delegate: **true**
Allow managers to claim tasks: **true**
Allow managers to reassign tasks: **true**

Select "Finish" to create this Provisioning Team Request.

<< Back Next >> Cancel Finish

javascript:handlePB('AFPB_Next') sigmund.qalab.wal.novell.com

Selecting the Team Definition for the Team Requests Object

To select the team definition:

- 1 Use the Object Selector to pick a team.

After you have made your selection, the team is displayed in the *Provisioning Team* field, and the team options settings for the team are displayed under *Provisioning Team Options*.

Create New Provisioning Team Request

Step 2 of 4: Edit selected Provisioning Team

A Provisioning Team Request serves as the "glue" that binds a Provisioning Team with a collection of Provisioning Requests. You must select a Provisioning Team and then, based on its definition, you may need provide a specific Provisioning Request or a list of Provisioning Categories that the Provisioning Team may act upon.

Provisioning Team:

Provisioning Team Options

- Managers are allowed to set team availability for team members
- Managers are allowed to set proxies for requests that fall under the domain of the team
- All team members will display in a select list
- Applies to all Provisioning Requests of the selected Provisioning Categories

<< Back Next >> Cancel Finish

https://sigmund.qalab.wal.novell.com/nps/servlet/frameservice?Autoparse=true&taskId=ApprovalFlow.AFTeamRights... sigmund.qalab.wal.novell.com

2 Click *Next*.

Specifying the Team Requests Options

To specify the team requests options:

- 1 Define the scope for the team requests object:
 - ♦ If the scope for the team is *Provisioned Resource Categories*, select one or more categories for this team requests object by moving them from the *Available Categories* list into the *Selected Categories* list.

Selected Categories:

- entitlements

Available Categories:

- Accounts
- Groups

- ♦ If the scope for the team is *Individual Provisioning Request*, use the Object Selector to choose the provisioning request for this team requests object.

Provisioning Request:

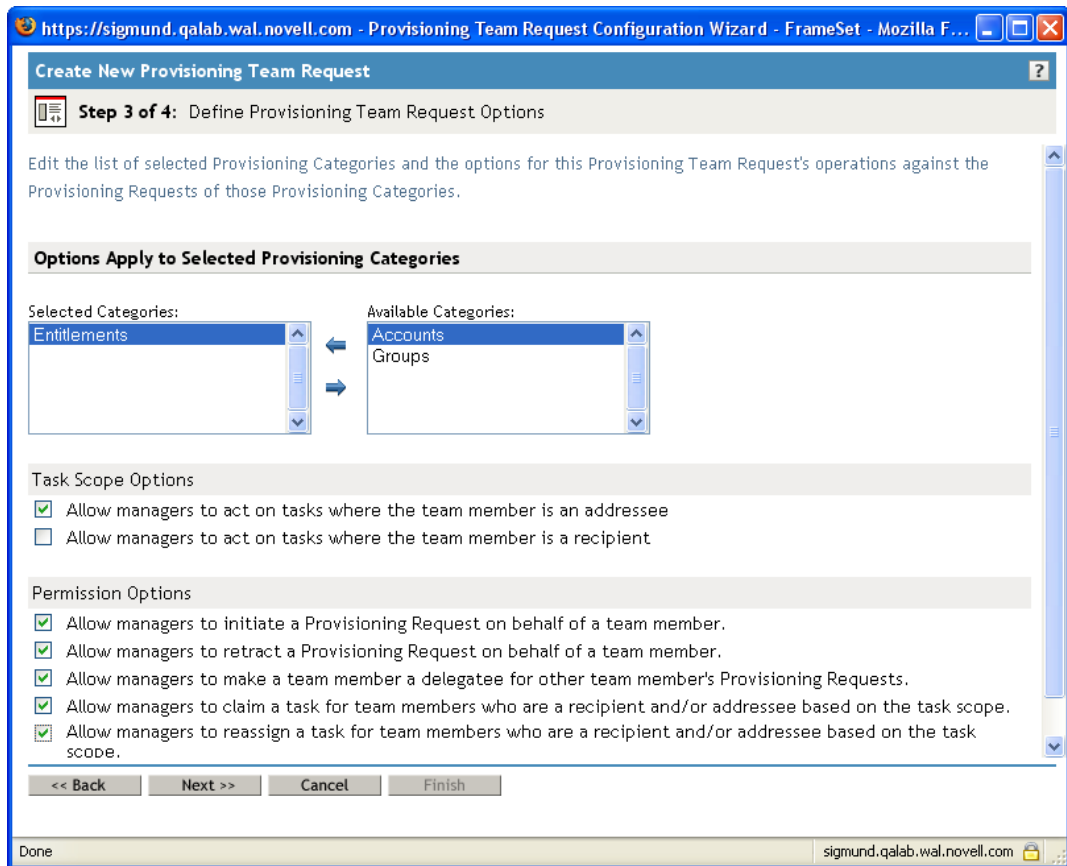
- ♦ If the scope for the team is All Provisioning Requests, you do not need to take any additional action in the team requests object.

2 Define the task scope options, as follows:

Setting	Description
<i>Allow managers to act on tasks where the team member is an addressee</i>	<p>When this setting is enabled, the team managers can use the <i>Team Tasks</i> action within the User Application to take actions on tasks for which the team members are addressees. These actions include approving and denying requests.</p> <p>If you do not permit team managers to act on tasks for which the team member is an addressee, you can view these tasks, but you cannot see details about them, or take actions on them.</p>
<i>Allow managers to act on tasks where the team member is a recipient</i>	<p>When this setting is enabled, the team managers can use the <i>Team Tasks</i> action within the User Application to take actions on tasks for which the team members are recipients. These actions include approving and denying requests.</p> <p>If you do not permit team managers to act on tasks for which the team member is a recipient, you can view these tasks, but you cannot see details about them, or take actions on them.</p> <p>NOTE: For security reasons, the recipient task scope option is disabled by default. Giving a team manager the ability to act on tasks where the recipient of the request is a team member can raise several security issues. First, the manager is then able to view data included on any of the forms that are displayed during the course of workflow execution, regardless of his or her trustee rights. Second, depending on the permission options (see below), a team manager could circumvent the approval process by claiming or approving the task or reassigning it to someone else.</p>

3 Define the permission options, as follows:

Setting	Description
<i>Allow managers to initiate a Provisioning Request on behalf of a team member</i>	When this setting is enabled, the list of resources on the <i>Request Team Resources</i> page of the User Application includes resources that are within the scope of this team. When this setting is disabled, these resources are not included.
<i>Allow managers to retract a Provisioning Request on behalf of a team member</i>	When this setting is enabled, the Retract button is displayed on the <i>Team Requests</i> page for requests that are within the scope of this team. When this setting is disabled, the Retract button is not displayed.
<i>Allow managers to make a team member a delegatee for other team member's Provisioning Requests</i>	<p>When this option is enabled, the manager can use the <i>Team Delegate Assignments</i> action to designate a team member as a delegate for another team member's provisioning requests.</p> <p>If this option is disabled, the manager can still view delegate settings defined for the team members by the administrator or by a manager of another team to which these users belong. However, the team manager cannot edit or delete these settings, view details for these settings, or create new delegate assignments.</p>
<i>Allow managers to claim a task for team members who are a recipient and/or addressee based on the task scope</i>	When this setting is enabled, the Claim button is enabled on the <i>Team Tasks</i> page for requests that are within the scope of this team. When this setting is disabled, the Claim button is greyed out.
<i>Allow managers to reassign a task for team members who are a recipient and/or addressee based on the task scope</i>	When this setting is enabled, the Reassign button is enabled on the <i>Team Tasks</i> page for requests that are within the scope of this team. When this setting is disabled, the Reassign button is greyed out.



4 Click *Next*.

NOTE: The Provisioning Team Requests plug-in allows you to configure two different team requests objects that use the same provisioning request or category with different sets of permissions for the same team. This might lead to conflicts that make the permissions associated with a team unclear. To avoid these sorts of conflicts, make sure you do not define two different team requests objects that specify different sets of permissions for the same provisioning request or category.

19.3.3 Deleting a Provisioning Team Requests Object

To delete a provisioning team requests object:

- 1 Select the provisioning team requests object you want to delete by clicking the check box next to the name.
- 2 Click the *Delete* command in the Provisioning Team Requests panel.

19.4 Creating a Team to Manage Direct Reports

To define a team that manages direct reports:

- 1 In iManager, create a dynamic group called Managers.

1a Set the *Search Scope* to *Search Sub Containers*.

Search Scope:

1b Specify the *Search Filter* as `(&(isManager=TRUE))`.

Search Filter:

For complete details on creating dynamic groups, see the *Novell Identity Manager: Administration Guide*.

2 In iManager, define a provisioning team by selecting *Provisioning Teams* under *Provisioning Configuration*.

2a Name the team DirectReports.

https://sigmund.qalab.wal.novell.com - Provisioning Team Configuration Wizard - FrameSet - Mozilla Firefox

Edit Existing Provisioning Team

Step 1 of 5: Edit general Provisioning Team information.

Enter or edit the display names and descriptions for the defined languages. English will be displayed for undefined languages.

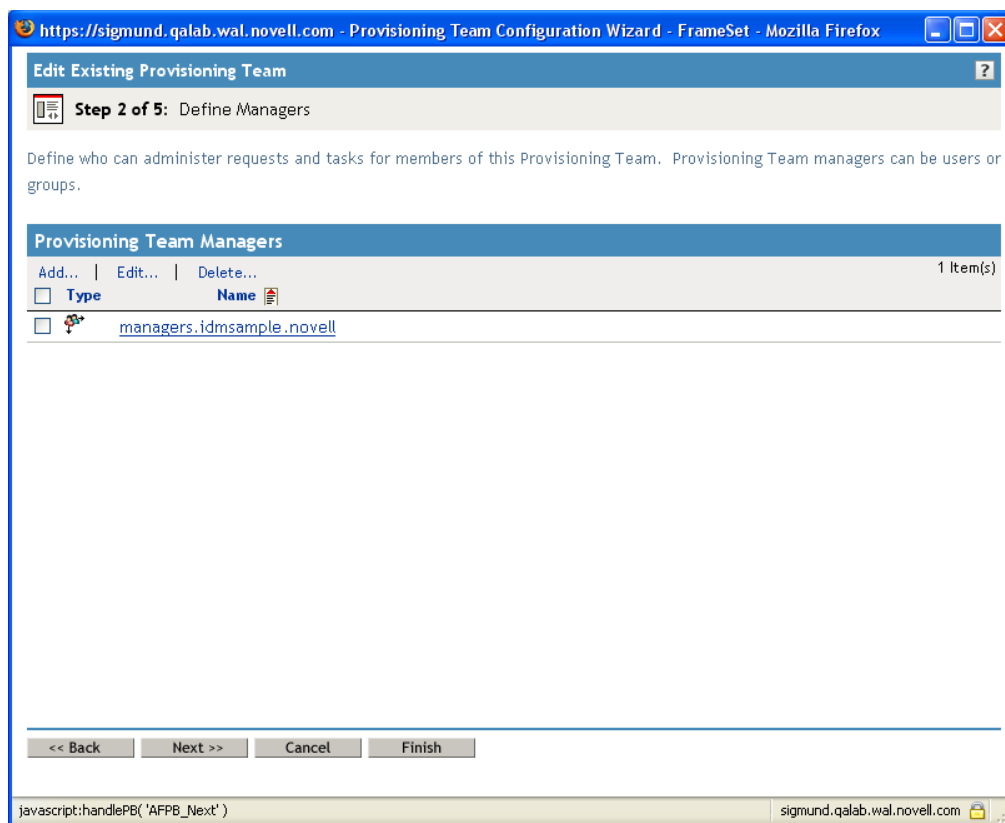
Name (CN): DirectReports

Provisioning Team Localized Strings		
Language	Display name	Description
<input type="checkbox"/> English	<input type="text" value="DirectReports"/>	Team that defines seafang administration manager (based on direct reports)

<< Back Next >> Cancel Finish

Done sigmund.qalab.wal.novell.com

2b To identify the team managers, pick the Managers dynamic group you created earlier.



2c To identify the team members, select the *Manager-Employee* relationship.

https://sigmund.qalab.wal.novell.com - Provisioning Team Configuration Wizard - FrameSet - Mozilla Firefox

Edit Existing Provisioning Team

Step 3 of 5: Define Members

A Directory Relationship exists within the DAL and is manually defined by selecting user, group and container objects in the directory. When the Directory Relationship option is selected, the Provisioning Team Members selections will be ignored.

☒ Directory Relationship: Manager-Employee

☐ Directory Object:

Provisioning Team Members

Add... | Edit... | Delete...

0 Item(s)

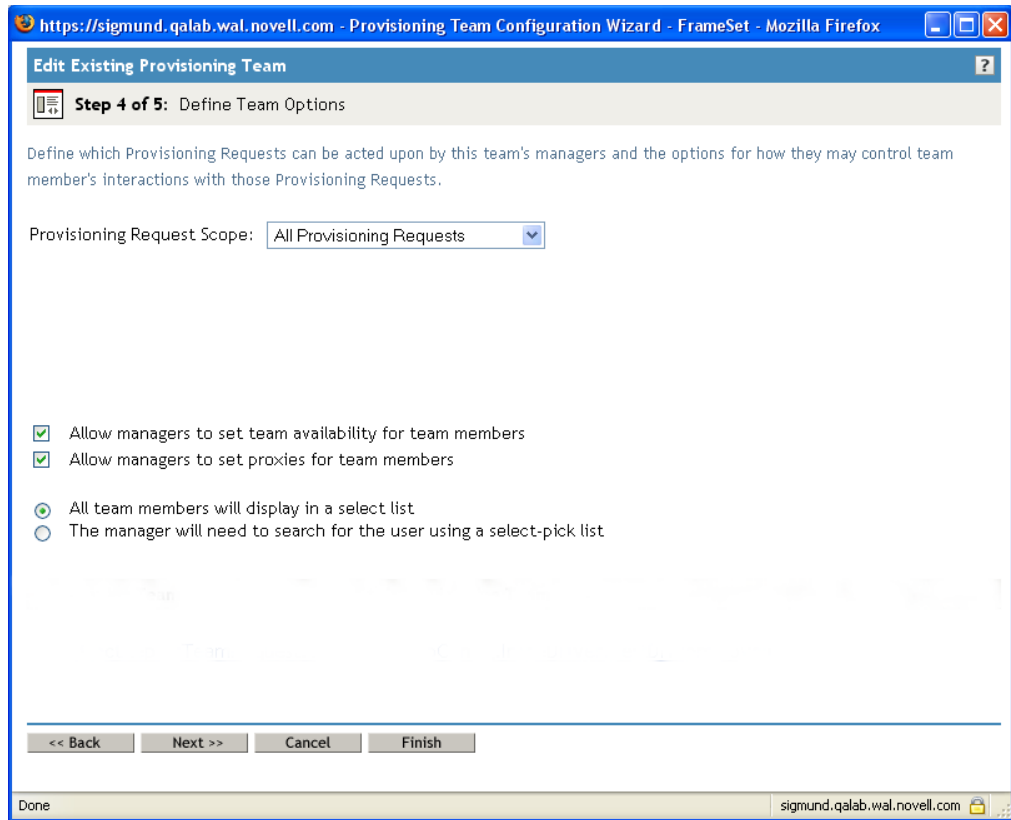
Type	Name
No items	

<< Back Next >> Cancel Finish

Done sigmund.qalab.wal.novell.com

2d To define the team options:

- ♦ Set the *Provisioning Request Scope* to *All Provisioning Requests*.
- ♦ Select *Allow managers to set team availability for team members*.
- ♦ Select *Allow managers to set proxies for team members*.
- ♦ Select *All team members will display in a select list*.



2e Review the summary page, then click *Finish*.

3 In iManager, define a provisioning team requests object by selecting *Provisioning Team Requests* under *Provisioning Configuration*.

3a Name the team *DirectReportsTeamRequestRights*.

3b To identify the associated team, select the *DirectReports* provisioning team you created earlier. When you select this team, the iManager interface shows you the settings for the team.

3c To specify the task scope options:

- ♦ Select *Allow managers to act on tasks where the team member is an addressee*.
- ♦ Deselect *Allow managers to act on tasks where the team member is a recipient*.

3d To define the permission options:

- ♦ Select *Allow managers to initiate a Provisioning Request on behalf of a team member*.
- ♦ Select *Allow managers to retract a Provisioning Request on behalf of a team member*.
- ♦ Select *Allow managers to make a team member a delegatee for other team member's Provisioning Requests*.
- ♦ Select *Allow managers to claim a task for team members who are a recipient and/or addressee based on the task scope*.
- ♦ Select *Allow managers to reassign a task for team members who are a recipient and/or addressee based on the task scope*.

3e Review the summary page, then click *Finish*.

Appendixes

VI

The following sections provide additional reference information and advanced topics for the Identity Manager user application.

- ♦ [Appendix A, “Schema Extensions,” on page 375](#)
- ♦ [Appendix B, “Metrics Web Service,” on page 381](#)
- ♦ [Appendix C, “Provisioning Web Service,” on page 399](#)

Schema Extensions

A

This section describes the schema extensions used by the User Application. It includes these sections:

- ♦ [Section A.1, “Attribute schema extensions,” on page 375.](#)
- ♦ [Section A.2, “Objectclass schema extensions,” on page 377.](#)

A.1 Attribute schema extensions

Attribute Name	Description
srvprvAllowMgrInitiate	A flag that indicates if the manager is allowed to initiate a provisioning request.
srvprvAllowMgrRetract	A flag to indicate if the manager is allowed to retract a provisioning request.
srvprvAllowMgrSetAvailability	A flag that indicates whether the manager can set a proxy for the team.
srvprvAllowMgrSetDelegate	A flag to indicate if the manager is allowed to set delegates for a provisioning request.
srvprvAllowMgrSetProxy	A flag to indicate if the manager is allowed to set a team proxy.
srvprvAllowMgrTaskClaim	A flag to indicate if the manager is allowed to claim a provisioning approval task.
srvprvAllowMgrTaskReassign	A flag to indicate if the manager is allowed to reassign a provisioning approval task.
srvprvAllRequests	A flag to indicate if the assignment covers all provisioning request definitions for a team.
srvprvAOLIMAddress	AOL IM address.
srvprvAssetRef	Representation of the aggregate asset properties for a named asset associated to a user via the <code>srvprvAssetRecipientAux</code> 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.
srvprvAssignStartTime	Time at which a delegation assignment takes effect.
srvprvAssignToRelationship	A target relationship of a delegate assignment.
srvprvAssignToUser	The User targets of a proxy or delegate assignment.
srvprvAutoDisplayTeam	Automatically display team members.

Attribute Name	Description
srvprvCapabilities1-5	Listing of skills for a user.
srvprvCategoryKey	Associates a given Provisioning Request Definition to a set of provisioning categories. Values are keys to a srvprvChoice instance.
srvprvCurrentDelegates	The delegations associated with a user.
srvprvCurrentDelegators	The delegations associated with a user.
srvprvDefaultTheme	The default theme.
srvprvDelegateeDef	The delegates definition DN.
srvprvDelegationDef	The delegation definition DN.
srvprvDelegators	The users who are defined as delegators by this assignment.
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.
srvprvHideAttributes	Flag indicating if certain attributes should be hidden and not displayed.
srvprvHideUser	Flag indicating if the user should be hidden when search list queries are executed.
srvprvIMAddress	Instant Messenger address.
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.
srvprvManager	Indicates users who are managers.
srvprvManagerGroup	Indicates a group containing managers.
srvprvManagerNotMember	Indicates that the manager is not a member of the team.
srvprvMember	Indicates users who are team members.
srvprvMemberContainer	The name of the container containing team members.
srvprvMemberGroup	The name of the group containing team members.
srvprvMemberRelationship	The name of the directory abstraction layer relationship that determines members based attribute in manager object.
srvprvModified	Flag to indicate changes to definitions object instances in the directory model container.

Attribute Name	Description
srvprvNotificationPrefs	Defines the set of notification types users want to receive.
srvprvPreferredLocale	Users preferred locale.
srvprvProcessXML	XML document representing a Provisioning process definition including Workflow and Provisioning Action.
srvprvQueryList	List of saved query/search criteria.
srvprvRelationship	Defines relationships between objects in the identity vault.
srvprvRequest	Exposes one item to be granted or revoked, including the workflow process which defines the run-time aspects of the Workflow and Provisioning Target.
srvprvRequestDefName	The provisioning request definition name associated with a delegate definition.
srvprvRequestScope	The scope of provisioning requests.
srvprvRequestXML	XML document representing the initial request form and its data bindings.
srvprvRevoke	If true, this flag specifies that the Provisioning Request Definition supports a Revoke operation.
srvprvStatus	Specifies the status of the Provisioning Object Supported values.
srvprvTaskGroups	Groups for which the user is a task manager.
srvprvTaskManager	Task manager of the task group.
srvprvTaskScopeAddressee	The addressee's task scope.
srvprvTaskScopeRecipient	The recipient's task scope.
srvprvTeam	The container for team definitions.
srvprvUser	The users associated with a delegation assignment.
srvprvUUID	Unique identifier for portlet.
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 initialize the Provisioning run-time environment, such as themes for the Identity Portal.
srvprvAssetRecipientAux	Records the provisioning of non-IT assets on a user.

Objectclass Name	Description
srvprvChoice	Enumeration of values that can be assigned to a particular attribute, used in a query, 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	Delegates assignment definition.
srvprvDelegateeDefs	Container for delegates definitions.
srvprvDelegationAssignment	Delegation or availability assignment definition.
srvprvDelegationDefs	Container for delegation and delegators definitions.
srvprvDelegatorAssignment	Delegation or availability assignment definition.
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 configuration 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 definitions, to be exposed by the Identity Portlets and Web Applications.
srvprvProxyAssignment	Proxy assignment definition.
srvprvProxyDefs	Container for proxy definitions.
srvprvQuery	Directory abstraction layer query definition.
srvprvQueryDefs	Container for directory abstraction layer query definition.
srvprvRelationship	Defines relationships between objects in the directory, for use in the Identity Portlets and other Web Application components.
srvprvRelationshipDefs	Container for Directory Abstraction Layer Relationship definitions, to be exposed by the Identity Portlets and Web Applications.
srvprvRequest	Exposes one 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 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.

Objectclass Name	Description
srvprvServiceDefs	Container for Service Definition objects, which wrap Web Services called by Workflows.
srvprvTaskGroupAux	Service provisioning task group.
srvprvTeam	Team for provisioning request management.
srvprvTeamDefs	Container for team definitions.
srvprvTeamRequest	Team provisioning requests.
srvprvTheme	Theme Object.
srvprvUserAux	Service provisioning user entity.
srvprvWebAppConfig	Web Application configuration 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.

Metrics Web Service

B

This section describes the Metrics Web Service, which provides metrics for provisioning workflows. Topics include:

- ♦ [Section B.1, “About the Metrics Web Service,” on page 381](#)
- ♦ [Section B.2, “Metrics Web Service API,” on page 390](#)
- ♦ [Section B.3, “Metrics Web Service Examples,” on page 395](#)

B.1 About the Metrics Web Service

The workflow engine includes a Web Service for gathering workflow metrics. The addition of the Metrics Web Service to the workflow engine lets you monitor an approval flow process. In addition, it provides indicators the business manager can use to modify the process for optimal performance.

The metrics are based on traditional business process flow management principles, which emphasize the need for metrics to be actionable. This ensures that the metrics provided match what an operations manager usually looks for when analyzing and optimizing business flows. Therefore, the metrics identify bottlenecks and provide other capacity indicators. The Metrics Web Service allows you to narrow down the metrics to a common and established set of data, instead of trying to anticipate the myriad of metrics and reports that can be created for a business process flow.

When working with the Metrics Web Service, you should keep in mind that the service is not intended to be an all-purpose metrics system:

- ♦ The Metrics Web Service is not a reporting tool or reporting engine. Consequently it does not use a complex query language.
- ♦ The Metrics Web Service is not designed as an all-purpose performance management system. This helps to limit the impact of the needed queries against the live system being monitored.

Operations management stresses three key internal process performance measures that together capture the essence of process flow. These three measures can serve as leading indicators of customer satisfaction: flow time, flow rate, and inventory.

With these measures, an operations manager can answer the following questions:

- ♦ On average, how much time does a provisioning request spend within the process boundaries? (Flow time)
- ♦ On average, how many provisioning requests pass through the process per unit of time? (Flow rate)
- ♦ On average, how many provisioning requests are within the process boundaries at any point in time? (Inventory)

These three measures are related by Little's law:

$$\text{Inventory} = \text{Flow Rate} \times \text{Flow Time}$$

B.1.1 Web Service Semantics

The following semantics apply to the use of the Metrics Web Service:

- ♦ Activities in the Metrics Web Service refer only to user-facing activities (Approval Activities). Negligible running time and the impossibility of controlling the other activities make collecting metrics for these inappropriate.
- ♦ The Metrics Web Service distinguishes between Working Days and Calendar Days. Calendar Days refer to all days between two dates. Working Days refer only to working days between two dates. Since working days may be specified differently in different environments, all Working Days methods return a raw data set that can be used to compute what is appropriate. If no such detail is required, the Calendar Days method will readily return the appropriate metric.

B.1.2 Web Service Endpoint

The Metrics Web Service endpoint can be accessed at the following URL:

`http://server:port/warcontext/metrics/service`

B.1.3 Web Service Methods Grouped by Security Permissions

The service is secured using Basic Authentication. Therefore, you should use SSL to connect to the service. The service uses the same security layer as the User Application and consequently not all service operations are allowed to all users. Only a Provisioning Administrator will have unconditional access to all the methods. On the other hand team managers will only have access to metrics that pertain to their team and team members.

Hence the Metrics Web Service operations are divided into 3 categories according to role and security permissions:

- ♦ Team manager operations
- ♦ Provisioning Application Administrator operations
- ♦ Utility operations

Team Metrics

Team managers can only retrieve metrics on a team for which they are managers. These are the methods available to team managers:

Table B-1 *Team Metrics Methods*

Method	Description
<code>getClaimedFlowTimeCalendarDays</code>	Returns the average time in hours the provisioning request was claimed for within the specified time interval
<code>getClaimedFlowTimeWorkingDays</code>	Returns the result set required to compute the average time the provisioning request was claimed for the specified time interval

Method	Description
getToClaimedFlowTimeCalendarDays	Returns the average time in hours it took the provisioning request to be claimed from the moment it was available to addressees
getToClaimedFlowTimeWorkingDays	Returns the average time it took the provisioning request to be claimed from the moment it was available to addressees, within the specified time interval
getClaimedInventory	Returns the average number of provisioning requests claimed within the specified interval
getClaimedThroughputWorkingDays	Returns the average number of provisioning requests claimed within the specified interval
getTeamLongestRunning	Returns a result set of the longest running request in seconds for which members of the team acted as addressees
getTeamFlowHistory	Returns a result set of the activity outcomes, addressee and addressee messages for the specified list of provisioning requests
getTeamHistoryForInitiators	Returns a result set of the provisioning request and their status for which members of the team acted as initiators
getTeamHistoryForRecipients	Returns a result set of the provisioning request and their status for which members of the team acted as recipients
getTeamRunningTime	Returns the average time in seconds the specified provisioning requests have been running
getTeamDecisionCount	Returns the number of decisions the team made as addressees for the specified provisioning request
getTeamInitiatedCount	Returns the number of provisioning requests initiated by the team
getTeamRecipientCount	Returns the provisioning requests for which a member of the team acts as a recipient

Provisioning Administrator Metrics

This role is unrestricted and may perform any of the service's operations. These are the methods that are only available to Provisioning Administrators.

Table B-2 *Provisioning Administrator Metrics Methods*

Method	Description
getActivityFlowTimeCalendarDays	Returns the average time in hours the user activity took to complete
getActivityFlowTimeWorkingDays	Returns the result set required to compute the average time the user activity took to complete

Method	Description
getActivityInventory	Returns the average number of provisioning requests at any one time for the specified user activity
getActivityThroughputCalendarDays	Returns the average number of provisioning requests per hours that exited the specified user activity within the specified time interval
getActivityThroughputWorkingDays	Returns the result set required to compute average time it takes a provisioning request to complete for the specified time interval
getFlowTimeCalendarDays	Returns average time in hours it takes a provisioning request to complete for the specified time interval
getFlowTimeWorkingDays	Returns the result set required to compute average time it takes a provisioning request to complete for the specified time interval
getInventory	Returns the average number of provisioning requests in the system at any one time for the specified time interval
getLongestClaimed	Returns a result set of the provisioning requests that have been claimed but not acted upon (time in seconds)
getLongestRunning	Returns a result set of the longest running provisioning requests (time in seconds)
getFlowCount	Returns the number of provisioning requests
getFlowHistory	Returns a result set of the activity outcomes, addressee and addressee messages for the specified list of provisioning requests
getFlowHistoryForInitiators	Returns the list of provisioning requests and their status for the specified initiators
getFlowHistoryForRecipients	Returns the list of provisioning requests and their status for the specified recipients
getRunningTime	Returns the average running time in seconds for the provisioning requests that are currently running
getThroughputCalendarDays	Returns the average number of provisioning requests per hour that completed within the specified interval
getThroughputWorkingDays	Returns the result set required to compute the average number per hour of provisioning requests that completed within the specified interval

Utility Operations

Both team managers and administrators may perform these operations:

Table B-3 *Utility Operations*

Method	Description
getVersion	Returns the server version of the Web service. This should always be used to ensure version matching between client and server code.
getAllProvisioningFlows	Returns the list of provisioning flows that the logged in user can see
getUserActivityOnlyFlow	Returns a GraphViz DOT (http://www.graphviz.org/) representation of the provisioning workflow
getTeams	Returns the list of teams the logged in user manages
getTeamMembers	Returns the list of team members for the specified team

B.1.4 Specifying Filters

As mentioned above, the Metrics Webservice does not use a complex query language. Instead filters can be used to narrow results by criteria such as date ranges or approval statuses.

These are the filters you can specify (see type `FilterConstants` in service's WSDL):

Table B-4 *Filters for Narrowing Metric Results*

Filter	Description
KEY_ACTIVITY_ID	A User Activity Id as defined in the provisioning request definition
KEY_APPROVAL_STATUS	<p>The approval status for the provisioning request. Possible values are:</p> <ul style="list-style-type: none"> ◆ <code>ApprovalStatusProcessing</code> ◆ <code>ApprovalStatusDenied</code> ◆ <code>ApprovalStatusRefused</code> ◆ <code>ApprovalStatusApproved</code> ◆ <code>ApprovalStatusRetract</code> ◆ <code>ApprovalStatusError</code>
KEY_ENTITLEMENT_STATE	<p>The state of the entitlement associated with the provisioning request. Possible values are:</p> <ul style="list-style-type: none"> ◆ <code>EntitlementUnknown</code> ◆ <code>EntitlementGranted</code> ◆ <code>EntitlementRevoked</code>

Filter	Description
KEY_ENTITLEMENT_STATUS	The status of the entitlement associated with the provisioning request. Possible values are: <ul style="list-style-type: none"> ♦ EntitlementSuccess ♦ EntitlementWarning ♦ EntitlementError ♦ EntitlementFatal
KEY_INITIATOR	The user DN of the workflow initiator
KEY_L_COMPLETION_TIME	The date indicating the start of the interval for workflow completion
KEY_S_COMPLETION_TIME	The date indicating the end of the interval for workflow completion
KEY_L_ENTITLEMENT_TIME	The date indicating the start of the interval for entitlement time
KEY_S_ENTITLEMENT_TIME	The date indicating the end of the interval for entitlement time
KEY_S_START_TIME	The date indicating the start of the interval for workflow start
KEY_L_START_TIME	The date indicating the end of the interval for workflow start
KEY_PROCESS_ID	The DN of the provisioning request
KEY_PROCESS_STATUS	The status of the provisioning request. Possible values are: <ul style="list-style-type: none"> ♦ ProcessStatusRunning ♦ ProcessStatusStopped ♦ ProcessStatusTerminated ♦ ProcessStatusCompleted
KEY_PROCESS_VERSION	The process version associated with the workflow version
KEY_RECIPIENT	The user DN of the workflow recipient
KEY_REQUEST_ID	The unique id associated with the workflow instance

Here is a Java example. Note that your code will obviously differ depending on the platform you use for your Web Service client:

```

HashMap map=new HashMap();
map.put(MetricsFilter.KEY_PROCESS_STATUS,
MetricsFilter.ProcessStatusRunning);
double flowtime = metrics.getFlowTimeCalendarDays(processId,
processVer, activity, 5, calendar1.getTime(),
calendar2.getTime(), MetricsFilter.ACTIVITY_CLAIMED,
MetricsFilter.ACTIVITY_FORWARDED, map);

```

...

Please consult the WebService WSDL for more information:

`http://server:port/warcontext/metrics/service?WSDL`

B.1.5 Generating the Stub Classes

To generate the stub classes for the Metrics Web Service, you need to run the `wsdl2java` tool. To ensure that the classes are generated correctly, you need to use JDK 1.5 or higher. The following libraries are required:

- ♦ `wssdk.jar`
- ♦ `mail.jar`
- ♦ `tools.jar`
- ♦ `activation.jar`
- ♦ `workflow.jar`

To generate the stub classes, follow these steps:

- 1 Extract the `IRemoteMetrics.wsdl` provided within the `workflow.jar`, or point your browser to the server URL where the metrics wsdl is located and save it to a file with the `.wsdl` extension. Here's the URL for the WSDL:

`http://server:port/warcontext/metrics/service?WSDL`

- 2 Copy the following command into a `.cmd` file and execute it from a command prompt:

```
"C:\Program Files\Java\jdk1.5.0\bin\java" -cp "../Provisioning Web
Service/lib/wssdk.jar;../Provisioning Web Service/lib/jaxrpc-
api.jar";"../Provisioning Web Service/lib/mail.jar";"../
Provisioning Web Service/lib/activation.jar";"c:\Program
Files\Java\jdk1.5.0\lib\tools.jar";
com.novell.soa.ws.impl.tools.wsdl2java.Main -verbose -ds gensrc -d
WEB-INF/classes -noskel -notie -genclient -keep -package
com.novell.soa.af.metrics.soap.impl -javadoc IRemoteMetrics.wsdl
```

After the command has been executed, you should find Java source in the `gensrc` directory and the compiled classes in the `WEB-INF/classes` directory.

- 3 Create a jar file containing the classes from the `WEB-INF/classes` directory so that the client application can resolve them.

B.1.6 Obtaining the Remote Interface

Before you can begin calling methods on the Metrics Web Service, you need to have a reference to the remote interface.

The code below shows how to obtain the remote interface.

```
import java.util.Locale;
import java.util.Properties;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.xml.rpc.Stub;
import com.novell.qa.soap.common.util.LoggerUtils;
```

```

import com.novell.qa.soap.common.util.LoginData;
import com.novell.qa.soap.common.util.ServiceUtils;
import com.novell.soa.af.ClusterException;
import com.novell.soa.af.impl.soap.Provisioning;
import com.novell.soa.af.impl.soap.ProvisioningService;
import com.novell.test.automator.framework.TestProgramException;
import com.rational.test.ft.script.RationalTestScript;
import com.novell.soa.af.metrics.soap.MetricsClientHelper;
import com.novell.soa.af.metrics.soap.MetricsStubWrapper;
import com.novell.soa.af.metrics.soap.impl.MetricsService;
import com.novell.soa.af.metrics.soap.impl.MetricsServiceException;
import com.novell.soa.af.metrics.soap.impl.IRemoteMetrics;

/**
 * Method to obtain the remote interface to the Metrics endpoint
 * @param _url
 * @param _username
 * @param _password
 * @return IRemoteMetrics interface
 * @throws Exception
 */
private IRemoteMetrics getStub(String _url, String _username, String
_password)
throws Exception
{
    Properties properties = new Properties();
    properties.put(Context.INITIAL_CONTEXT_FACTORY,
"org.jnp.interfaces.NamingContextFactory");

    String lookup =
"xmlrpc:soap:com.novell.soa.af.metrics.soap.impl.MetricsService";

    InitialContext ctx = new InitialContext();
    MetricsService svc = (MetricsService) ctx.lookup(lookup);

    Stub stub = (Stub)svc.getIRemoteMetricsPort();

    stub._setProperty(Stub.USERNAME_PROPERTY, _username);
    stub._setProperty(Stub.PASSWORD_PROPERTY, _password);
    stub._setProperty(Stub.SESSION_MAINTAIN_PROPERTY, Boolean.TRUE);
    stub._setProperty(Stub.ENDPOINT_ADDRESS_PROPERTY, _url);

    return (IRemoteMetrics) stub;
}

```


Here's the code to call the method defined above:

```
IRemoteMetrics stub = null;
    try
    {
        //
        // Get the stub
        String url = m_loginData.getURL();
        stub = getStub(url, _username, _password);
    }
    catch(Exception e)
    {
        String msg = e.getMessage();
        LoggerUtils.logError(msg);
        throw new TestProgramException(msg);
    }
    return stub;
```

In order for this code to work, the URL passed to the getStub() method would need to point to the SOAP endpoint, as shown below:

`http://myserver:8080/IDMProv/metrics/service`

The user name needs to be a fully qualified DN such as the following:

`"cn=admin,ou=idmsample,o=novell"`

B.1.7 Metrics Configuration Settings

The Metrics Web Service impact on the live system is limited by 4 settings that may be modified in the IDMfw.jar/WorkflowService-conf/config.xml file:

Table B-5 Metrics Configuration Settings

Key in config.xml	Description
<key>Metrics/TimeRequiredBetweenClientRequests</key>	Required time between client requests in ms (default is 250 ms)
<key>Metrics/MaxClients</key>	Maximum number of concurrent client sessions (default is 10)
<key>Metrics/MaxRows</key>	Maximum number of rows any query can return
<key>Metrics/MaxTeamMembers</key>	Maximum Number of Team Members
<key>Metrics/SecondsToAnythingDivider</key>	The divider used in all throughput computations (default 3600). Original values are in seconds so all throughputs are consequently per hour.

When the limit has been reached for any of these settings a Web Service fault is generated indicating the problem. In addition, for settings 1 and 2, the fault includes an error code.

- ♦ If the fault is caused by a `TimeRequiredBetweenClientRequests` error, the error code is 100.
- ♦ If the fault is caused by a `MaxClients` errors, the error code is 200.
- ♦ If the fault is caused by a closed client connection error, the error code is 300.

Client consumers of the Metrics Web Service will have to include in their code provisions for retrying a request. Here is a simple Java listing that shows how this can be achieved:

```
try {
    for (int i = 0; i < retries; i++) {
        try {
            return metrics.getFlowCount(strDN, strId, new
                HashMap());
        } catch (MetricsServiceException e) {
            if (e.getErrorCode() == 100 //subsequent call
                error
                || e.getErrorCode() == 200) { //too many
                    clients
                try {
                    Thread.sleep(retryPause);
                } catch (Exception ex) {
                    // to nothing
                }
            } else {
                throw e2;
            }
        } else {
            throw new RuntimeException(e);
        }
    } catch (Exception e) {
        throw e;
    }
}
...
}
```

B.2 Metrics Web Service API

This section provides details about the methods available with the Metrics web service.

All of the methods throw `MetricsServiceException` and `RemoteException`. To improve readability, the throws clause has been omitted from the method signatures.

B.2.1 Team Manager Methods

This section provides reference information for each method available to team managers.

getClaimedFlowTimeCalendarDays

Syntax: Here's the method signature:

```
double getClaimedFlowTimeCalendarDays(String processId, String
processVersion, Date startCompletionTime, Date endCompletionTime,
String teamDN, Map filters)
```

getClaimedFlowTimeWorkingDays

Syntax: Here is the method signature:

```
MetricsResultset getClaimedFlowTimeWorkingDays(String processId,
String processVersion, Date startCompletionTime, Date
endCompletionTime, String teamDN, Map filters)
```

getToClaimedFlowTimeCalendarDays

Syntax: Here is the method signature:

```
double getToClaimFlowTimeCalendarDays(String processId, String
processVersion, Date startCompletionTime, Date endCompletionTime,
String teamDN, Map filters)
```

getToClaimedFlowTimeWorkingDays

Syntax: Here is the method signature:

```
MetricsResultset getToClaimFlowTimeWorkingDays(String processId,
String processVersion, Date startCompletionTime, Date
endCompletionTime, String teamDN, Map filters)
```

getClaimedInventory

Syntax: Here is the method signature:

```
double getClaimedInventory(String processId, String processVersion,
Date startCompletionTime, Date endCompletionTime, String teamDN, Map
filters)
```

getClaimedThroughputCalendarDays

Syntax: Here is the method signature:

```
double getClaimedThroughputCalendarDays(String processId, String
processVersion, Date startCompletionTime, Date endCompletionTime,
String teamDN Map filters)
```

getClaimedThroughputWorkingDays

Syntax: Here is the method signature:

```
MetricsResultset getClaimedThroughputWorkingDays(String processId,
String processVersion, Date startCompletionTime, Date
endCompletionTime, String teamDN, Map filters)
```

getTeamLongestRunning

Syntax: Here is the method signature:

```
MetricsResultset getTeamLongestRunning(String processId, String  
processVersion, String teamDN, Map filters)
```

getTeamLongestClaimed

Syntax: Here is the method signature:

```
MetricsResultset getTeamLongestClaimed(String processId, String  
processVersion, String teamDN, Map filters)
```

getTeamFlowHistory

Syntax: Here is the method signature:

```
MetricsResultset getTeamFlowHistory(List requestIds)
```

getTeamHistoryForInitiators

Syntax: Here is the method signature:

```
MetricsResultset getTeamHistoryForInitiators(String teamDN, Map  
filters)
```

getTeamHistoryForRecipients

Syntax: Here is the method signature:

```
MetricsResultset getTeamHistoryForRecipients(String teamDN, Map  
filters)
```

getTeamRunningTime

Syntax: Here is the method signature:

```
double getTeamRunningTime(String processId, String processVersion,  
String teamDN, Map filters)
```

getTeamDecisionCount

Syntax: Here is the method signature:

```
int getTeamDecisionCount(String processId, String processVersion,  
String teamDN, Map filters)
```

getTeamInitiatedCount

Syntax: Here is the method signature:

```
int getTeamInitiatedCount(String processId, String processVersion,  
String teamDN, Map filters)
```

getTeamRecipientCount

Syntax: Here is the method signature:

```
int getTeamRecipientCount(String processId, String processVersion,  
String teamDN, Map filters)
```

B.2.2 Provisioning Application Administrator Methods

This section provides reference information for each method available to the Provisioning Application Administrator.

getActivityFlowTimeCalendarDays

Syntax: Here is the method signature:

```
double getActivityFlowTimeCalendarDays(String processId, String
processVer, String activityId, Date startTime, Date completeTime, Map
filters)
```

getActivityFlowTimeWorkingDays

Syntax: Here is the method signature:

```
MetricsResultset getActivityFlowTimeWorkingDays(String processId,
String processVer, String activityId, Date startTime, Date
completeTime, Map filters)
```

getActivityInventory

Syntax: Here is the method signature:

```
double getActivityInventory(String processId, String processVersion,
String activityId, Date startTime, Date completeTime, Map filters)
```

getActivityThroughputCalendarDays

Syntax: Here is the method signature:

```
double getActivityThroughputCalendarDays(String processId, String
processVersion, String activityId, Date startTime, Date
completiontime, Map filters)
```

getActivityThroughputWorkingDays

Syntax: Here is the method signature:

```
MetricsResultset getActivityThroughputWorkingDays(String processId,
String processVersion, String activityId, Date startTime, Date
completiontime, Map filters)
```

getInventory

Syntax: Here is the method signature:

```
double getInventory(String processId, String processVersion, Date
startTime, Date completeTime, Map filters)
```

getLongestClaimed

Syntax: Here is the method signature:

```
MetricsResultset getLongestClaimed(String processId, String
processVersion, Map filters)
```

getLongestRunning

Syntax: Here is the method signature:

```
MetricsResultset getLongestRunning(String processId, String  
processVersion, Map filters)
```

getFlowCount

Syntax: Here is the method signature:

```
int getFlowCount(String processId, String processVersion, Map filters)
```

getFlowHistory

Syntax: Here is the method signature:

```
MetricsResultset getFlowHistory(List requestIds)
```

getFlowHistoryForInitiators

Syntax: Here is the method signature:

```
MetricsResultset getFlowHistoryForInitiators(List initiators, Map  
filters)
```

getFlowHistoryForRecipients

Syntax: Here is the method signature:

```
MetricsResultset getFlowHistoryForRecipients(List recipients, Map  
filters)
```

getRunningTime

Syntax: Here is the method signature:

```
double getRunningTime(String processId, String processVersion, Map  
filters)
```

getThroughputCalendarDays

Syntax: Here is the method signature:

```
double getThroughputCalendarDays(String processId, String  
processVersion, Date startTime, Date completiontime, Map filters)
```

getThroughputWorkingDays

Syntax: Here is the method signature:

```
MetricsResultset getActivityThroughputWorkingDays(String processId,  
String processVersion, String activityId, Date startTime, Date  
completiontime, Map filters)
```

B.2.3 Utility Methods

This section provides reference information for each utility method. Both team managers and administrators can call these methods.

getVersion

Syntax: Here is the method signature:

```
VersionVO getVersion()
```

getAllProvisioningFlows

Syntax: Here is the method signature:

```
MetricsResultset getAllProvisioningFlows()
```

getUserActivityOnlyFlow

Syntax: Here is the method signature:

```
BasicModelVO getUserActivityOnlyFlow(String processId, String processVer)
```

getTeams

Syntax: Here is the method signature:

```
MetricsResultset getTeams()
```

getTeamMembers

Syntax: Here is the method signature:

```
MetricsResultset getTeamMembers(String teamDN)
```

B.3 Metrics Web Service Examples

This section provides examples that show how to use the Metrics Web Service to gather workflow metrics. The examples assume that you have obtained a stub, as shown in [Section B.1.6, “Obtaining the Remote Interface,” on page 387](#), and potentially wrapped it in an object that handles the potential error conditions, as described in [Section B.1.7, “Metrics Configuration Settings,” on page 389](#).

B.3.1 General Examples

This example uses the KEY_APPROVAL_STATUS filter to compare the decision outcomes for a provisioning request type. This could be used to generate a pie chart for example.

```
FilterConstants constants=new FilterConstants();
Map<MetricsFilter, Object> map = new HashMap<MetricsFilter, Object>();
map.put(MetricsFilter.KEY_APPROVAL_STATUS, constants.getApprovalStatusApproved());
double
accepted=stubWrapper.getFlowCount(processId, processVersion, map);
map.put(MetricsFilter.KEY_APPROVAL_STATUS, constants.getApprovalStatusDenied());
double denied=stubWrapper.getFlowCount(processId, processVersion, map);
map.put(MetricsFilter.KEY_APPROVAL_STATUS, constants.getApprovalStatusError());
double error=stubWrapper.getFlowCount(processId, processVersion, map);
map.put(MetricsFilter.KEY_APPROVAL_STATUS, constants.getApprovalStatusExtract());
```

```
double
retracted=stubWrapper.getFlowCount(processId,processVersion,map);
map.put(MetricsFilter.KEY_APPROVAL_STATUS,
constants.getApprovalStatusRefused());
double refused = stubWrapper.getFlowCount(processId,
processVersion, map);
```

Additional filters may be specified by adding appropriate entries to the filter map. The following examples illustrate how you might do add various types of filters.

Adding a start date filter

To add a start date filter (01/01/2006 < date < 02/01/2006):

```
Calendar startDate=Calendar.getInstance();
startDate.set(2006,0,1);
Calendar endDate=Calendar.getInstance();
endDate.set(2006,1,1);
map.put(MetricsFilter.KEY_L_START_TIME,startDate);
map.put(MetricsFilter.KEY_S_START_TIME,endDate)
```

Adding a completion date filter

To add a completion date filter (02/01/2005 < date <03/01/2005)

```
Calendar startDate=Calendar.getInstance();
startDate.set(2006,0,1);
Calendar endDate=Calendar.getInstance();
endDate.set(2006,1,1);
map.put(MetricsFilter.KEY_L_COMPLETION_TIME,startDate);
map.put(MetricsFilter.KEY_S_COMPLETION_TIME,endDate)
```

Narrowing requests to a specific initiator

To narrow down counted requests to a specific initiator

```
map.put(MetricsFilter.KEY_INITIATOR,"cn=admin,ou=idmsample,o=novell");
```

Narrowing requests to a specific recipient

To narrow down counted requests to a specific recipient

```
map.put(MetricsFilter.KEY_RECIPIENT,"cn=admin,ou=idmsample,o=novell");
```

B.3.2 Other Examples

The following examples illustrate the use of various methods for retrieving workflow counts.

Retrieving decision counts for a team

This example describes how to retrieve the various decision outcomes of a team. The team's DN is required and can be obtained by using the getTeams() method:

```
FilterConstants constants=new FilterConstants();
Map<MetricsFilter, Object> map = new HashMap<MetricsFilter, Object>();
```



```

map.put (MetricsFilter.KEY_ACTIVITY_END,
constants.getActivityApproved());
double accepted = stubWrapper.getTeamDecisionCount (processId,
processVersion, teamDN, map);
map.put (MetricsFilter.KEY_ACTIVITY_END,
constants.getActivityDenied());
double denied = stubWrapper.getTeamDecisionCount (processId,
processVersion, teamDN, map);
map.put (MetricsFilter.KEY_ACTIVITY_END,
constants.getActivityReassigned());
double reassigned = stubWrapper.getTeamDecisionCount (processId,
processVersion, teamDN, map);
map.put (MetricsFilter.KEY_ACTIVITY_END,
constants.getActivityRefused());
double refused = stubWrapper.getTeamDecisionCount (processId,
processVersion, teamDN, map);

```

Retrieving decision counts for requests where team members are recipients

This example describes how to retrieve the various decisions outcomes for requests for which members of the team act as recipients

```

FilterConstants constants = new FilterConstants();
Map<MetricsFilter, Object> map = new HashMap<MetricsFilter, Object>();
map.put (MetricsFilter.KEY_APPROVAL_STATUS,
constants.getActivityApproved());
double accepted = stubWrapper.getTeamRecipientCount (processId,
processVersion, teamDN, map);
map.put (MetricsFilter.KEY_APPROVAL_STATUS,
constants.getApprovalStatusDenied());
double denied = stubWrapper.getTeamRecipientCount (processId,
processVersion, teamDN, map);
map.put (MetricsFilter.KEY_APPROVAL_STATUS,
constants.getApprovalStatusError());
double error = stubWrapper.getTeamRecipientCount (processId,
processVersion, teamDN, map);
map.put (MetricsFilter.KEY_APPROVAL_STATUS,
constants.getApprovalStatusError());
double retracted = stubWrapper.getTeamRecipientCount (processId,
processVersion, teamDN, map);
map.put (MetricsFilter.KEY_APPROVAL_STATUS,
constants.getApprovalStatusRefused());
double refused = stubWrapper.getTeamRecipientCount (processId,
processVersion, teamDN, map);

```

Retrieving requests that have been claimed but not acted on

This example describes how to retrieve the requests started after 03/01/2006 that have been claimed but not acted upon.

```

Map<MetricsFilter, Object> map = new HashMap<MetricsFilter, Object>();

```

```

Calendar startDate=Calendar.getInstance();
startDate.set(2006,2,1);
map.put(MetricsFilter.KEY_L_START_TIME,startDate);
MetricsResultset rset = stubWrapper.getLongestClaimed(processId,
processVersion, map);

```

Retrieving the longest running requests started by a particular user

This example describes how to retrieve the longest running requests that have been started by initiator "cn=admin,ou=idmsample,o=novell";

```

Map<MetricsFilter, Object> map = new HashMap<MetricsFilter, Object>();
map.put(MetricsFilter.KEY_INITIATOR, "\"cn=admin,ou=idmsample,o=novell\"");
;
MetricsResultset rset = stubWrapper.getLongestRunning(processId,
processVersion, map);

```

Retrieving activity inventory

This example describes the average inventory for users handling decision with activity id "managerApproval" between 01/01/2006 and 02/01/2006

```

Map<MetricsFilter, Object> map = new HashMap<MetricsFilter, Object>();
Calendar startDate=Calendar.getInstance();
startDate.set(2006,0,1);
Calendar endDate=Calendar.getInstance();
endDate.set(2006,1,1);
MetricsResultset rset = stubWrapper.getActivityInventory(processId,
processVersion,"managerApproval", startDate, endDate, map );

```

Retrieving the Claimed Throughput and Inventory for a Team

This example describes the team's throughput and inventory over the time interval between 01/01/2006 and 02/01/2006

```

Map<MetricsFilter, Object> map = new HashMap<MetricsFilter, Object>();
Calendar startDate=Calendar.getInstance();
startDate.set(2006,0,1);
Calendar endDate=Calendar.getInstance();
endDate.set(2006,1,1);
double throughput =
stubWrapper.getClaimedThroughputCalendarDays(processId,
processVersion, startDate, endDate,teamDN, map);
double inventory = stubWrapper.getClaimedInventory(processId,
processVersion, startDate, endDate, teamDN, map)

```

Provisioning Web Service

C

This section describes the Provisioning Web Service, which allows SOAP clients to access Provisioning functionality. Topics include:

- ♦ [Section C.1, “About the Provisioning Web Service,” on page 399](#)
- ♦ [Section C.2, “Developing Clients for the Provisioning Web Service,” on page 400](#)
- ♦ [Section C.3, “Provisioning Web Service API,” on page 411](#)

C.1 About the Provisioning Web Service

The Identity Manager provisioning module includes a workflow system that executes approval flows. A workflow process is based on a provisioning request definition, which is an XML document stored in the Identity Vault. The provisioning request definition describes an arbitrary topology using activities and links. For example, a provisioning request to grant an entitlement might have a workflow that collects approvals from relevant users and writes the entitlement to the directory.

To support access by third-party software applications, the provisioning workflow system includes a Web service endpoint. The endpoint offers all provisioning functionality (for example, allowing SOAP clients to start a new approval flow, or list currently executing flows). The Web service is built using the Novell Web Service SDK (WSSDK), which supports the WS-I Basic Profile, thus guaranteeing interoperability with other standards based SOAP implementations.

This Appendix describes the provisioning Web service in detail and shows how to access it using the Web or by writing a Java or C# client. We provide an overview of the operations in the SOAP endpoint and describe how to use the Web interface. We show how to develop a Java client using the SOAP toolkit included with Identity Manager provisioning, followed by how to write a C# client using Mono. The sample source code a the Java client and associated ANT build file is provided.

C.1.1 Provisioning Web Service Overview

Identity Manager is composed of two main systems: the Identity Vault and the workflow application. The Identity Vault is capable of connecting to a large number of different systems such as databases, financial systems, and other enterprise applications, and keep these systems synchronized. The rules for synchronizing the remote systems can be very complex and the Identity Vault engine supports a sophisticated scripting language for expressing the rules.

The workflow application is composed of several subsystems. The User Application provides a user-interface for workflows. The User Application is a Web application for requesting and managing approval flows. The Web application runs in a portal, which also includes administration portlets. The workflow application contains a security layer, a directory abstraction layer and a logging subsystem, which can send log events to Novell Audit and Novell Sentinel. The workflow subsystem is responsible for executing approval flows. The User Application runs on an application server (for example, JBoss) and uses a database (for example, Oracle, MySQL) for persistence.

The Web service for the workflow system is only used by the user application driver, which is capable of listening to certain events emitted by the Identity Vault engine and convert these events into an appropriate SOAP message. For example, when a specific attribute in the Identity Vault

changes, the Identity Vault engine emits an event, which the User Application picks up from the subscriber channel. The user application driver then sends a SOAP message to the provisioning Web service to start a new approval flow.

C.1.2 Provisioning Web Service Method Categories

The methods provided by the provisioning Web service endpoint are divided into six categories:

Table C-1 *Provisioning Web Service Operation Categories*

Category	Description
Comments	Methods for retrieving comments and for adding a comment to a pending user activity
Configuration	Methods for getting and setting configuration parameters for the workflow system (for example, timeouts, thread pool settings).
Miscellaneous	Several unrelated methods (for example, for getting a JPG with a provisioning request's topology, for getting the XML definition of a provisioning request, and for getting the XML for the request form).
Processes	Methods for getting information about running and completed workflow processes.
Provisioning Requests	Methods for working with provisioning requests (for example, listing available provisioning requests, listing provisioning categories)
Work Entries	Methods for retrieving and manipulating work entries (items awaiting approval).

The methods provided by the provisioning Web service are described in detail in [Section C.3, “Provisioning Web Service API,”](#) on page 411.

C.2 Developing Clients for the Provisioning Web Service

This section includes the following topics:

- ♦ [Section C.2.1, “Web Access to the Provisioning Web Service,”](#) on page 400
- ♦ [Section C.2.2, “A Java Client for the Provisioning Web Service,”](#) on page 402
- ♦ [Section C.2.3, “Developing a Mono Client,”](#) on page 408
- ♦ [Section C.2.4, “Sample Ant File,”](#) on page 409
- ♦ [Section C.2.5, “Sample Log4J File,”](#) on page 411

C.2.1 Web Access to the Provisioning Web Service

A SOAP-based Web service is usually accessed by inserting a SOAP message in the body of an HTTP Post request. The Web service toolkit used to build the provisioning Web service also supports

access using HTTP GET. In other words, you can open the URL of the Web service endpoint in a browser and interact with the Web service. In particular, the provisioning Web service lets you invoke each of its operations.

Accessing the Test Page

You can access the provisioning Web Service endpoint using a URL similar to the following:

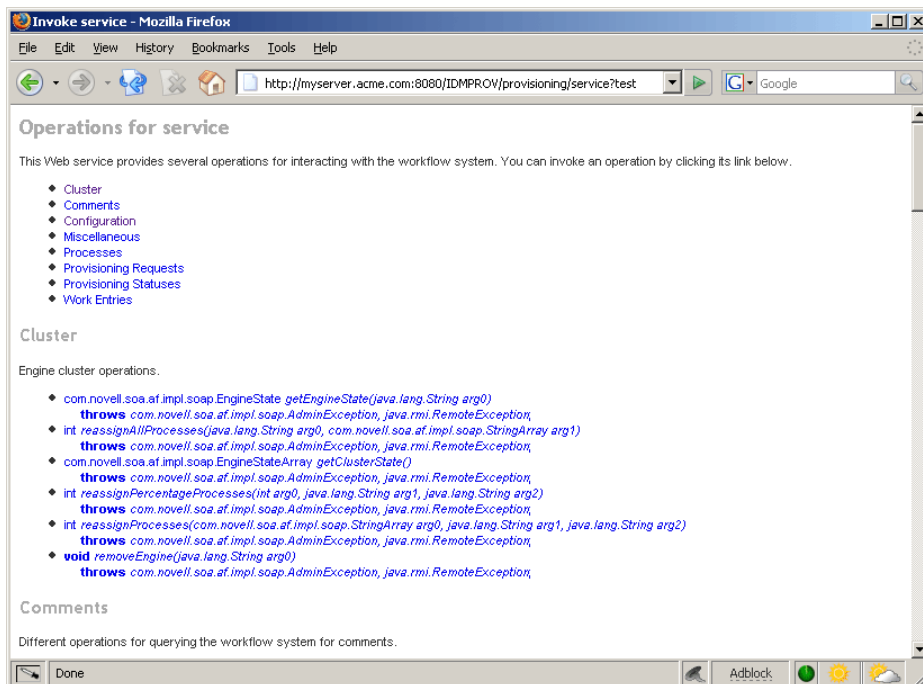
```
http://server:port/warcontext/provisioning/service?test
```

For example, if your server is named “myserver”, your User Application is listening on port 8080, and your User Application war file is named “IDMPROV”, the URL would be:

```
http://myserver:8080/IDMPROV/provisioning/service?test
```

The following page is displayed:

Figure C-1 Web Service Test Page



Entering Arguments for Operations

To see an example of an operation that is particularly useful to invoke from the browser, scroll down to the *Miscellaneous* section and click *getGraph*.

NOTE: The Graphviz program must be installed on the computer where the application server and the IDM User Application is running. For more information about Graphviz, see [Graphviz \(http://www.graphviz.org\)](http://www.graphviz.org).

A page is displayed that allows you to enter the parameters for the *getGraph* method.

Figure C-2 Parameters for *getGraph* Method

Enter Parameters to Invoke *getGraph*

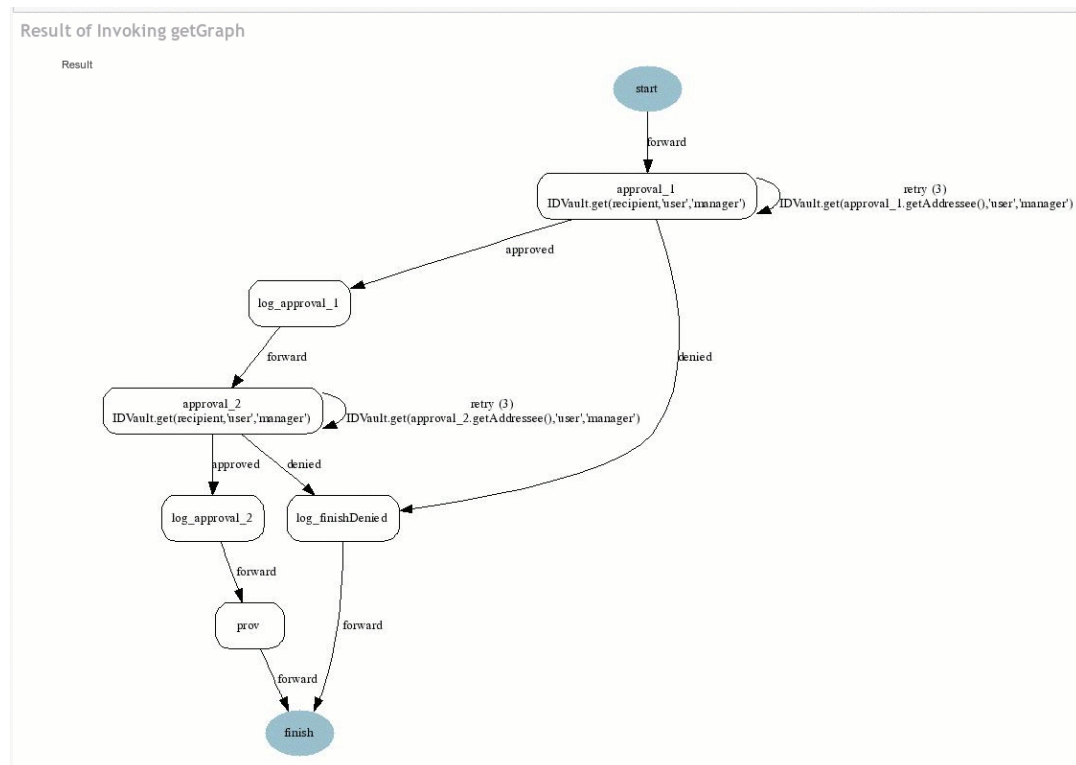
Get JPG image for workflow.

`processId` (java.lang.String):

[Back to home.](#)

The method takes one argument, which is the distinguished name of a provisioning request. Enter the DN, and the underlying workflow is displayed as a JPG file..

Figure C-3 Output of *getGraph*



C.2.2 A Java Client for the Provisioning Web Service

This section describes how to develop a simple Java client for the provisioning Web service, which lists all the processes in the workflow system. For complete source code for the client, see [“Sample Code for the Java Client” on page 406](#).

Prerequisites

To develop a Java client you must install a supported Java Developer’s Kit (see [System Requirements for Identity Manager \(http://www.novell.com/documentation/idm35/](#)

<index.html?page=/documentation/idm35/install/data/b2mbjps.html>)). Also, a client program needs the following JAR files:

```
activation.jar
commons-httpclient.jar
IDMfw.jar
log4j.jar
saa-j-api.jar
wssdk.jar
commons-codec-1.3.jar
commons-logging.jar
jaxrpc-api.jar
mail.jar
workflow.jar
xpp3.jar
```

Developing a Java Client

Developing a client that accesses a Web service consists of two steps:

- ♦ Get the stub, which is the object that represents the remote service
- ♦ Invoke one or more of the operations available in the remote service

The Java programming model for Web services is very similar to RMI. The first step is to lookup the stub using JNDI:

```
InitialContext ctx = new InitialContext();
ProvisioningService service = (ProvisioningService)
ctx.lookup("xmlrpc:soap:com.novell.soa.af.impl.soap.ProvisioningService");
Provisioning prov = service.getProvisioningPort();
```

The first line of code creates the initial context for JNDI lookups. The second line looks up the service object, which is a kind of factory that can be used to retrieve the stub for the provisioning Web service. The last line gets the provisioning stub from the service.

Before invoking an operation on the provisioning stub, it is necessary to set some properties, including the credentials used for authentication on the service, as well as the endpoint URL.

```
Stub stub = (Stub) prov;
// set username and password
stub._setProperty(Stub.USERNAME_PROPERTY, USERNAME);
stub._setProperty(Stub.PASSWORD_PROPERTY, PASSWORD);
// set the endpoint URL
stub._setProperty(Stub.ENDPOINT_ADDRESS_PROPERTY, url);
```

These and other stub properties are described in more detail in [“Frequently Used Stub Constants” on page 404](#). Now that we have a fully configured stub, we can invoke the `getAllProcesses` operation and dump information about each of the processes returned on the console:

```
// invoke the getAllProcesses method
ProcessArray array = prov.getAllProcesses();
Process[] procs = array.getProcess();
```

```

// print process array
System.out.println("list of all processes:");
if (procs != null) {
for (int i = 0; i < procs.length; i++) {
System.out.println(" process with request identifier " +
procs[i].getRequestId());
System.out.println(" initiator = " + procs[i].getInitiator());
System.out.println(" recipient = " + procs[i].getRecipient());
System.out.println(" processId = " + procs[i].getProcessId());
System.out.println(" created = " +
8
9
procs[i].getCreationTime().getTime());
if (null != procs[i].getCompletionTime()) {
System.out.println(" completed = " +
procs[i].getCompletionTime().getTime());
}
System.out.println(" approval status = " +
procs[i].getApprovalStatus());
System.out.println(" process status = " +
procs[i].getProcessStatus());
if (i != procs.length - 1)
System.out.println();
}
}

```

A method invocation on the stub results in a SOAP message being sent using the HTTP transport to the provisioning Web service. For operations that have arguments, the stub takes care of marshaling those Java objects into XML. The Web service returns a SOAP message, and the stub unmarshals the XML, in this case converging it into a ProcessArray Java object.

Running the Client

The sample ANT build file has a target for running the client (see [“Sample Ant File” on page 409](#)). The client needs the JAR files described in [“Prerequisites” on page 402](#) to be in the CLASSPATH. You can change the code to have a different default address for the provisioning Web service SOAP endpoint, or simply specify it as a command line argument. For example:

```
ant -Durl=http://www.company.com:80/IDMProv/provisioning/service run
```

Frequently Used Stub Constants

The com.novell.soa.ws.portable.Stub class (which is part of WSSDK) supports several properties that can be used to configure a stub instance (for example, to fine-tune aspects of the HTTP communication). The following table lists a small subset of these properties, which are frequently used:

Table C-2 *Provisioning Web Service Stub Constants*

Property	Type	Description
ENDPOINT_ADDRESS_PROPERTY	java.lang.String	The URL of the Web service. The URL protocol scheme can be HTTP or HTTPS depending on the requirements of the server. The path portion should be: <code>/IDMProv/provisioning/service</code>
HTTP_HEADERS	java.util.Map	Additional HTTP headers as String name/value pairs.
HTTP_TIME_OUT	java.lang.Integer	The number of seconds to wait to establish a connection to the host before timing out.
HTTP_MAX_TOTAL_CONNECTIONS	java.lang.Integer	The number of concurrent connections that this client program can establish to all server hosts it accesses. The default limit is 20.
HTTP_MAX_HOST_CONNECTIONS	java.lang.Integer	The number of concurrent connections this client program can establish to an individual server host. The default limit is 2. This value may not exceed that of <code>HTTP_MAX_TOTAL_CONNECTIONS</code> , so if a client requires more than 20 connections to the server, it must also set <code>HTTP_MAX_TOTAL_CONNECTIONS</code> to the desired value.
USERNAME	java.lang.String	The user ID for HTTP authentication.
PASSWORD	java.lang.String	The password for HTTP authentication.
HTTP_PROXY_HOST	java.lang.String	The host DNS name of a proxy. Setting this property requires setting <code>HTTP_PROXY_PORT</code> as well.
HTTP_PROXY_PORT	java.lang.Integer	The port to use on a proxy. Setting this property requires setting <code>HTTP_PROXY_HOST</code> as well.
HTTP_PROXY_AUTH_SCHEME	java.lang.Integer	The authentication scheme (Basic or Digest) to use for a proxy.
HTTP_PROXY_USERNAME	java.lang.String	The user ID for HTTP authentication using a proxy.
HTTP_PROXY_PASSWORD	java.lang.String	The password for HTTP authentication via proxy.

The TCP Tunnel

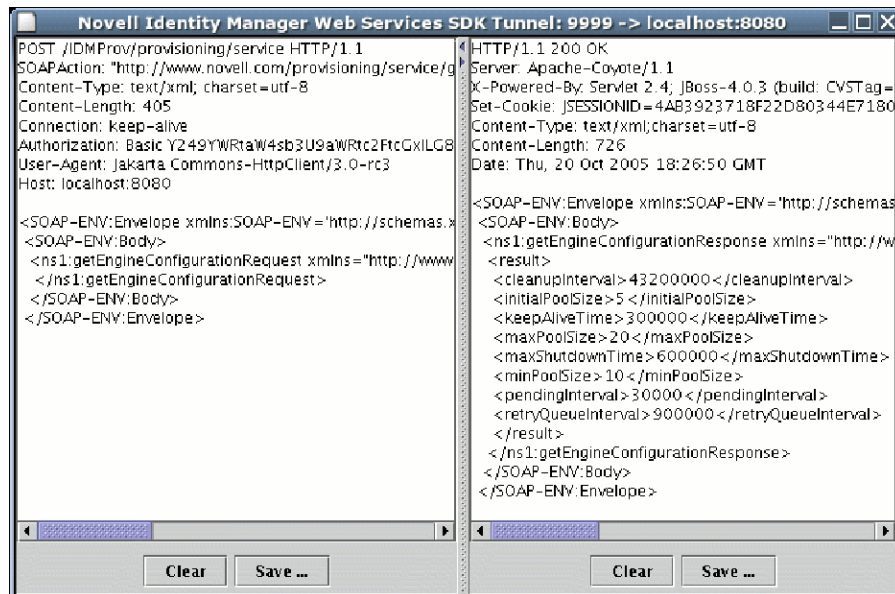
The TCP Tunnel is a useful tool for looking at the SOAP messages that are exchanged between a client and a server. The ANT build file (see [“Sample Ant File” on page 409](#)) has a target for starting the tunnel. Once the tunnel starts you need to enter the port on which the tunnel will listen, and the host/port of the remote Web service. The default settings cause the tunnel to listen on port 9999 and connect to a service running on localhost port 8080. The client program (see [“Developing a Java Client” on page 403](#)) uses the first command line parameter to set the

ENDPOINT_ADDRESS_PROPERTY. Using the default values, you can run the client using the following command, after starting the tunnel:

```
ant -Durl=http://localhost:9999/IDMProv/provisioning/service run
```

Figure C-4 shows the TCP tunnel with a request SOAP message in the left panel and the message in the right panel.

Figure C-4 TCP Tunnel



Sample Code for the Java Client

The following is the code for the Java client for listing all processes in the workflow system

```
package com.novell.examples;

import javax.naming.InitialContext;
import com.novell.soa.af.impl.soap.AdminException;
import com.novell.soa.af.impl.soap.Process;
import com.novell.soa.af.impl.soap.ProcessArray;
import com.novell.soa.af.impl.soap.Provisioning;
import com.novell.soa.af.impl.soap.ProvisioningService;
import com.novell.soa.ws.portable.Stub;

public class Client
{
    private static final String USERNAME = "admin";
    private static final String PASSWORD = "test";

    public static void main(String[] args)
    {
        try {
            String url = args.length > 0 ? args[0] :
                "http://localhost:8080/IDMProv/provisioning/service";
            listProcesses(url);
        } catch (AdminException ex) {
```

```

System.out.println("command failed: " + ex.getReason());
} catch (Exception ex) {
ex.printStackTrace();
}
}
private static void listProcesses(String url)
throws Exception
{
// get the stub
InitialContext ctx = new InitialContext();
ProvisioningService service = (ProvisioningService)
ctx.lookup("xmlrpc:soap:com.novell.soa.af.impl.soap.ProvisioningService");
Provisioning prov = service.getProvisioningPort();
Stub stub = (Stub) prov;
// set username and password
stub._setProperty(Stub.USERNAME_PROPERTY, USERNAME);
stub._setProperty(Stub.PASSWORD_PROPERTY, PASSWORD);
// set the endpoint URL
stub._setProperty(Stub.ENDPOINT_ADDRESS_PROPERTY, url);
// invoke the getAllProcesses method
ProcessArray array = prov.getAllProcesses();
Process[] procs = array.getProcess();
// print process array
System.out.println("list of all processes:");
if (procs != null) {
for (int i = 0; i < procs.length; i++) {
System.out.println(" process with request identifier " +
procs[i].getRequestId());
System.out.println(" initiator = " + procs[i].getInitiator());
System.out.println(" recipient = " + procs[i].getRecipient());
System.out.println(" processId = " + procs[i].getProcessId());
System.out.println(" created = " +
procs[i].getCreationTime().getTime());
if (null != procs[i].getCompletionTime()) {
System.out.println(" completed = " +
procs[i].getCompletionTime().getTime());
}
}
17
18
System.out.println(" approval status = " +
procs[i].getApprovalStatus());
System.out.println(" process status = " +
procs[i].getProcessStatus());
if (i != procs.length - 1)
System.out.println();
}
}

```

```
}
}
}
```

C.2.3 Developing a Mono Client

The previous section described how to create a Java client using the Web service toolkit and the pre-compiled stub code included with Identity Manager. This section describes how to develop a client using just the WSDL for the provisioning Web service. This example uses Mono and creates a C# client that changes the default retention time of 120 days for completed workflows to 30.

Prerequisites

To get started, you need to download Mono and install it on your system (see the [Mono Project Website \(http://www.mono-project.com/\)](http://www.mono-project.com/)). The version of Mono available at the time this document was written did not support complex schema types in which an element has the nillable attribute set to true. Because this construct is used in the provisioning WSDL, you must manually edit the Provisioning.WSDL file and remove the three places where `nillable="true"` is used.

Generating the Stub

Compared to the Java client developed in “[Developing a Java Client](#)” on page 403, there is one additional step required when building the C# client. Since the stub for accessing the Web service SOAP endpoint is not provided, you must generate the stub from the WSDL document. Mono includes a compiler called `wsdl` that processes the WSDL file and creates the stub. You can download the WSDL file from your User Application server by accessing the following URL:

```
http://myserver:8080/IDMProv/provisioning/service?wsdl
```

Replace “myserver” with the name of your server, and “IDMProv” with the name of your User Application war file.

Compile the WSDL file using the following command:

```
wsdl Provisioning.wsdl
```

This will generate a C# file called `ProvisioningService.cs`, which you need to compile into a DLL using the following Mono C# compiler command:

```
mcs /target:library /r:System.Web.Services.dll ProvisioningService.cs
```

Compared to the Java client, the resulting `ProvisioningService.dll` file is the equivalent of `workflow.jar`, which contains the stub code and supporting classes for accessing the provisioning Web service. The following is the source code for the simple C# client that sets the flow retention time and displays the new value on the console:

```
using System;
using System.Net;
class provclient {
public static void Main(string [] args) {
// create the provisioning service proxy
ProvisioningService service = new ProvisioningService();
// set the credentials for basic authentication
service.Credentials = new NetworkCredential("admin", "test");
service.PreAuthenticate = true;
```

```
// set the value for completed request retention to 30 days
setCompletedProcessTimeoutRequest req = new
setCompletedProcessTimeoutRequest();
11
12
req.arg0 = 30;
service.setCompletedProcessTimeout(req);
// display the new value on the console
getCompletedProcessTimeoutResponse res =
service.getCompletedProcessTimeout(new
getCompletedProcessTimeoutRequest());
Console.WriteLine(res.result);
}
}
```

You need to edit the file using the administrator credentials on your deployed Identity Manager system. Compile the client using the following command:

```
mcs /r:ProvisioningService.dll /r:System.Web provclient.cs
```

This generates the provclient.exe file.

Running the Client

Use the following command to run the client:

```
mono provclient.exe
```

C.2.4 Sample Ant File

The sample Ant file includes useful targets for extracting the necessary JAR files from the Identity Manager installation, compiling and running the Java client, and for launching the TCP Tunnel.

```
<?xml version="1.0"?>
<project name="client" default="all" basedir=".">
<target name="all" depends="clean, extract, compile"></target>
<!-- main clean target -->
<target name="clean">
<delete quiet="true" dir="classes"/>
<delete quiet="true" dir="lib"/>
</target>
<!-- init sets up the build environment -->
<target name="init">
<mkdir dir="classes"/>
<copy todir="${basedir}/lib">
<fileset dir="${basedir}" includes="log4j.properties"/>
</copy>
<!-- classpath -->
<path id="CLASSPATH">
<pathelement location="${basedir}/classes"/>
<fileset dir="${basedir}/lib" includes="*.jar"/>
```

```

</path>
</target>
<!-- extract -->
<target name="extract">
  <property name="idm.home" value="/opt/novell/idm3"/>
  <property name="jboss.lib" value="${idm.home}/jboss-4.0.3/server/
IDMProv/lib"/>
  <mkdir dir="lib"/>
  <unzip src="${idm.home}/IDMProv.war" dest="${basedir}/lib">
    <patternset>
      <include name="WEB-INF/lib/commons-codec-1.3.jar"/>
      <include name="WEB-INF/lib/commons-httpclient.jar"/>
      <include name="WEB-INF/lib/commons-logging.jar"/>
      <include name="WEB-INF/lib/jaxrpc-api.jar"/>
      <include name="WEB-INF/lib/saaj-api.jar"/>
      <include name="WEB-INF/lib/xpp3.jar"/>
      <include name="WEB-INF/lib/workflow.jar"/>
      <include name="WEB-INF/lib/wssdk.jar"/>
      <include name="WEB-INF/lib/IDMfw.jar"/>
    </patternset>
  </unzip>
  <move todir="${basedir}/lib">
    <fileset dir="${basedir}/lib/WEB-INF/lib" includes="*.jar"/>
  </move>
  <delete quiet="true" dir="${basedir}/lib/WEB-INF"/>
  <copy todir="${basedir}/lib">
    <fileset dir="${jboss.lib}" includes="activation.jar, mail.jar,
log4j.jar"/>
  </copy>
</target>
18
19
<!-- tunnel -->
<target name="tunnel" depends="init">
  <java classname="com.novell.soa.ws.impl.tools.tcptunnel.Tunnel"
fork="true"
spawn="true">
    <classpath refid="CLASSPATH"/>
  </java>
</target>
<!-- compile -->
<target name="compile" depends="init">
  <javac srcdir="${basedir}" destdir="classes"
includes="Client.java">
    <classpath refid="CLASSPATH"/>
  </javac>
</target>

```

```

<!-- run -->
<target name="run" depends="init">
<property name="url" value="http://localhost:8080/IDMProv/
provisioning/service"/>
<java classname="com.novell.examples.Client" fork="true">
<arg line="${url}"/>
<classpath refid="CLASSPATH"/>
</java>
</target>
</project>

```

C.2.5 Sample Log4J File

The following log4j file sets the default log level to “error”:

```

log4j.rootCategory=ERROR, R
log4j.appender.R=org.apache.log4j.ConsoleAppender
log4j.appender.R.layout=org.apache.log4j.PatternLayout
log4j.appender.R.layout.ConversionPattern=%-5p: %m%n

```

C.3 Provisioning Web Service API

This section provides details about the Provisioning Web service methods.

All of the methods throw `com.novell.soa.af.impl.soap.AdminException` and `java.rmi.RemoteException`. To improve readability, the throws clause has been omitted from the method signatures.

This section includes the following topics:

- ♦ [Section C.3.1, “Processes,” on page 411](#)
- ♦ [Section C.3.2, “Provisioning,” on page 421](#)
- ♦ [Section C.3.3, “Work Entries,” on page 434](#)
- ♦ [Section C.3.4, “Comments,” on page 452](#)
- ♦ [Section C.3.5, “Configuration,” on page 459](#)
- ♦ [Section C.3.6, “Miscellaneous,” on page 463](#)
- ♦ [Section C.3.7, “Cluster,” on page 466](#)

C.3.1 Processes

This section provides reference information for each Processes method. The methods include:

- ♦ [“getProcessesByQuery” on page 412](#)
- ♦ [“getProcessesByStatus” on page 413](#)
- ♦ [“getProcesses” on page 413](#)
- ♦ [“getAllProcesses” on page 414](#)
- ♦ [“getProcessesById” on page 415](#)
- ♦ [“terminate” on page 415](#)

- ♦ “getProcess” on page 416
- ♦ “getProcessesByCreationTime” on page 417
- ♦ “getProcessesByApprovalStatus” on page 418
- ♦ “getProcessesByRecipient” on page 418
- ♦ “getProcessesByInitiator” on page 418
- ♦ “setResult” on page 419
- ♦ “getProcessesByCreationInterval” on page 420

getProcessesByQuery

Used to get information about processes.

Method Signature

```
com.novell.soa.af.impl.soap.ProcessArray
getProcessesByQuery(com.novell.soa.af.impl.soap.T_ProcessInfoQuery
query, int maxRecords)
```

Example

```
//
// Query information about processes for a user that are
running and
// have not been approved yet.
String logic = "AND";
T_ProcessInfoOrder order = T_ProcessInfoOrder.APPROVAL_STATUS;
int CHOICE_SIZE = 4;
Integer approvalStatusInteger = new
Integer(ProcessConstants.PROCESSING);
Integer processStatusInteger = new
Integer(ProcessConstants.RUNNING);
//
// Setup the query with the above params
T_ProcessInfoQueryChoice [] choice = new
T_ProcessInfoQueryChoice[CHOICE_SIZE];
choice[0] = new T_ProcessInfoQueryChoice();
choice[0].setApprovalStatus(approvalStatusInteger);
choice[1] = new T_ProcessInfoQueryChoice();
choice[1].setProcessStatus(processStatusInteger);
choice[2] = new T_ProcessInfoQueryChoice();
choice[2].setRecipient(recipient);
choice[3] = new T_ProcessInfoQueryChoice();
choice[3].setRequestId(requestId);

int maxRecords = -1;
T_ProcessInfoQuery processInfoQuery =
    new T_ProcessInfoQuery(T_Logic.fromString(logic),
order, choice);
```



```

        ProcessArray processArray =
stub.getProcessesByQuery(processInfoQuery, maxRecords);

```

getProcessesByStatus

Used to get information about processes with a specified status (for example, running processes).

Method Signature

```

public com.novell.soa.af.impl.soap.ProcessArray
getProcessesByStatus(com.novell.soa.af.impl.soap.T_ProcessStatus
status)

```

Example

```

        T_ProcessStatus processStatus = T_ProcessStatus.Running;
        //
        // Get processes by status
        ProcessArray processArray =
stub.getProcessesByStatus(processStatus);
        Process [] process = processArray.getProcess();

```

getProcesses

Used to get information about processes, specified by processID.

Method Signature

```

com.novell.soa.af.impl.soap.ProcessArray getProcesses(java.lang.String
id, long time, com.novell.soa.af.impl.soap.T_Operator op,
java.lang.String initiator, java.lang.String recipient)

```

Parameters

Parameter	Description
processId	The process Id (java.lang.String).
creationTime	The time at which the process was started (long).
op	The operator to use. The operators are: EQ - equals LT - less than LE - less than or equal to GT - greater than GE - greater than or equal to
initiator	The initiator of the workflow.
recipient	The recipient of the approval activity.

Example

```

        int processMatchCount = 0;
        T_Operator operator = T_Operator.GT;
        long currentTimeInMillis = System.currentTimeMillis();

```

```

String [] requestIds = requestIdArray.getString();
//
// Initialize and start a provisioning request
HashMap provMap = new HashMap();
provMap.put(Helper.RECIPIENT, recipient);
provMap.put("Provisioning_Request_To_Start_Key", "Enable
Active Directory Account (Mgr Approve-No Timeout)");
//
// Start request
// Calls method startProvisioningRequest on the provUtils
// utility object which refers to a utility class that does not
// ship with the Identity Manager User Application.
String requestId = provUtils.startProvisioningRequest(provMap, null);
sleep(5);

Process process = stub.getProcess(requestId);
if(process != null)
{
    String processId = process.getProcessId();
    String initiator = process.getInitiator();

    ProcessArray processArray = stub.getProcesses(processId,
currentTimeInMillis, operator, initiator, recipient);
}

```

getAllProcesses

Used to get information about all running and completed provisioning requests.

Method Signature

```
com.novell.soa.af.impl.soap.ProcessArray getAllProcesses()
```

Example

```

ProcessArray array = stub.getAllProcesses();
Process [] processes = array.getProcess();
if(_process != null)
{
    sb = new StringBuffer();
    sb.append("\nProcess List:");
    for(int index = 0; index < _process.length; index++)
    {
        String processId = _process[index].getProcessId();
        String approvalStatus =
        _process[index].getApprovalStatus();
        Calendar completionTime =
        _process[index].getCompletionTime();
        Calendar creationTime = _process[index].getCreationTime();
        String engineId = _process[index].getEngineId();
    }
}

```

```

        String proxy = _process[index].getProxy();
        String initiator = _process[index].getInitiator();
        String processName = _process[index].getProcessName();
        String processStatus = _process[index].getProcessStatus();
        String p_recipient = _process[index].getRecipient();
        String p_requestId = _process[index].getRequestId();
        int valueOfapprovalStatus =
        _process[index].getValueOfApprovalStatus();
        int valueOfprocessStatus =
        _process[index].getValueOfProcessStatus();
        String version = _process[index].getVersion();
    }

```

getProcessesById

Used to get information about a specific process, specified by the Process Id.

Method Signature

```

com.novell.soa.af.impl.soap.ProcessArray
getProcessesById(java.lang.String id)

```

Example

```

Process [] allProcesses = stub.getAllProcesses().getProcess();
if(allProcesses != null)
{
    String processId = allProcesses[0].getProcessId;
    ProcessArray array = stub.getProcessesById(processId);
    Process [] processes = array.getProcess();
}

```

terminate

Used to terminate a running provisioning request.

Method Signature

```

void terminate(java.lang.String requestId,
com.novell.soa.af.impl.soap.T_TerminationType state, java.lang.String
comment)

```

Parameters

Parametere	Description
requestId	The Id of the provisioning request.
state	The reason for terminating the process. The choices are: RETRACT ERROR

Parametere	Description
comment	Adds a comment about the terminate action.

Example

```
//
// Initialize and start a provisioning request
HashMap provMap = new HashMap();
provMap.put(Helper.RECIPIENT, recipient);
provMap.put(I"Provisioning_Request_To_Start_Key", "Enable Active
Directory Account (Mgr Approve-No Timeout)");
//
// Start request
// Calls method startProvisioningRequest on the provUtils
// utility object which refers to a utility class that does not
// ship with the Identity Manager User Application.
String requestId = provUtils.startProvisioningRequest(provMap, null);
sleep(5);
//
// Now retract the request
T_TerminationType terminationType = T_TerminationType.RETRACT;
stub.terminate(requestId, terminationType,
terminationType.getValue() + " the request");
```

getProcess

Used to get information about a running or completed provisioning request, specified by Request ID.

Method Signature

```
com.novell.soa.af.impl.soap.Process getProcess(java.lang.String
requestId)
```

Example

```
//
// Initialize and start a provisioning request
HashMap provMap = new HashMap();
provMap.put(Helper.RECIPIENT, recipient);
provMap.put(I"Provisioning_Request_To_Start_Key", "Enable Active
Directory Account (Mgr Approve-No Timeout)");
//
// Start request
// Calls method startProvisioningRequest on the provUtils
// utility object which refers to a utility class that does not
// ship with the Identity Manager User Application.
String requestId = provUtils.startProvisioningRequest(provMap,
null);
sleep(5);
```

```

Process process = stub.getProcess(requestId);
if(process != null)
{
    boolean bMatchProcess = false;
    if( (recipient.compareTo(process.getRecipient()) == 0) &&
(requestId.compareTo(process.getRequestId()) == 0) )
    {
        bMatchProcess = true;
    }
    if(bMatchProcess)
    {
        String msg = "Found process with requestId : " + requestId;
        LoggerUtils.sendToLogAndConsole(msg);
    }
    //
    // Assert if we could not find a match
    Assert.assertTrue("Could not find process with request id: " +
requestId, bMatchProcess);
}

```

getProcessesByCreationTime

Used to get information about processes created between the current time and the time at which the workflow process was created.

Method Signature

```

com.novell.soa.af.impl.soap.ProcessArray
getProcessesByCreationTime(long time,
com.novell.soa.af.impl.soap.T_Operator op)

```

Parameters

Parameter	Description
creationTime	The time at which the process was started.
op	The operator to use. The operators are: EQ - equals LT - less than LE - less than or equal to GT - greater than GE - greater than or equal to

Example

```

T_Operator operator = T_Operator.GT;
//
// Get processes with operator relative to the current time

```

```

        long currentTime = System.currentTimeMillis(); //
        currentDate.getTime();
        ProcessArray processArray =
stub.getProcessesByCreationTime(currentTime, operator);

```

getProcessesByApprovalStatus

Used to get information about processes with a specified approval status (Approved, Denied, or Retracted).

Method Signature

```

com.novell.soa.af.impl.soap.ProcessArray
getProcessesByApprovalStatus(com.novell.soa.af.impl.soap.T_ApprovalSta
tus status)

```

Example

```

        T_ApprovalStatus approvalStatus = T_ApprovalStatus.Approved;
        //
        // Get all the processes based upon approval status above
        ProcessArray processArray =
stub.getProcessesByApprovalStatus(approvalStatus);
        Process [] processes = processArray.getProcess();

```

getProcessesByRecipient

Used to get information about processes that have a specific recipient Id.

Method Signature

```

com.novell.soa.af.impl.soap.ProcessArray
getProcessesByRecipient(java.lang.String recipient)

```

Example

```

        String recipient = "cn=ablake,ou=users,ou=idmsample-
komodo,o=novell";

        //
        // Get processes by recipient
        ProcessArray processArray =
stub.getProcessesByRecipient(recipient);
        Process [] process = processArray.getProcess();

```

getProcessesByInitiator

Used to get information about processes that have a specific initiator Id.

Method Signature

```

com.novell.soa.af.impl.soap.ProcessArray
getProcessesByInitiator(java.lang.String initiator)

```

Example

```

        String initiator = "cn=admin,ou=idmsample-komodo,o=novell";

```

```

//
// Get processes by initiator
ProcessArray processArray =
stub.getProcessesByInitiator(initiator);
Process [] process = processArray.getProcess();

```

setResult

Used to set the entitlement result (approval status) of a previously completed provisioning request.

Method Signature

```

void setResult(java.lang.String requestId,
com.novell.soa.af.impl.soap.T_EntitlementState state,
com.novell.soa.af.impl.soap.T_EntitlementStatus status,
java.lang.String message)

```

Parameters

Parameter	Description
requestId	The Id of the provisioning request.
state	<p>The state of the provisioning request. The possible values are:</p> <p>Unknown Granted Revoked</p>
status	<p>The status of the provisioning request. The possible values are:</p> <p>Unknown Success Warning Error Fatal Submitted</p>
message	A message about the entitlement result.

Example

```

//
// Initialize and start a provisioning request
HashMap provMap = new HashMap();
provMap.put(Helper.RECIPIENT, recipient);
provMap.put(I"Provisioning_Request_To_Start_Key", "Enable Active
Directory Account (Mgr Approve-No Timeout)");
//
// Start request
// Calls method startProvisioningRequest on the provUtils

```

```

        // utility object which refers to a utility class that does not
        // ship with the Identity Manager User Application.
        String requestId = provUtils.startProvisioningRequest(provMap,
null);
        sleep(5);

        //
        // Get the process id for this running process
        Process process = stub.getProcess(requestId);
        String processId = null;
        if (process != null)
            processId = process.getProcessId();
        //
        // Reset the state of the provisioning request
        T_EntitlementState newEntitlementState =
T_EntitlementState.Revoked;
        T_EntitlementStatus newEntitlementStatus =
T_EntitlementStatus.Success;
        String comment = "Revoked the provisioning request";
        stub.setResult(processId, newEntitlementState,
newEntitlementStatus, comment);

```

getProcessesByCreationInterval

Used to get information about processes started between two specified times.

Method Signature

```

com.novell.soa.af.impl.soap.ProcessArray
getProcessesByCreationInterval(long start, long end)

```

Parameters

Parameter	Description
startTime	The start time (YYYY/MM/DD).
endTime	The end time (YYYY/MM/DD).

Example

```

        long startTime = System.currentTimeMillis();
        //
        // Initialize and start a provisioning request
        HashMap provMap = new HashMap();
        provMap.put(Helper.RECIPIENT, recipient);
        provMap.put(I"Provisioning_Request_To_Start_Key", "Enable Active
Directory Account (Mgr Approve-No Timeout)");
        //
        // Start request

```



```

        // Calls method startProvisioningRequest on the provUtils
        // utility object which refers to a utility class that does not
        // ship with the Identity Manager User Application.
        String requestId = provUtils.startProvisioningRequest(provMap,
null);
        sleep(5);

        long endTime = System.currentTimeMillis();
        //
        // Get all the processes between the start and end time
        ProcessArray processArray =
stub.getProcessesByCreationInterval(startTime, endTime);
        Process [] processes = processArray.getProcess();

```

C.3.2 Provisioning

This section provides reference information for each Provisioning method. The Provisioning methods include:

- ♦ [“multiStart” on page 421](#)
- ♦ [“start” on page 423](#)
- ♦ [“getAllProvisioningRequests” on page 425](#)
- ♦ [“getProvisioningRequests” on page 426](#)
- ♦ [“getProvisioningCategories” on page 427](#)
- ♦ [“startAsProxy” on page 427](#)
- ♦ [“getProvisioningStatuses” on page 428](#)
- ♦ [“startWithDigitalSignature” on page 430](#)
- ♦ [“startAsProxyWithDigitalSignature” on page 432](#)

multiStart

Used to start a workflow request for each specified recipient.

Method Signature

```

com.novell.soa.af.impl.soap.StringArray multiStart(java.lang.String
processId, com.novell.soa.af.impl.soap.StringArray recipients,
com.novell.soa.af.impl.soap.DataItemArray items)

```

Parameters

Parameter	Description
processId	The Id of the provisioning request to start.
recipients	The DN of each recipient.
dataItem	The list of data items for the provisioning request.

Example

```
ProvisioningRequestArray requestArray =
stub.getAllProvisioningRequests(recipient);

//
// If there are some then,
if(requestArray != null)
{
    String Id = " ";
    StringArray requestIdStringArray = null;
    String [] listOfRecipients = {recipient, addressee};
    //
    // Select a provisioning resource
    String requestNameToStart = "Enable Active Directory Account
(Mgr Approve-No Timeout)";
    //
    // Loop thru and find the request that we want to start
    ProvisioningRequest [] requests =
requestArray.getProvisioningrequest();
    for(int index = 0; index < requests.length; index++)
    {
        //
        // Is this the name of the request to start?
        if(requests[index].getName().compareTo(requestNameToStart)
== 0)
        {
            //
            // Get the current associated data items. Replicate a
new
            // dataitem array excluding the null values.
            Id = requests[index].getId();
            DataItem [] dataItem =
requests[index].getItems().getDataitem();
            if(dataItem != null)
            // Call method replicateDataItemArray on the
            // provUtils utility object, which refers to a
            // utility class that does not ship with the
            // Identity Manager User Application.
            {
                DataItemArray newDataItemArray =
provUtils.replicateDataItemArray(dataItem);
                //
                // Create a string array initializing with multiple
recipients
                StringArray listOfRecipientsStringArray = new
StringArray(listOfRecipients);
                //
```

```

        // Start the request for multiple recipients
        logStep("Calling stub.multiStart(" + Id +
",listOfRecipientsStringArray,newDataItemArray)");
        requestIdStringArray = stub.multiStart(Id,
listOfRecipientsStringArray, newDataItemArray);
    }
}
}

```

start

Used to start a provisioning request.

Method Signature

```
java.lang.String start(java.lang.String processId, java.lang.String
recipient, com.novell.soa.af.impl.soap.DataItemArray items)
```

Parameters

Parameter	Description
processId	The Id of the provisioning request to start.
recipient	The DN of each recipient.
dataItem	The list of data items for the provisioning request.

Example

```

//
// Initialize and start a provisioning request
HashMap provMap = new HashMap();
provMap.put(Helper.RECIPIENT, recipient);
provMap.put(I"Provisioning_Request_To_Start_Key", "Enable
Active Directory Account (Mgr Approve-No Timeout)");
//
// Start request
// Calls method startProvisioningRequest on the provUtils
// utility object which refers to a utility class that does not
// ship with the Identity Manager User Application.
String requestId = provUtils.startProvisioningRequest(provMap,
null);
sleep(5);

```

The example above calls the startProvisioningRequest method. This method is not part of the IDM User Application. We show it here to finish illustrating the example:

```

/**
 *Method to start a provisioning request using the supplied
 *Map and dataitem object. Handling of digital certificate
 *resources is also handled.
 * @param _map

```

```

        * @param _in_dataItem
        * @return String
        * @throws TestProgramException
        */
        public String startProvisioningRequest(Map _map, DataItem []
_in_dataItem) throws TestProgramException
    {
        String requestId = null;
        try
        {
            String recipient =(String)_map.get(Helper.RECIPIENT);
            String requestToStart =
            (String)_map.get(IProvisioningConstants.PROVISIONING_REQUEST_TO_START)
;

            String proxyUser
            =(String)_map.get(IWorkflowConstants.PROXY_USER);
            String digitalSignature =
            (String)_map.get(IDigitalSignatureConstants.DIGITAL_SIGNATURE);
            RationalTestScript.logInfo("Step: Calling
            startProvisioningRequest(_map)");
            //
            //Get the stub
            Provisioning stub =
            ServiceUtils.getInstance().getProvisioningStub();
            //
            //Get all the available resource requests for the recipient
            RationalTestScript.logInfo("Step: Calling
            stub.getAllProvisioningRequests(" + recipient + ")");
            ProvisioningRequestArray requestArray =
            stub.getAllProvisioningRequests(recipient);

            if(requestArray != null)
            {
                //
                //Get the provisioning request from the array
                ProvisioningRequest request =
                getProvisioningRequestFromArray(requestArray, requestToStart);
                if(request != null)
                {
                    DataItem [] dataItem = null;
                    DataItemArray newDataItemArray = null;
                    //
                    // If the supplied data item is null then just replicate
                    // what currently exists with the request.
                    if(_in_dataItem == null)
                    {
                        //

```

```

        // Use the current data item associated with the request
        dataItem = request.getItems().getDataitem();
        if(dataItem != null)
        {
            newDataItemArray = replicateDataItemArray(dataItem);
        }
    }
    else
    {
        //
        // Set the incoming data item array
        newDataItemArray = new DataItemArray();
        newDataItemArray.setDataitem(_in_dataItem);
    }
    //
    // Start the Provisioning request for the recipient
    if(proxyUser == null && digitalSignature == null)
    {
        RationalTestScript.logInfo("Step: Calling stub.start(" +
request.getId() + "," + recipient + "dataItemArray)");
        requestId = stub.start(
            request.getId(),
            recipient,
            newDataItemArray);
    }
    else if(proxyUser != null && digitalSignature == null)
    {
        .
        .
        .
        .
        .
    }

```

getAllProvisioningRequests

Used to return an array of available provisioning requests.

Method Signature

```

com.novell.soa.af.impl.soap.ProvisioningRequestArray
getAllProvisioningRequests(java.lang.String recipient)

```

Example

```

//

// Get all the provisioning requests for this recipient

```

```

        ProvisioningRequestArray provReqArray =
stub.getAllProvisioningRequests(recipient);
        ProvisioningRequest [] provRequest =
provReqArray.getProvisioningrequest();
        if(provRequest != null)
        {
            String description = provRequest[0].getDescription();
            String category = provRequest[0].getCategory();
            String digitalSignatureType =
provRequest[0].getDigitalSignatureType();
            String requestId = provRequest[0].getId();
            DataItemArray itemArray = provRequest[0].getItems();
            String legalDisclaimer = provRequest[0].getLegalDisclaimer();
            String name = provRequest[0].getName();
            String operation = provRequest[0].getOperation();
        }

```

getProvisioningRequests

Used to return an array of provisioning requests for a specified category and operation.

Method Signature

```

com.novell.soa.af.impl.soap.ProvisioningRequestArray
getProvisioningRequests(java.lang.String recipient, java.lang.String
category, java.lang.String operation)

```

Parameters

Parameter	Description
recipient	The recipient of the provisioning request.
category	The category of the provisioning request.
operation	The provisioning request operation (0=Grant, 1=Revoke, 2=Both)

Example

```

        String operation = IProvisioningRequest.GRANT;
        try
        {
            //
            // Get the stub
            Provisioning stub =
ServiceUtils.getInstance().getProvisioningStub();
            logStep("Calling stub.getProvisioningCategories()");
            StringArray categoriesStringArray =
stub.getProvisioningCategories();
            String [] categories = categoriesStringArray.getString();
            //

```

```

        // Loop thru and get the provisioning requests for each
category
        for(int index = 0; index < categories.length; index++)
        {
            //
            // Get the provisioning request based upon recipient
            logStep("Calling stub.getProvisioningRequests(" + recipient
+ "," + categories[index] + "," + operation + ")");
            ProvisioningRequestArray provRequestArray =
stub.getProvisioningRequests(recipient, categories[index], operation);
            ProvisioningRequest [] provRequests =
provRequestArray.getProvisioningrequest();
        }

```

getProvisioningCategories

Used to get the list of available provisioning categories.

Method Signature

```
com.novell.soa.af.impl.soap.StringArray getProvisioningCategories()
```

Example

```

StringArray categoriesStringArray =
stub.getProvisioningCategories();
String [] categories = categoriesStringArray.getString();

```

startAsProxy

Used to start a workflow as a proxy.

Method Signature

```

java.lang.String startAsProxy(java.lang.String processId,
java.lang.String recipient, com.novell.soa.af.impl.soap.DataItemArray
items, java.lang.String proxyUser)

```

Parameters

Parameter	Description
processId	The Id of the provisioning request.
recipeint	The recipient of the provisioning request.
Items	The data items for the provisioning request.
proxyUser	The DN of the proxy user.

Example

```

ProvisioningRequestArray requestArray =
stub.getAllProvisioningRequests(recipient);
//
// If there are some then,

```

```

        if(requestArray != null)
        {
            String Id = " ";
            String requestId = " ";
            String requestNameToStart = "Enable Active Directory Account
(Mgr Approve-No Timeout)";
            //
            // Loop thru and find the request that we want to start
            ProvisioningRequest [] requests =
requestArray.getProvisioningrequest();
            for(int index = 0; index < requests.length; index++)
            {
                //
                // Is this the name of the request to start?
                if(requests[index].getName().compareTo(requestNameToStart)
== 0)
                {
                    //
                    // Get the current associated data items. Replicate a
new
                    // dataitem array excluding the null values.
                    Id = requests[index].getId();
                    DataItem [] dataItem =
requests[index].getItems().getDataitem();
                    if(dataItem != null)
                    {
                        // Call method replicateDataItemArray on the
                        // provUtils utility object, which refers to a
                        // utility class that does not ship with the
                        // Identity Manager User Application.
                        DataItemArray newDataItemArray =
provUtils.replicateDataItemArray(dataItem);
                        //
                        // Start the Provisioning request for the recipient
                        logStep("Calling stub.startAsProxy(" + Id + "," +
recipient + ",newDataItemArray," + proxyUser + ")");
                        requestId = stub.startAsProxy(Id, recipient,
newDataItemArray, proxyUser);
                    }
                }
            }
        }
    }
}

```

getProvisioningStatuses

Used to get the status of provisioning requests.

Method Signature

```
com.novell.soa.af.impl.soap.ProvisioningStatusArray  
getProvisioningStatuses (com.novell.soa.af.impl.soap.T_ProvisioningStat  
usQuery query, int maxRecords)
```

Parameters

Parameter	Description
query	<p>Used to specify the provisioning status query. The query has the following components:</p> <ul style="list-style-type: none">♦ <code>choice</code> - the parameters used to filter the results. You can specify multiple parameters. The possible parameters are: Recipient - a DN RequestID ActivityID Status (an integer) State (an integer) ProvisioningTime (YYYY/MM/DD) ResultTime (YYYY/MM/DD)♦ <code>logic</code> - AND or OR♦ <code>order</code> - the order in which to sort the results. Possible values for <code>order</code> are: ACTIVITY_ID RECIPIENT PROVISIONING_TIME RESULT_TIME STATE STATUS REQUEST_ID MESSAGE
maxRecords	<p>Used to specify maximum number of records to retrieve. A value of -1 returns unlimited records.</p>

Example

```
//  
// Initialize and start a provisioning request  
HashMap provMap = new HashMap();  
provMap.put (Helper.RECIPIENT, recipient);  
provMap.put ("Provisioning_Request_To_Start_Key", "Enable Active  
Directory Account (Mgr Approve-No Timeout)");  
//  
// Start request  
// Calls method startProvisioningRequest on the provUtils  
// utility object which refers to a utility class that does not  
// ship with the Identity Manager User Application.
```

```

        String requestId = provUtils.startProvisioningRequest(provMap,
null);
        sleep(5);
        //
        //
        T_ProvisioningStatusQueryChoice [] choice = new
T_ProvisioningStatusQueryChoice[3];
        choice[0] = new T_ProvisioningStatusQueryChoice();
        choice[0].setRecipient(recipient);
        choice[1] = new T_ProvisioningStatusQueryChoice();
        choice[1].setRequestId(requestId);
        choice[2] = new T_ProvisioningStatusQueryChoice();
        choice[2].setStatus(new Integer(ProcessConstants.PROCESSING) );
        //
        // Initialize the query
        T_ProvisioningStatusQuery query = new
T_ProvisioningStatusQuery(T_Logic.AND,
T_ProvisioningStatusOrder.STATUS, choice);
        //
        // Make the query
        StringBuffer sb = new StringBuffer();
        int maxRecords = -1;

        ProvisioningStatusArray provStatusArray =
stub.getProvisioningStatuses(query, maxRecords);

```

startWithDigitalSignature

Used to start a workflow and specify that a digital signature is required.

Method Signature

```

java.lang.String startWithDigitalSignature(java.lang.String processId,
java.lang.String recipient, com.novell.soa.af.impl.soap.DataItemArray
items, java.lang.String digitalSignature,
com.novell.soa.af.impl.soap.SignaturePropertyArray
digitalSignaturePropertyArray)

```

Parameters

Parameter	Description
processId	The request identifier.
recipient	The request recipient.
items	The data items for the provisioning request.
digital signature	The digital signature.
digitalSignaturePropertyArray.	The digital signature property map.

Example

```
String recipient =
ServiceUtils.getInstance().getLoginData().getUsername(LoginData.RECIPI
ENT_TYPE);
//
// Get the digital signature string for admin
String digitalSignature =
DigitalSignatureUtils.getDigitalSignatureFromFile(IDigitalSignatureCon
stants.ADMIN_DIGITAL_SIGNATURE_FILENAME);

ProvisioningRequestArray requestArray =
stub.getAllProvisioningRequests(recipient);
//
// If there are some then,
if(requestArray != null)
{
    String Id = " ";
    String requestId = " ";
    String requestNameToStart = "Enable Active Directory Account
(Mgr Approve-No Timeout)";
    //
    // Loop thru and find the request that we want to start
    ProvisioningRequest [] requests =
requestArray.getProvisioningrequest();
    for(int index = 0; index < requests.length; index++)
    {
        //
        // Is this the name of the request to start?
        if(requests[index].getName().compareTo(requestNameToStart)
== 0)
        {
            //
            // Get the current associated data items. Replicate a
new
            // dataitem array excluding the null values.
            Id = requests[index].getId();
            DataItem [] dataItem =
requests[index].getItems().getDataitem();
            if(dataItem != null)
            {
                // Call method replicateDataItemArray on the
                // provUtils utility object, which refers to a
                // utility class that does not ship with the
                // Identity Manager User Application.
                DataItemArray newDataItemArray =
provUtils.replicateDataItemArray(dataItem);
                //
```

```

        // Start a digitally signed provisioning resource
for the recipient

        requestId =
stub.startWithDigitalSignature(request.getId(), recipient,
newDataItemArray, digitalSignature, null); // Don't get any property
values (optional)
    }
    }
    }
}

```

startAsProxyWithDigitalSignature

Used to start a workflow using a proxy for the initiator, and specify that a digital signature is required.

Method Signature

```

java.lang.String startAsProxyWithDigitalSignature(java.lang.String
processId, java.lang.String recipient,
com.novell.soa.af.impl.soap.DataItemArray items, java.lang.String
digitalSignature, com.novell.soa.af.impl.soap.SignaturePropertyArray
digitalSignaturePropertyArray, java.lang.String proxyUser)

```

Parameters

Parameter	Description
processId	The request identifier.
recipient	The request recipient.
items	The data items for the provisioning request.
digital signature	The digital signature.
digitalSignaturePropertyArray.	The digital signature property map.
proxyUser	The DN of the proxy user.

Example

```

//
// Get the digital signature string for admin
String digitalSignature =
DigitalSignatureUtils.getDigitalSignatureFromFile(IDigitalSignatureCon
stants.ADMIN_DIGITAL_SIGNATURE_FILENAME);

ProvisioningRequestArray requestArray =
stub.getAllProvisioningRequests(recipient);
//
// If there are some then,
if(requestArray != null)

```

```

    {
        String Id = " ";
        String requestId = " ";
        String requestNameToStart = "Enable Active Directory Account
(Mgr Approve-No Timeout)";
        //
        // Loop thru and find the request that we want to start
        ProvisioningRequest [] requests =
requestArray.getProvisioningrequest();
        for(int index = 0; index < requests.length; index++)
        {
            //
            // Is this the name of the request to start?
            if(requests[index].getName().compareTo(requestNameToStart)
== 0)
            {
                //
                // Get the current associated data items. Replicate a
new
                // dataitem array excluding the null values.
                Id = requests[index].getId();
                DataItem [] dataItem =
requests[index].getItems().getDataitem();
                if(dataItem != null)
                {
                    // Call method replicateDataItemArray on the
                    // provUtils utility object, which refers to a
                    // utility class that does not ship with the
                    // Identity Manager User Application.
                    DataItemArray newDataItemArray =
provUtils.replicateDataItemArray(dataItem);
                    //
                    // Start a digitally signed provisioning resource
as proxy for the recipient

                    requestId =
stub.startAsProxyWithDigitalSignature(request.getId(), recipient,
newDataItemArray, digitalSignature, null, proxyUser);
                }
            }
        }
    }
}

```

C.3.3 Work Entries

This section provides reference information for each Work Entries method. The Work Entries methods include:

- ♦ “forward” on page 434
- ♦ “reassignWorkTask” on page 436
- ♦ “getWork” on page 438
- ♦ “forwardWithDigitalSignature” on page 439
- ♦ “forwardAsProxy” on page 441
- ♦ “unclaim” on page 443
- ♦ “forwardAsProxyWithDigitalSignature” on page 444
- ♦ “reassign” on page 447
- ♦ “getWorkEntries” on page 448
- ♦ “getQuorumForWorkTask” on page 450
- ♦ “resetPriorityForWorkTask” on page 451

forward

Used to forward a task to the next activity in the workflow with the appropriate action (approve, deny, refuse).

Method Signature

```
void forward(java.lang.String wid,  
com.novell.soa.af.impl.soap.T_Action action,  
com.novell.soa.af.impl.soap.DataItemArray items, java.lang.String  
comment)
```

Parameters

Parameter	Description
wid	The work Id.
action	The action to take (approve, deny, refuse).
items	The data items required by the workflow.
comment	The comment.

Example

```
//  
// Initialize and start a provisioning request  
HashMap provMap = new HashMap();  
provMap.put(Helper.RECIPIENT, recipient);  
provMap.put(I"Provisioning_Request_To_Start_Key", "Enable Active  
Directory Account (Mgr Approve-No Timeout)");  
//  
// Start request
```

```

        // Calls method startProvisioningRequest on the provUtils
        // utility object which refers to a utility class that does not
        // ship with the Identity Manager User Application.
        String requestId = provUtils.startProvisioningRequest(provMap,
null);
        sleep(5);
        //
        // Get the process id for this running process
        Process process = stub.getProcess(requestId);
        String processId = null;
        if(process != null)
            processId = process.getProcessId();

        T_Action action = T_Action.APPROVE;

        T_Logic logic = T_Logic.AND;

        T_WorkEntryOrder workEntryOrder = T_WorkEntryOrder.REQUEST_ID;

        T_WorkEntryQueryChoice [] workEntryqueryChoice = new
T_WorkEntryQueryChoice[3];
        workEntryqueryChoice[0] = new T_WorkEntryQueryChoice();
        workEntryqueryChoice[0].setRecipient(recipient);
        workEntryqueryChoice[1] = new T_WorkEntryQueryChoice();
        workEntryqueryChoice[1].setRequestId(requestId);
        workEntryqueryChoice[2] = new T_WorkEntryQueryChoice();
        workEntryqueryChoice[2].setProcessId(processId);
        //
        // Create work entry query
        T_WorkEntryQuery query = new T_WorkEntryQuery(logic,
_workEntryOrder, workEntryqueryChoice);
        //
        // Get all work entries (max records)
        WorkEntryArray workEntryArray = stub.getWorkEntries(query, -1);

        WorkEntry [] workEntry = workEntryArray.getWorkentry();

        if(workEntry != null

        {
            for(int wIndex = 0; wIndex < workEntry.length; wIndex++)
            {
                String workId = workEntry[wIndex].getId();
                //

```

```

        //
        LoggerUtils.sendToLogAndConsole("Forwarding : " +
workEntry[wIndex].getActivityName() + " work id: " + workId);
        //
        // Get the dataitem for this item of work
        DataItemArray dataItemArray = stub.getWork(workId);
        DataItem [] dataItem = dataItemArray.getDataitem();
        DataItemArray newDataItemArray = null;
        if(dataItem != null)
            // Call method replicateDataItemArray on the
            // provUtils utility object, which refers to a
            // utility class that does not ship with the
            // Identity Manager User Application.
            newDataItemArray =
provUtils.replicateDataItemArray(dataItem);
        else
            throw new TestProgramException("DataItem is null.");
        //
        // Claim request for recipient
        String comment = _action.toString() + " this request: " +
requestId + " for " + recipient;
        stub.forward(workId, _action, newDataItemArray, comment);
    }

}

```

reassignWorkTask

Used to reassign a task from one user to another.

Method Signature

```
void reassignWorkTask(java.lang.String wid, java.lang.String
addressee, java.lang.String comment)
```

Parameters

Parameter	Description
wid	The Id of the task.
addressee	The addressee of the task.
comment	A comment about the task.

Example

```

//
// Initialize and start a provisioning request
HashMap provMap = new HashMap();
provMap.put(Helper.RECIPIENT, recipient);

```



```

        provMap.put(I"Provisioning_Request_To_Start_Key", "Enable
Active Directory Account (Mgr Approve-No Timeout)");
        //
        // Start request
        // Calls method startProvisioningRequest on the provUtils
        // utility object which refers to a utility class that does not
        // ship with the Identity Manager User Application.
        String requestId = provUtils.startProvisioningRequest(provMap,
null);
        sleep(5);
        //
        // Get the process id for this running process
        Process process = stub.getProcess(requestId);
        if(process != null)
        {
            String processId = process.getProcessId();
            String initiator = process.getInitiator();
            //
            // Setup for the query
            HashMap map = new HashMap();
            map.put(Helper.REQUESTID, requestId);
            map.put(Helper.RECIPIENT, recipient);
            map.put(Helper.PROCESSID, processId);
            map.put(Helper.INITIATOR, initiator);
            WorkEntry [] workEntry =
workEntryUtils.getWorkEntriesUsingQuery(map,
T_WorkEntryOrder.REQUEST_ID, T_Logical.AND);

            if(workEntry == null)
                throw new TestProgramException("Work list is empty.");
            //
            // Reassign the work entry from recipient to the addressee
            //
            // Should only be one item
            String reassignComment = null;
            String workId = workEntry[0].getId();
            if(workId != null)
            {
                //
                // Reassign work entry(s) to addressee
                reassignComment = "Reassigning work entry " + workId +
" from " + recipient + " to " + addressee;
                stub.reassign(workId, addressee, reassignComment);
                LoggerUtils.sendToLogAndConsole("Reassign work entry "
+ workId + " from " + recipient + " to " + addressee);
            }
        }
    }

```

getWork

Used to retrieve data items for a work entry identified by the Id (UUID) of a task.

Method Signature

```
com.novell.soa.af.impl.soap.DataItemArray getWork(java.lang.String
workId)
```

Example

```
//
// Initialize and start a provisioning request
HashMap provMap = new HashMap();
provMap.put(Helper.RECIPIENT, recipient);
provMap.put(I"Provisioning_Request_To_Start_Key", "Enable
Active Directory Account (Mgr Approve-No Timeout)");
//
// Start request
// Calls method startProvisioningRequest on the provUtils
// utility object which refers to a utility class that does not
// ship with the Identity Manager User Application.
String requestId = provUtils.startProvisioningRequest(provMap,
null);
sleep(5);
//
// Get the process id for this running process
Process process = stub.getProcess(requestId);
if(process != null)
{
    String processId = process.getProcessId();
    String initiator = process.getInitiator();
    //
    // Setup for the query
    HashMap map = new HashMap();
    map.put(Helper.REQUESTID, requestId);
    map.put(Helper.RECIPIENT, recipient);
    map.put(Helper.PROCESSID, processId);
    map.put(Helper.INITIATOR, initiator);
    WorkEntry [] workEntry =
workEntryUtils.getWorkEntriesUsingQuery(map,
T_WorkEntryOrder.REQUEST_ID, T_Logic.AND);
    //
    // Do assertion here
    Assert.assertNotNull("WorkEntry is null for recipient : " +
recipient + " with request id : " + requestId, workEntry);
    DataItemArray dataItemArray =
stub.getWork(workEntry[0].getId() );
    DataItem [] dataItem = dataItemArray.getDataitem();
    if(dataItem != null)
```

```

        LoggerUtils.sendToLogAndConsole(dataItem[0].getName());
    }

```

forwardWithDigitalSignature

Used to forward a provisioning request with a digital signature and optional digital signature properties. For example, this can be used by an administrator to force a user-facing activity to be approved, denied or refused.

Method Signature

```

void forwardWithDigitalSignature(java.lang.String wid,
com.novell.soa.af.impl.soap.T_Action action,
com.novell.soa.af.impl.soap.DataItemArray items, java.lang.String
comment, java.lang.String digitalSignature,
com.novell.soa.af.impl.soap.SignaturePropertyArray
digitalSignaturePropertyArray)

```

Parameters

Parameter	Description
wid	The workId.
action	The action to take (approve, deny, refuse).
items	The data items required by the workflow.
comment	A comment about the action.
digitalSignature	The digital signature.
digitalSignaturePropertyArray	The digital signature property map.

Example

```

//
// Initialize and start a provisioning request
HashMap provMap = new HashMap();
provMap.put(Helper.RECIPIENT, recipient);
provMap.put(I"Provisioning_Request_To_Start_Key", "Enable Active
Directory Account (Mgr Approve-No Timeout)");
//
// Start request
// Calls method startProvisioningRequest on the provUtils
// utility object which refers to a utility class that does not
// ship with the Identity Manager User Application.
String requestId = provUtils.startProvisioningRequest(provMap,
null);
sleep(5);
//
// Get the process id for this running process
Process process = stub.getProcess(requestId);
String processId = null;

```

```

        if(process != null)
            processId = process.getProcessId();

        T_Action action = T_Action.APPROVE;

        T_Logic logic = T_Logic.AND;

        T_WorkEntryOrder workEntryOrder = T_WorkEntryOrder.REQUEST_ID;

        // Get the digital signature string for admin
        String digitalSignature =
        DigitalSignatureUtils.getDigitalSignatureFromFile(IDigitalSignatureConstants.ADMIN_DIGITAL_SIGNATURE_FILENAME);

        T_WorkEntryQueryChoice [] workEntryqueryChoice = new
        T_WorkEntryQueryChoice[3];
        workEntryqueryChoice[0] = new T_WorkEntryQueryChoice();
        workEntryqueryChoice[0].setRecipient(recipient);
        workEntryqueryChoice[1] = new T_WorkEntryQueryChoice();
        workEntryqueryChoice[1].setRequestId(requestId);
        workEntryqueryChoice[2] = new T_WorkEntryQueryChoice();
        workEntryqueryChoice[2].setProcessId(processId);
        //
        // Create work entry query
        T_WorkEntryQuery query = new T_WorkEntryQuery(logic,
        _workEntryOrder, workEntryqueryChoice);
        //
        // Get all work entries (max records)
        WorkEntryArray workEntryArray = stub.getWorkEntries(query, -1);

        WorkEntry [] workEntry = workEntryArray.getWorkentry();

        if(workEntry != null

        {
            for(int wIndex = 0; wIndex < workEntry.length; wIndex++)
            {
                String workId = workEntry[wIndex].getId();
                //
                //
                LoggerUtils.sendToLogAndConsole("Forwarding : " +
        workEntry[wIndex].getActivityName() + " work id: " + workId);
                //
                // Get the dataitem for this item of work
                DataItemArray dataItemArray = stub.getWork(workId);
            }
        }
    }

```

```

        DataItem [] dataItem = dataItemArray.getDataitem();
        DataItemArray newDataItemArray = null;
        if(dataItem != null)
            // Call method replicateDataItemArray on the
            // provUtils utility object, which refers to a
            // utility class that does not ship with the
            // Identity Manager User Application.
            newDataItemArray =
provUtils.replicateDataItemArray(dataItem);
        else
            throw new TestProgramException("DataItem is null.");
        //
        // Claim request for recipient
        String comment = _action.toString() + " this request: " +
requestId + " for " + recipient;
        stub.forwardWithDigitalSignature(workId, _action,
newDataItemArray, comment, digitalSignature, null);
    }

}

```

forwardAsProxy

Used to forward a provisioning request. For example, this can be used by an administrator to force a user-facing activity to be approved, denied or refused.

Method Signature

```

void forwardAsProxy(java.lang.String wid,
com.novell.soa.af.impl.soap.T_Action action,
com.novell.soa.af.impl.soap.DataItemArray items, java.lang.String
comment, java.lang.String proxyUser)

```

Parameters

Parameter	Description
wid	The workId (activity Id).
action	The action to take (approve, deny, refuse).
items	The data items required by the workflow.
comment	The comment to add to the activity.
proxyUser	The DN of the proxy user.

Example

```

//
// Initialize and start a provisioning request
HashMap provMap = new HashMap();
provMap.put (Helper.RECIPIENT, recipient);

```

```

        provMap.put(I"Provisioning_Request_To_Start_Key", "Enable Active
Directory Account (Mgr Approve-No Timeout)");
        //
        // Start request
        // Calls method startProvisioningRequest on the provUtils
        // utility object which refers to a utility class that does not
        // ship with the Identity Manager User Application.
        String requestId = provUtils.startProvisioningRequest(provMap,
null);
        sleep(5);
        //
        // Get the process id for this running process
        Process process = stub.getProcess(requestId);
        String processId = null;
        if(process != null)
            processId = process.getProcessId();

T_Action action = T_Action.APPROVE;

T_Logic logic = T_Logic.AND;

T_WorkEntryOrder workEntryOrder = T_WorkEntryOrder.REQUEST_ID;

T_WorkEntryQueryChoice [] workEntryqueryChoice = new
T_WorkEntryQueryChoice[3];
        workEntryqueryChoice[0] = new T_WorkEntryQueryChoice();
        workEntryqueryChoice[0].setRecipient(recipient);
        workEntryqueryChoice[1] = new T_WorkEntryQueryChoice();
        workEntryqueryChoice[1].setRequestId(requestId);
        workEntryqueryChoice[2] = new T_WorkEntryQueryChoice();
        workEntryqueryChoice[2].setProcessId(processId);
        //
        // Create work entry query
        T_WorkEntryQuery query = new T_WorkEntryQuery(logic,
_workEntryOrder, workEntryqueryChoice);
        //
        // Get all work entries (max records)
        WorkEntryArray workEntryArray = stub.getWorkEntries(query, -1);

        WorkEntry [] workEntry = workEntryArray.getWorkentry();

        if(workEntry != null

        {

```

```

        for(int wIndex = 0; wIndex < workEntry.length; wIndex++)
        {
            String workId = workEntry[wIndex].getId();
            //
            //
            LoggerUtils.sendToLogAndConsole("Forwarding : " +
workEntry[wIndex].getActivityName() + " work id: " + workId);
            //
            // Get the dataitem for this item of work
            DataItemArray dataItemArray = stub.getWork(workId);
            DataItem [] dataItem = dataItemArray.getDataitem();
            DataItemArray newDataItemArray = null;
            if(dataItem != null)
                // Call method replicateDataItemArray on the
                // provUtils utility object, which refers to a
                // utility class that does not ship with the
                // Identity Manager User Application.
                newDataItemArray =
provUtils.replicateDataItemArray(dataItem);
            else
                throw new TestProgramException("DataItem is null.");
            //
            // Claim request for recipient
            String comment = _action.toString() + " this request: " +
requestId + " for " + recipient;
            String proxyUser =
ServiceUtils.getInstance().getLoginData().getUsername(LoginData.PROXY_
TYPE);
            stub.forwardAsProxy(workId, _action, newDataItemArray,
comment, proxyUser);
        }
    }

```

unclaim

Used to unclaim a provisioning request. This method only works if the request was claimed in the User Application. You cannot unclaim a request once it has been forwarded using the SOAP interface, because the forward API method (see [“forward” on page 434](#)) claims and forwards in one operation.

Method Signature

```
void unclaim(java.lang.String wid, java.lang.String comment)
```

Parameters

Parameter	Description
workId	The Id of the activity to unclaim.

Parameter	Description
comment	A comment about the action.

Example

```
// Action and Approval Types
final int SELECTED_ACTION = 0; final int CLAIMED_SELECTED_ACTION =
0;

T_Action [] action = {T_Action.APPROVE, T_Action.REFUSE,
T_Action.DENY};
T_ApprovalStatus [] claimedAction = {T_ApprovalStatus.Approved,
T_ApprovalStatus.Retracted, T_ApprovalStatus.Denied};
//
// Get the process id for this running process
Process process = stub.getProcess(requestId);
String processId = null;
if(process != null)
    processId = process.getProcessId();

HashMap map = new HashMap();
map.put(Helper.REQUESTID, requestId);
map.put(Helper.RECIPIENT, recipient);
map.put(Helper.PROCESSID, processId);
//
// Claim the request
WorkEntry workEntry = workEntryUtils.claimWorkEntry(map,
action[SELECTED_ACTION]);
if(workEntry != null)
{
    //
    // Now unclaim the entry
    String workId = workEntry.getId();
    stub.unclaim(workId, "Unclaiming this work item : " + workId +
" for request id : " + requestId);
}
```

forwardAsProxyWithDigitalSignature

Used to forward a provisioning request with a digital signature and digital signature properties. For example, this can be used by an administrator to force a user-facing activity to be approved, denied or refused.

Method Signature

```
void forwardAsProxyWithDigitalSignature(java.lang.String wid,
com.novell.soa.af.impl.soap.T_Action action,
com.novell.soa.af.impl.soap.DataItemArray items, java.lang.String
comment, java.lang.String digitalSignature,
com.novell.soa.af.impl.soap.SignaturePropertyArray
digitalSignaturePropertyArray, java.lang.String proxyUser)
```


Parameters

Parameter	Description
wid	The workId (activity Id).
action	The action to take (approve, deny, refuse).
items	The data items required by the workflow.
comment	The comment to add to the activity.
digitalSignature	The digital signature.
digitalSignaturePropertyArray	The digital signature property map.
proxyUser	The DN of the proxy user.

Example

```
//
// Initialize and start a provisioning request
HashMap provMap = new HashMap();
provMap.put(Helper.RECIPIENT, recipient);
provMap.put(I"Provisioning_Request_To_Start_Key", "Enable Active
Directory Account (Mgr Approve-No Timeout)");
//
// Start request
// Calls method startProvisioningRequest on the provUtils
// utility object which refers to a utility class that does not
// ship with the Identity Manager User Application.
String requestId = provUtils.startProvisioningRequest(provMap,
null);
sleep(5);
//
// Get the process id for this running process
Process process = stub.getProcess(requestId);
String processId = null;
if(process != null)
    processId = process.getProcessId();

T_Action action = T_Action.APPROVE;

T_Logic logic = T_Logic.AND;

T_WorkEntryOrder workEntryOrder = T_WorkEntryOrder.REQUEST_ID;

T_WorkEntryQueryChoice [] workEntryqueryChoice = new
T_WorkEntryQueryChoice[3];
```

```

workEntryqueryChoice[0] = new T_WorkEntryQueryChoice();
workEntryqueryChoice[0].setRecipient(recipient);
workEntryqueryChoice[1] = new T_WorkEntryQueryChoice();
workEntryqueryChoice[1].setRequestId(requestId);
workEntryqueryChoice[2] = new T_WorkEntryQueryChoice();
workEntryqueryChoice[2].setProcessId(processId);
//
// Create work entry query
T_WorkEntryQuery query = new T_WorkEntryQuery(logic,
_workEntryOrder, workEntryqueryChoice);
//
// Get all work entries (max records)
WorkEntryArray workEntryArray = stub.getWorkEntries(query, -1);

WorkEntry [] workEntry = workEntryArray.getWorkentry();

if(workEntry != null

{
    for(int wIndex = 0; wIndex < workEntry.length; wIndex++)
    {
        String workId = workEntry[wIndex].getId();
        //
        //
        LoggerUtils.sendToLogAndConsole("Forwarding : " +
workEntry[wIndex].getActivityName() + " work id: " + workId);
        //
        // Get the dataitem for this item of work
        DataItemArray dataItemArray = stub.getWork(workId);
        DataItem [] dataItem = dataItemArray.getDataitem();
        DataItemArray newDataItemArray = null;
        if(dataItem != null)
            // Call method replicateDataItemArray on the
            // provUtils utility object, which refers to a
            // utility class that does not ship with the
            // Identity Manager User Application.
            newDataItemArray =
provUtils.replicateDataItemArray(dataItem);
        else
            throw new TestProgramException("DataItem is null.");
        //
        // Claim request for recipient
        String comment = _action.toString() + " this request: " +
requestId + " for " + recipient;
        String digitalSignature =
DigitalSignatureUtils.getDigitalSignatureFromFile(IDigitalSignatureCon
stants.MMACKENZIE_DIGITAL_SIGNATURE_FILENAME);

```

```

        String proxyUser =
ServiceUtils.getInstance().getLoginData().getUsername(LoginData.PROXY_
TYPE);

        stub.forwardAsProxyWithDigitalSignature(workId, _action,
newDataItemArray, comment, digitalSignature, null, proxyUser);
    }

}

```

reassign

Used to reassign a task from one user to another.

Method Signature

```

void reassign(java.lang.String wid, java.lang.String addressee,
java.lang.String comment)

```

Parameters

Parameter	Description
wid	The Id of the activity to be reassigned.
addressee	The addressee of the activity.
comment	A comment about the action.

Example

```

//
// Initialize and start a provisioning request
HashMap provMap = new HashMap();
provMap.put(Helper.RECIPIENT, recipient);
provMap.put(I"Provisioning_Request_To_Start_Key", "Enable
Active Directory Account (Mgr Approve-No Timeout)");
//
// Start request
// Calls method startProvisioningRequest on the provUtils
// utility object which refers to a utility class that does not
// ship with the Identity Manager User Application.
String requestId = provUtils.startProvisioningRequest(provMap,
null);
sleep(5);
//
// Get the process id for this running process
Process process = stub.getProcess(requestId);
if(process != null)
{
    String processId = process.getProcessId();
    String initiator = process.getInitiator();
}

```

```

//
// Setup for the query
HashMap map = new HashMap();
map.put(Helper.REQUESTID, requestId);
map.put(Helper.RECIPIENT, recipient);
map.put(Helper.PROCESSID, processId);
map.put(Helper.INITIATOR, initiator);
WorkEntry [] workEntry =
workEntryUtils.getWorkEntriesUsingQuery(map,
T_WorkEntryOrder.REQUEST_ID, T_Logic.AND);

if(workEntry == null)
    throw new TestProgramException("Work list is empty.");
//
// Reassign the work entry from recipient to the addressee
//
// Should only be one work item
String reassignComment = null;
String workId = workEntry[0].getId();
if(workId != null)
{
    //
    // Reassign work entry(s) to addressee
    reassignComment = "Reassigning work entry " + workId +
" from " + recipient + " to " + addressee;
    stub.reassign(workId, addressee, reassignComment);
    LoggerUtils.sendToLogAndConsole("Reassign work entry "
+ workId + " from " + recipient + " to " + addressee);
}
}

```

getWorkEntries

Used to query the work entries (activities) and returns a list of WorkEntry objects that satisfy the query.

Method Signature

```

com.novell.soa.af.impl.soap.WorkEntryArray
getWorkEntries(com.novell.soa.af.impl.soap.T_WorkEntryQuery query, int
maxRecords)

```

Parameters

Parameter	Description
query	<p>Used to specify the query used to retrieve the list of activities. The query has the following components:</p> <ul style="list-style-type: none">♦ choice - the parameters used to filter the results. You can specify multiple parameters. The possible parameters are: Adresse - a DN ProcessId RequestId ActivityId Status (an integer) Owner Priority CreationTime (YYYY/MM/DD) ExpTime (YYYY/MM/DD) CompletionTime (YYYY/MM/DD) Recipient Initiator ProxyFor♦ logic - AND or OR♦ order - the order in which to sort the results. Possible values for order are: ACTIVITY_ID RECIPIENT PROVISIONING_TIME RESULT_TIME STATE STATUS REQUEST_ID MESSAGE
maxRecords	Used to specify maximum number of records to retrieve. A value of -1 returns unlimited records.

Example

```
T_Action action = T_Action.APPROVE;

T_Logic logic = T_Logic.AND;

T_WorkEntryOrder workEntryOrder = T_WorkEntryOrder.REQUEST_ID;

T_WorkEntryQueryChoice [] workEntryqueryChoice = new
T_WorkEntryQueryChoice[3];
```

```

workEntryqueryChoice[0] = new T_WorkEntryQueryChoice();
workEntryqueryChoice[0].setRecipient(recipient);
workEntryqueryChoice[1] = new T_WorkEntryQueryChoice();
workEntryqueryChoice[1].setRequestId(requestId);
workEntryqueryChoice[2] = new T_WorkEntryQueryChoice();
workEntryqueryChoice[2].setProcessId(processId);
//
// Create work entry query
T_WorkEntryQuery query = new T_WorkEntryQuery(logic,
_workEntryOrder, workEntryqueryChoice);
//
// Get all work entries (max records)
WorkEntryArray workEntryArray = stub.getWorkEntries(query, -1);

WorkEntry [] workEntry = workEntryArray.getWorkentry();

```

getQuorumForWorkTask

Used to get information about the quorum for a workflow activity. A quorum must have actually been specified for the workflow activity by the workflow designer for this method to work.

Method Signature

```

com.novell.soa.af.impl.soap.Quorum
getQuorumForWorkTask((java.lang.String workId)

```

Example

```

//

// Note: Provisioning resource must contain a quorum in the flow
for this api method to work

//
// Action and Approval Types
final int SELECTED_ACTION = 0; final int CLAIMED_SELECTED_ACTION =
0;

T_Action [] action = {T_Action.APPROVE, T_Action.REFUSE,
T_Action.DENY};
T_ApprovalStatus [] claimedAction = {T_ApprovalStatus.Approved,
T_ApprovalStatus.Retracted, T_ApprovalStatus.Denied};
//
// Get the process id for this running process
Process process = stub.getProcess(requestId);
String processId = null;
if(process != null)
    processId = process.getProcessId();
//
// Setup for the query
HashMap map = new HashMap();

```

```

        map.put(Helper.REQUESTID, requestId);
        map.put(Helper.RECIPIENT, recipient);
        map.put(Helper.PROCESSID, processId);
        map.put(Helper.INITIATOR, process.getInitiator() );
        WorkEntry [] workEntry =
workEntryUtils.getWorkEntriesUsingQuery(map,
T_WorkEntryOrder.REQUEST_ID, T_Logic.AND);

        Assert.assertNotNull("WorkEntry is null for recipient : " +
recipient + " with request id : " + requestId, workEntry);
        //
        //
        String workId = workEntry[0].getId();

        Quorum quorum = stub.getQuorumForWorkTask(workId);

        Assert.assertNotNull("Quorum for work task is null for recipient :
" + recipient + " with request id : " + requestId, quorum);
        //

        // Extract some data
        int approvalCondition = quorum.getApprovalCondition();
        int status = quorum.getStatus();
        int approveCount = quorum.getApproveCount();
        int participantCount = quorum.getParticipantCount();
        int refuseCount = quorum.getRefuseCount();

```

resetPriorityForWorkTask

Used to reset the priority for a task. You should only use this method on provisioning requests that have a single approval branch.

Method Signature

```
void resetPriorityForWorkTask(java.lang.String workId, int priority,
java.lang.String comment)
```

Parameters

Parameter	Description
workId	The Id of the activity.
priority	The priority to set for the activity.
comment	A comment about the action.

Example

```

// Calls method getProvisioningResourceNameForRecipient
// on the provUtils utility object, which refers to a utility class
// that does not ship with the Identity Manager User Application.

```

```

String requestNameToStart =
provUtils.getProvisioningResourceNameForRecipient(recipient, "Enable
Active Directory Account");
    Map map = MapUtils.createAndSetMap(new Object[] {
        Helper.RECIPIENT, recipient,
        IProvisioningConstants.PROVISIONING_REQUEST_TO_START,
requestNameToStart});
    //
    // Try and start the provisioning request
    String requestId =
provWrapper.startProvisioningRequest(recipient, requestNameToStart);
    RationalTestScript.sleep(5);
    //
    // Get the process id for this running process
    Process process = stub.getProcess(requestId);
    if(process != null)
    {
        //
        // Setup for the query
        HashMap map = new HashMap();
        map.put(Helper.REQUESTID, requestId);
        map.put(Helper.RECIPIENT, recipient);
        map.put(Helper.PROCESSID, process.getProcessId());
        map.put(Helper.INITIATOR, process.getInitiator());
        WorkEntry [] workEntry =
workEntryUtils.getWorkEntriesUsingQuery(map,
T_WorkEntryOrder.REQUEST_ID, T_Logic.AND);
        //
        // Now reset the priority for this work item.
        String workId = workEntry[0].getId();
        String comment = "Resetting priority for this work item.";
        int priority = 0;
        stub.resetPriorityForWorkTask(workId, priority, comment);
    }
}

```

C.3.4 Comments

This section provides reference information for each Comments method. The Comments methods include:

- ◆ [“getCommentsByType” on page 453](#)
- ◆ [“getCommentsByActivity” on page 454](#)
- ◆ [“getCommentsByUser” on page 455](#)
- ◆ [“getCommentsByCreationTime” on page 456](#)
- ◆ [“addComment” on page 457](#)
- ◆ [“getComments” on page 458](#)

getCommentsByType

Used to get workflow comments that are of a specific type (for example, user, system).

Method Signature

```
com.novell.soa.af.impl.soap.CommentArray  
getCommentsByType(java.lang.String requestId,  
com.novell.soa.af.impl.soap.T_CommentType type)
```

Parameters

Parameter	Description
requestId	The process identifier.
type	The comment type (USER or SYSTEM)

Example

```
//  
// Initialize and start a provisioning request  
HashMap provMap = new HashMap();  
provMap.put(Helper.RECIPIENT, recipient);  
provMap.put("Provisioning_Request_To_Start_Key", "Enable  
Active Directory Account (Mgr Approve-No Timeout)");  
//  
// Start request  
// Calls method startProvisioningRequest on the provUtils  
// utility object which refers to a utility class that does not  
// ship with the Identity Manager User Application.  
String requestId = provUtils.startProvisioningRequest(provMap,  
null);  
sleep(5);  
//  
// Get the comments by type : either User or System  
T_CommentType [] commentTypes = {T_CommentType.User,  
T_CommentType.System};  
  
for(int types = 0; types < commentTypes.length; types++)  
{  
    CommentArray commentArray = stub.getCommentsByType(requestId,  
commentTypes[types]);  
    Comment [] comments = commentArray.getComment();  
    if(comments != null)  
    {  
        for(int index = 0; index < comments.length; index++)  
        {  
            LoggerUtils.sendToLogAndConsole(" \nComment Type = " +  
commentTypes[types].getValue() + "\n" +
```

```

        "Activity Id: " +
comments[index].getActivityId() + "\n" +
        "Comment : " + comments[index].getComment()
+ "\n" +
        "User : " + comments[index].getUser() + "\n"
+
        "System comment : " +
comments[index].getSystemComment() + "\n" +
        "Time stamp : " +
comments[index].getTimestamp().getTime().toString() );
    }
}
}

```

getCommentsByActivity

Used to get the comments for a specific activity.

Method Signature

```

com.novell.soa.af.impl.soap.CommentArray
getCommentsByActivity(java.lang.String requestId, java.lang.String
aid)

```

Parameters

Parameter	Description
requestId	The process identifier.
aid	The activity identifier.

Example

```

//
// Initialize and start a provisioning request
HashMap provMap = new HashMap();
provMap.put(Helper.RECIPIENT, recipient);
provMap.put("Provisioning_Request_To_Start_Key", "Enable
Active Directory Account (Mgr Approve-No Timeout)");
//
// Start request
// Calls method startProvisioningRequest on the provUtils
// utility object which refers to a utility class that does not
// ship with the Identity Manager User Application.
String requestId = provUtils.startProvisioningRequest(provMap,
null);
sleep(5);
//
// Get the process id for this running process
Process process = stub.getProcess(requestId);

```

```

if(process != null)
{
    String processId = process.getProcessId();
    String initiator = process.getInitiator();
    //
    // Setup for the query
    HashMap map = new HashMap();
    map.put(Helper.REQUESTID, requestId);
    map.put(Helper.RECIPIENT, recipient);
    map.put(Helper.PROCESSID, processId);
    map.put(Helper.INITIATOR, initiator);
    WorkEntry [] workEntry =
workEntryUtils.getWorkEntriesUsingQuery(map,
T_WorkEntryOrder.REQUEST_ID, T_Logic.AND);
    //
    // Get the activity id associated with the item of work
    String activityId = workEntry[0].getActivityId();
    //
    // Get the comments based on activity
    if(activityId != null)
    {
        CommentArray commentArray =
stub.getCommentsByActivity(requestId, activityId);
        Comment [] comments = commentArray.getComment();
    }
}

```

getCommentsByUser

Used to get the comments made by a specific user.

Method Signature

```

com.novell.soa.af.impl.soap.CommentArray
getCommentsByUser(java.lang.String requestId, java.lang.String user)

```

Parameters

Parameter	Description
requestId	The process identifier.
user	The the DN of the user (recipient) who created the comments

Example

```

//
// Initialize and start a provisioning request
HashMap provMap = new HashMap();

```

```

        provMap.put(Helper.RECIPIENT, recipient);
        provMap.put(I"Provisioning_Request_To_Start_Key", "Enable
Active Directory Account (Mgr Approve-No Timeout)");
        //
        // Start request
        // Calls method startProvisioningRequest on the provUtils
        // utility object which refers to a utility class that does not
        // ship with the Identity Manager User Application.
        String requestId = provUtils.startProvisioningRequest(provMap,
null);
        sleep(5);
        //
        // Get the comments by recipient (should be the same as user)
        CommentArray commentArray = stub.getCommentsByUser(requestId,
recipient);
        Comment [] comments = commentArray.getComment();

```

getCommentsByCreationTime

Used to get comments made at a specific time.

Method Signature

```

com.novell.soa.af.impl.soap.CommentArray
getCommentsByCreationTime(java.lang.String requestId, long time,
com.novell.soa.af.impl.soap.T_Operator op)

```

Parameters

Parameter	Description
requestId	The process identifier.
time	The time stamp.
op	The query operator to use. Possible values for operator are: EQ - equals LT - less than LE - less than or equal to GT - greater than GE - greater than or equal to

Example

```

//
// Initialize and start a provisioning request
HashMap provMap = new HashMap();
provMap.put(Helper.RECIPIENT, recipient);
provMap.put(I"Provisioning_Request_To_Start_Key", "Enable
Active Directory Account (Mgr Approve-No Timeout)");
//

```

```

        // Start request
        // Calls method startProvisioningRequest on the provUtils
        // utility object which refers to a utility class that does not
        // ship with the Identity Manager User Application.
        String requestId = provUtils.startProvisioningRequest(provMap,
null);
        sleep(5);
        //
        // Get comments by creation time for the provisioning request
        started above.
        long currentTime = System.currentTimeMillis();
        LoggerUtils.sendToLogAndConsole("-->Current date = " + new
java.util.Date(currentTime).toString() );
        //
        //
        T_Operator operator = T_Operator.GT;
        CommentArray commentArray =
stub.getCommentsByCreationTime(requestId, currentTime, operator);
        Comment [] comments = commentArray.getComment();

```

addComment

Used to add a comment to a workflow activity.

Method Signature

```
void addComment(java.lang.String workId, java.lang.String comment)
```

Parameters

Parameter	Description
workId	The activity identifier (UUID).
comment	A comment about the activity.

Example

```

//
// Initialize and start a provisioning request
HashMap provMap = new HashMap();
provMap.put(Helper.RECIPIENT, recipient);
provMap.put(I"Provisioning_Request_To_Start_Key", "Enable
Active Directory Account (Mgr Approve-No Timeout)");
//
// Start request
// Calls method startProvisioningRequest on the provUtils
// utility object which refers to a utility class that does not
// ship with the Identity Manager User Application.

```

```

        String requestId = provUtils.startProvisioningRequest(provMap,
null);
        sleep(5);
        //
        // Setup for the query
        HashMap map = new HashMap();
        map.put(Helper.REQUESTID, requestId);
        map.put(Helper.RECIPIENT, recipient);
        WorkEntry [] workEntry =
workEntryUtils.getWorkEntriesUsingQuery(map,
T_WorkEntryOrder.REQUEST_ID, T_Logic.AND);
        //
        // Add comment to the work entry
        String workId = workEntry[0].getId();
        String processId = workEntry[0].getProcessId();
        String addComment = "Test comment for work id " + workId;
        stub.addComment(workId, addComment);
        sleep(2);

```

getComments

Used to get comments from a workflow.

Method Signature

```
com.novell.soa.af.impl.soap.CommentArray getComments(java.lang.String
workId, int maxRecords)
```

Parameters

Parameter	Description
workId	The activity Id (UUID).
maxRecords	An integer specifying the maximum number of records to retrieve.

Example

```

//
// Setup for the query
HashMap map = new HashMap();
map.put(Helper.RECIPIENT, addressee);
WorkEntry [] workEntry =
workEntryUtils.getWorkEntriesUsingQuery(map,
T_WorkEntryOrder.ADDRESSEE, T_Logic.OR);
//
// Get all the comment records for this workId
int maxRecords = -1;
CommentArray commentArray = stub.getComments(workId, maxRecords);
Comment [] comment = commentArray.getComment();

```

C.3.5 Configuration

This section provides reference information for each Configuration method. The Configuration methods include:

- ♦ “setCompletedProcessTimeout” on page 459
- ♦ “setEngineConfiguration” on page 459
- ♦ “getCompletedProcessTimeout” on page 460
- ♦ “setEmailNotifications” on page 460
- ♦ “clearNIMCaches” on page 461
- ♦ “setWebServiceActivityTimeout” on page 461
- ♦ “getUserActivityTimeout” on page 461
- ♦ “getEmailNotifications” on page 462
- ♦ “setUserActivityTimeout” on page 462
- ♦ “getEngineConfiguration” on page 462
- ♦ “getWebServiceActivityTimeout” on page 462

setCompletedProcessTimeout

Used to set the timeout for completed processes. Processes that were completed more than timeout days ago are removed from the system. The default value is 120 days. The valid range is 0 days to 365 days.

Method Signature

```
void setCompletedProcessTimeout(int time)
```

Example

```
accessConfigurationSettings (SET_COMPLETED_PROCESS_TIMEOUT, new  
Integer(212) );
```

setEngineConfiguration

Used to set workflow engine configuration parameters.

Method Signature

```
void setEngineConfiguration(com.novell.soa.af.impl.soap.Configuration  
config)
```

Parameters

Parameter	Description
minPoolSize	The mininum thread pool size.
maxnPoolSize	The maximum thread pool size.
initialPoolSize	The initial thread pool size.
keepAliveTime	Thread pool keep live time.

Parameter	Description
pendingInterval	The cluster synchronization time.
cleanupInterval	The interval between purging processes from databases.
retryQueueInterval	The interval between retrying failed processes.
maxShutdownTime	The maximum time to let threads complete work before engine shutdown.
userActivityTimeout	The default user activity timeout.
completedProcessTimeout	The default completed process timeout.
webServiceActivityTimeout	The default Web service activity timeout.
emailNotification	Turns email notification on or off.
processCacheInitialCapacity	The process cache initial capacity.
processCacheMaxCapacity	The process cache maximum capacity.
processCacheLoadFactor	The process cache load factor.
heartbeatInterval	The heartbeat interval.
heartbeatFactor	The heartbeat factor.

Example

```
accessConfigurationSettings (SET_ENGINE_CONFIGURATION, new Integer (313)
);
```

getCompletedProcessTimeout

Used to get the timeout for completed processes.

Method Signature

```
int getCompletedProcessTimeout()
```

Example

```
accessConfigurationSettings (GET_COMPLETED_PROCESS_TIMEOUT, new
Integer (121) );
```

setEmailNotifications

Used to globally enable or disable e-mail notifications.

Method Signature

```
void setEmailNotifications (boolean enable)
```


Parameters

Parameter	Description
enable	E-mail notifications are enabled if true; otherwise they are disabled.

Example

```
accessConfigurationSettings (SET_EMAIL_NOTIFICATIONS, new  
Boolean(false) );
```

clearNIMCaches

Clear the Novell Integration Manager (previously named exteNd Composer) caches.

Method Signature

```
void clearNIMCaches()
```

Example

```
accessConfigurationSettings (CLEAR_NIM_CACHES, new Object() );
```

setWebServiceActivityTimeout

Used to set the timeout for Web service activities. The default value is 50 minutes. The valid range is 1 minute to 7 days.

Method Signature

```
void setWebServiceActivityTimeout(int time)
```

Parameters

Parameter	Description
time	The timeout value in minutes.

Example

```
accessConfigurationSettings (SET_WEBSERVICE_ACTIVITY_TIMEOUT, new  
Integer(767) );
```

getUserActivityTimeout

Used to get the timeout for user-facing activities.

Method Signature

```
int getUserActivityTimeout()
```

Example

```
accessConfigurationSettings (GET_USER_ACTIVITY_TIMEOUT, new  
Integer(3767) );
```

getEmailNotifications

Used to determine if global e-mail notifications are enabled or disabled.

Method Signature

```
boolean getEmailNotifications()
```

Example

```
accessConfigurationSettings(GET_EMAIL_NOTIFICATIONS, new Boolean(true)
);
```

setUserActivityTimeout

Used to set the timeout for user-facing activities. The default value is no timeout (a value of zero). The valid range is 1 hour to 365 days.

Method Signature

```
void setUserActivityTimeout(int time)
```

Parameters

Parameter	Description
time	The timeout value in hours.

Example

```
accessConfigurationSettings(SET_USER_ACTIVITY_TIMEOUT, new
Integer(1767) );
```

getEngineConfiguration

Used to get the workflow engine configuration parameters.

Method Signature

```
com.novell.soa.af.impl.soap.Configuration getEngineConfiguration()
```

Example

```
accessConfigurationSettings(GET_ENGINE_CONFIGURATION, new Integer(141)
);
```

getWebServiceActivityTimeout

Used to get the timeout for Web service activities.

Method Signature

```
int getWebServiceActivityTimeout()
```

Example

```
accessConfigurationSettings(GET_WEBSERVICE_ACTIVITY_TIMEOUT, new
Integer(808) );
```

C.3.6 Miscellaneous

This section provides reference information for each Miscellaneous method. The Miscellaneous methods include:

- ♦ “getGraph” on page 463
- ♦ “getFlowDefinition” on page 464
- ♦ “getFormDefinition” on page 465
- ♦ “getVersion” on page 466

getGraph

Used to get a JPG image of the workflow. The Graphviz program must be installed on the computer where the application server and the IDM User Application is running. For more information about Graphviz, see [Graphviz \(http://www.graphviz.org\)](http://www.graphviz.org).

Method Signature

```
byte[] getGraph(java.lang.String processId)
```

Parameters

Parameters	Description
processId	The request Id.

Example

```
//  
    // Initialize and start a provisioning request  
    HashMap provMap = new HashMap();  
    provMap.put(Helper.RECIPIENT, recipient);  
    provMap.put(I"Provisioning_Request_To_Start_Key", "Enable Active  
Directory Account (Mgr Approve-No Timeout)");  
    //  
    // Start request  
    // Calls method startProvisioningRequest on the provUtils  
    // utility object which refers to a utility class that does not  
    // ship with the Identity Manager User Application.  
    String requestId = provUtils.startProvisioningRequest(provMap,  
null);  
    sleep(5);  
    //  
  
    //  
  
    Process process = stub.getProcess(requestId);  
    if(process != null)  
    {  
        byte [] graph = null;
```

```

        if( (recipient.compareTo(process.getRecipient()) == 0) &&
(requestId.compareTo(process.getRequestId()) == 0) )
        {
            graph = stub.getGraph(process.getProcessId() );
        }
        //
        // Do assert
        Assert.assertNotNull("Graph is null.", graph);
    }

```

getFlowDefinition

Used to get the XML for a provisioning request.

Method Signature

```
java.lang.String getFlowDefinition(java.lang.String processId)
```

Parameters

Parameters	Description
processId	The request Id.

Example

```

//
// Initialize and start a provisioning request
HashMap provMap = new HashMap();
provMap.put(Helper.RECIPIENT, recipient);
provMap.put(I"Provisioning_Request_To_Start_Key", "Enable Active
Directory Account (Mgr Approve-No Timeout)");
//
// Start request
// Calls method startProvisioningRequest on the provUtils
// utility object which refers to a utility class that does not
// ship with the Identity Manager User Application.
String requestId= provUtils.startProvisioningRequest(provMap,
null);
sleep(5);
//

//

Process process = stub.getProcess(requestId);
if(process != null)
{
    String XMLFlowDefinition = null;
    if( (recipient.compareTo(process.getRecipient()) == 0) &&
(requestId.compareTo(process.getRequestId()) == 0) )

```

```

        {
            XMLFlowDefinition =
stub.getFlowDefinition(process.getProcessId() );
        }
        //
        // Do assert
        Assert.assertNotNull("Flow Definition is null.",
XMLFlowDefinition);
    }

```

getFormDefinition

Used to get the XML for a form for a provisioning request.

Method Signature

```
java.lang.String getFormDefinition(java.lang.String processId)
```

Parameters

Parameters	Description
processId	The request Id.

Example

```

//
// Initialize and start a provisioning request
HashMap provMap = new HashMap();
provMap.put(Helper.RECIPIENT, recipient);
provMap.put(I"Provisioning_Request_To_Start_Key", "Enable Active
Directory Account (Mgr Approve-No Timeout)");
//
// Start request
// Calls method startProvisioningRequest on the provUtils
// utility object which refers to a utility class that does not
// ship with the Identity Manager User Application.
String requestId = provUtils.startProvisioningRequest(provMap,
null);
sleep(5);
//

//

Process process = stub.getProcess(requestId);
if(process != null)
{

    String XMLFormDefinition = null;

```

```

        if( (recipient.compareTo(process.getRecipient()) == 0) &&
(requestId.compareTo(process.getRequestId()) == 0) )
        {
            XMLFormDefinition =
stub.getFormDefinition(process.getProcessId() );
        }
        //
        // Do assert
        Assert.assertNotNull("Form Definition is null.",
XMLFormDefinition);
    }

```

getVersion

Used to get the version of the workflow system.

Method Signature

```
com.novell.soa.af.impl.soap.T_Version getVersion()
```

Example

```

StringBuffer result = new StringBuffer();

    T_Version version = stub.getVersion();
    if (version != null)
    {
        result.append(" Major = " + version.getMajor() );
        result.append(" Minor = " + version.getMinor() );
        result.append(" Revision = " + version.getRevision() );

        System.out.println("Version Information " + result.toString()
);
    }

```

C.3.7 Cluster

This section provides reference information for each Cluster method. The Cluster methods include:

- ♦ [“getEngineState” on page 466](#)
- ♦ [“reassignAllProcesses” on page 467](#)
- ♦ [“getEngineState” on page 468](#)
- ♦ [“reassignPercentageProcesses” on page 468](#)
- ♦ [“reassignProcesses” on page 469](#)
- ♦ [“removeEngine” on page 469](#)

getEngineState

Used to get the IEngineState for a workflow engine, specified by engine Id.

Method Signature

```
com.novell.soa.af.impl.soap.EngineState  
getEngineState(java.lang.String engineId)
```

Parameters

Parameter	Description
engineId	The Id of the workflow engine.

Example

```
EngineStateArray engineStateArray = stub.getClusterState();  
    EngineState [] engineState = engineStateArray.getEngineStates();  
    if(engineState != null)  
    {  
        LoggerUtils.sendToLogAndConsole("EngineCount in cluster:" +  
engineState.length);  
        for(int index = 0; index < engineState.length; index++)  
        {  
            EngineState engine =  
stub.getEngineState(engineState[index].getEngineId() );  
            LoggerUtils.sendToLogAndConsole(  
                "Engine Id: " + engine.getEngineId() + "\n" +  
                "Engine status: " + engine.getEngineStatus() + "\n" +  
                "Value of engine status: " +  
engine.getValueOfEngineStatus() + "\n" +  
                "Heartbeat: " + ( (engine.getHeartbeat() != null) ?  
engine.getHeartbeat().getTime().toString() : "null") + "\n" +  
                "Shutdown time: " + ((engine.getShutdownTime() != null)  
? engine.getShutdownTime().getTime().toString() : "null") + "\n" +  
                "Start time: " + ((engine.getStartTime() != null) ?  
engine.getStartTime().getTime().toString() : "null") );  
        }  
    }
```

reassignAllProcesses

Used to reassign all processes from the source engine to a list of target engines.

Method Signature

```
int reassignAllProcesses(java.lang.String sourceEngineId,  
com.novell.soa.af.impl.soap.StringArray targetEngineIds)
```

Parameters

Parameter	Description
sourceEngineId	The Id of the source workflow engine.

Parameter	Description
targetEngineIds	The Ids of the target workflow engines.

getEngineState

Used to get a list that contains an IEngineState object for each engine in the cluster.

Method Signature

```
public com.novell.soa.af.impl.soap.EngineState
getEngineState(java.lang.String engineId)
```

Parameters

Parameter	Description
engineId	The Id of the workflow engine.

Example

```
EngineStateArray engineStateArray = stub.getClusterState();
EngineState [] engineState = engineStateArray.getEngineStates();
if(engineState != null)
{
    LoggerUtils.sendToLogAndConsole("EngineCount in cluster:" +
engineState.length);
    for(int index = 0; index < engineState.length; index++)
    {
        EngineState engine =
stub.getEngineState(engineState[index].getEngineId() );
        LoggerUtils.sendToLogAndConsole(
            "Engine Id: " + engine.getEngineId() + "\n" +
            "Engine status: " + engine.getEngineStatus() + "\n" +
            "Value of engine status: " +
engine.getValueOfEngineStatus() + "\n" +
            "Heartbeat: " + ( (engine.getHeartbeat() != null) ?
engine.getHeartbeat().getTime().toString() : "null") + "\n" +
            "Shutdown time: " + ((engine.getShutdownTime() != null)
? engine.getShutdownTime().getTime().toString() : "null") + "\n" +
            "Start time: " + ((engine.getStartTime() != null) ?
engine.getStartTime().getTime().toString() : "null") );
    }
}
```

reassignPercentageProcesses

Used to reassign a percentage of processes from the source engine to the target engine.

Method Signature

```
int reassignPercentageProcesses(int percent, java.lang.String  
sourceEngineId, java.lang.String targetEngineId)
```

Parameters

Parameter	Description
percent	An integer representing the percentage of processes to be reassigned.
sourceEngineId	The Id of the source workflow engine.
targetEngineIds	The Id of the target workflow engine.

reassignProcesses

Used to reassign one or more processes from the source engine to the target engine.

Method Signature

```
int reassignProcesses(com.novell.soa.af.impl.soap.StringArray  
requestIds, java.lang.String sourceEngineId, java.lang.String  
targetEngineId)
```

Parameters

Parameter	Description
requestIds	A list of requestIds of the processes to be reassigned.
sourceEngineId	The Id of the source workflow engine.
targetEngineIds	The Id of the target workflow engine.

removeEngine

Used to remove an engine from the cluster state table. The engine must be in the SHUTDOWN or TIMEDOUT state.

Method Signature

```
void removeEngine(java.lang.String engineId)
```

Parameters

Parameter	Description
engineId	The Id of the workflow engine to be removed.