AUTHORIZED DOCUMENTATION

Administration Guide

Novell®
Designer for Identity Manager

3.5

September 18, 2009

www.novell.com

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 page (http://www.novell.com/info/exports/) for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

Copyright © 2010 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.
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, see the Novell Documentation Web page (http://www.novell.com/documentation).

Novell Trademarks

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

Third-Party Materials

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

Contents

About This Guide 15

1 Installing Designer 17

1.1 Requirements . . . 17
1.1.1 System Requirements . . . 17
1.1.2 Platform Requirements . . . 17
1.2 Installing . . . 17
1.2.1 Installing Designer If Eclipse Is Not Already Installed . . . 18
1.2.2 Installing Designer If Eclipse Is Already Installed on Your System . . . 20
1.3 Upgrading Designer 2.1x To Designer 3.x . . . 21
1.4 Using the Silent Install . . . 21
1.5 Uninstalling Designer . . . 21

2 Creating a Project 23

2.1 When No Project Exists . . . 23
2.2 When You Want to Create an Additional Project . . . 24
2.3 When You Want to Import a Project . . . 26
2.4 When You Want to Disable a Project . . . 26

3 Creating a Model 27

3.1 Basic Tasks . . . 27
3.2 Accessing the Modeler . . . 28
3.3 Selecting a Modeling Mode . . . 29
3.3.1 Developer Mode . . . 30
3.3.2 Architect Mode . . . 31
3.3.3 Dataflow Mode . . . 35
3.3.4 Table Mode . . . 36
3.4 Working from the Palette . . . 36
3.4.1 About the Palette . . . 37
3.4.2 Palette Operations . . . 37
3.4.3 Using Generic Applications . . . 38
3.4.4 Fly-Out Palette . . . 38
3.4.5 Resizing the Palette . . . 39
3.4.6 Docking the Palette . . . 39
3.4.7 Arranging Folders and Applications . . . 40
3.4.8 Changing the Layout . . . 40
3.4.9 Keyboard Support for the Palette . . . 41
3.5 Copying and Pasting . . . 41
3.5.1 Copying Applications . . . 41
3.5.2 Copying a Driver Set . . . 42
3.5.3 Copying an Identity Vault . . . 43
3.5.4 Copying a Domain Group . . . 44
3.5.5 Copying between Editors . . . 44
3.6 Moving Items . . . 45
3.7 In-Line Editing . . . 46
3.8 Tooltips and Toolbar . . . 46
3.9 Organizing by Domain Groups . . . 47

3.9.1 About Domain Groups . . . 47
3.9.2 Key Features . . . 48
3.9.3 Creating a Domain Group . . . 48
3.9.4 Minimizing (Collapsing) Domain Groups . . . 50
3.9.5 Restoring Domain Groups . . . 51
3.9.6 Maximizing Domain Groups . . . 51
3.9.7 Auto-Placement of Neighbors . . . 51
3.9.8 Grouping into a New Domain Group . . . 51
3.9.9 Ungrouping a Domain Group . . . 52
3.9.10 Clearing Contents . . . 53
3.9.11 Changing a Domain Group Icon . . . 53
3.9.12 Keyboard Support for Domain Groups . . . 54
3.10 Connecting Applications . . . 54
3.10.1 Automatic Connections . . . 55
3.10.2 Connection Target Highlights . . . 55
3.10.3 Automatically Creating Objects . . . 56
3.10.4 Auto Redraw . . . 56
3.10.5 Manually Connecting . . . 56
3.10.6 eDir-to-eDir Connections . . . 56
3.10.7 Multiple Driver Connections . . . 57
3.10.8 Straightening Connections . . . 58
3.10.9 Reconnecting . . . 60
3.10.10 Driver Icons . . . 60
3.10.11 Selected Drivers . . . 60
3.10.12 Auto-Layout of Imported Objects . . . 61
3.10.13 Keyboard Support for Connections . . . 61
3.11 Aligning and Laying Out Components . . . 61
3.11.1 Alignment Hints . . . 62
3.11.2 Using Rulers . . . 63
3.11.3 Using a Grid . . . 65
3.11.4 Distributing Applications . . . 66
3.11.5 Auto-Layouts . . . 67
3.11.6 Layouts to Use for Imports . . . 67
3.12 Editing Multiple Objects . . . 68
3.13 Modeling Active Directory Domain Controllers . . . 68
3.13.1 Configuring a Connection . . . 68
3.13.2 Discovering Controllers . . . 69
3.13.3 Information about Domain Controllers . . . 69
3.14 Saving Your Model . . . 70

4 Configuring Objects in Designer 71

4.1 Viewing Object Properties . . . 71
4.1.1 Properties View . . . 71
4.1.2 Properties Dialog Box . . . 72
4.1.3 Operations Relating to Properties . . . 73
4.2 Configuring a Domain Group . . . 73
4.3 Configuring Identity Vaults . . . 74
4.3.1 Configuration Settings . . . 75
4.3.2 Administrator . . . 76
4.3.3 Server List . . . 76
4.3.4 iManager . . . 76
4.3.5 Local Hostname . . . 77
4.4 Configuring Servers . . . 77
4.5 Configuring Driver Sets . . . 78
4.5.1 Driver Set General Options . . . 78
4.5.2 Driver Set Global Configuration Values . . . 80

4.5.3 Java Environment Parameters . . . 80
4.5.4 Driver Set Log Levels . . . 81
4.5.5 Driver Set Named Passwords . . . 82
4.5.6 Driver Set Server List . . . 82
4.5.7 Driver Set Trace . . . 83
4.6 Configuring Libraries . . . 86
4.7 Configuring Drivers . . . 87
4.7.1 Driver General Settings . . . 88
4.7.2 Driver Global Configuration Values . . . 89
4.7.3 Driver Log Level . . . 95
4.7.4 Driver Configuration . . . 96
4.7.5 Driver Manifest . . . 99
4.7.6 Driver Named Passwords . . . 99
4.7.7 Driver iManager Icon . . . 100
4.7.8 Driver Trace Levels . . . 100
4.7.9 Engine Control Values . . . 103
4.7.10 Driver Health Configuration . . . 105
4.7.11 Reciprocal Attributes . . . 113
4.8 Configuring Jobs . . . 116
4.8.1 General . . . 116
4.8.2 Trace . . . 117
4.9 Configuring ID Policy Containers . . . 117
4.10 Configuring ID Policies . . . 118
4.11 Configuring Application Properties . . . 120
4.11.1 General . . . 121
4.11.2 Environment . . . 121
4.11.3 Administrator . . . 121
4.11.4 Connectivity . . . 121
4.11.5 AD Domain . . . 125
4.12 Configuring Policies . . . 125
4.12.1 Editing a Policy Name . . . 125
4.12.2 Viewing References . . . 125
4.12.3 Editing a Credential Provisioning Policy . . . 125
4.13 Adding Prompts to a Driver Configuration File . . . 126
4.14 Editing a Notification Template Name . . . 127
4.15 Synchronizing Passwords . . . 127

5 Managing Identity Manager Versions 129

5.1 Key Differences in Identity Manager Versions . . . 130
5.2 Changing the Identity Manager Version . . . 130
5.3 Tracking Versions of Identity Manager . . . 131
5.4 Information in the Driver Configuration Wizard . . . 133
5.5 Checking Projects for Version Issues . . . 134
5.6 Adjusting the UI Based on the Version Number . . . 135

6 Managing the Schema 137

6.1 Using the Manage Schema Tool . . . 137
6.1.1 The Classes Tab . . . 138
6.1.2 The Attributes Tab . . . 141
6.2 Creating Classes and Attributes . . . 147
6.2.1 Creating Identity Vault Classes . . . 147
6.2.2 Creating Identity Vault Attributes . . . 148
6.3 Modifying the Schema . . . 149
6.3.1 Deleting Schema Definitions . . . 149

6.3.2 Modifying Classes or Attributes . . . 149
6.3.3 Renaming Schema Definitions . . . 150
6.4 Deploying the Schema into the Identity Vault . . . 150
6.5 Exporting the Schema to a File . . . 152
6.5.1 Exporting the Schema to a .sch File . . . 152
6.5.2 Exporting the Schema to an LDIF File . . . 153
6.6 Importing the Schema . . . 155
6.6.1 Importing the Schema from the Identity Vault . . . 155
6.6.2 Importing the Schema from a File . . . 157
6.7 Managing a Copy of an Application Schema . . . 160
6.7.1 Editing an Application's Schema . . . 160
6.7.2 Refreshing the Application Schema . . . 162
6.8 Mapping Identity Vault to an LDAP Schema . . . 162
6.9 Comparing the Schema . . . 163

7 Managing the Flow of Data 167

7.1 The Dataflow View . . . 167
7.1.1 Accessing the Dataflow View . . . 168
7.1.2 Flow Arrows in the Modeler . . . 168
7.1.3 Viewing How Attributes Are Synchronized . . . 170
7.1.4 Changing the Data Flow . . . 170
7.2 The Dataflow Editor . . . 173
7.2.1 Filtering Views . . . 174
7.2.2 Filtering Identity Vaults and Applications . . . 176
7.2.3 Pinning the Identity Vault . . . 176
7.2.4 Expanding and Collapsing the Identity Vault . . . 177
7.2.5 Switching to an eDirectory Tree Icon . . . 179
7.2.6 Viewing an eDir-to-eDir Driver . . . 180
7.2.7 Keyboard Support . . . 180
7.3 Adding Items in the Dataflow Editor . . . 181
7.3.1 Adding an Identity Vault in the Dataflow Editor . . . 181
7.3.2 Adding a Driver in the Dataflow Editor . . . 182
7.3.3 Adding an Application in the Dataflow Editor . . . 182
7.3.4 Adding Classes and Attributes . . . 183
7.3.5 Adding Non-Filter Attributes . . . 183
7.4 Editing Items . . . 185
7.4.1 Editing within the Dataflow Editor . . . 185
7.4.2 Editing Non-Filter Attributes . . . 187
7.4.3 Managing Schema . . . 187
7.4.4 Removing a Flow . . . 187
7.4.5 Changing How Data Flows . . . 188
7.5 Generating HTML Reports . . . 189
7.6 Integrating Passwords . . . 191

8 Creating and Managing Policies 193

9 Enabling Staging of Projects 195

9.1 Prerequisites . . . 196
9.2 Importing Additional Objects from eDirectory . . . 196
9.3 Deploying Additional Objects into eDirectory . . . 201
9.4 Exporting LDIF Container Objects to an LDIF File . . . 205
9.5 Importing Objects from an LDIF File into an LDIF Container . . . 207
9.6 Setting up Driver Security Equivalences . . . 208

9.7 Editing the LDIF Container Data by Using an Editor . . . 213
9.8 Deleting the LDIF Container . . . 213
9.9 Changing the LDAP Properties . . . 214

10 Setting Up E-Mail Notification Templates 215

10.1 Viewing Notification Templates . . . 215
10.2 Editing a Notification Template . . . 218
10.2.1 Selecting a Format . . . 219
10.2.2 Specifying a Subject . . . 219
10.2.3 Working with Tokens . . . 219
10.2.4 Editing a Template Message . . . 220
10.3 Adding and Deploying a Notification Template . . . 221
10.3.1 Adding a Notification Template . . . 221
10.3.2 Importing a Notification Template . . . 222
10.3.3 Deploying a Template . . . 222
10.4 Policy Builder and Notification Templates . . . 223
10.5 Configuring the E-Mail Server . . . 223

11 Importing into Designer 225

11.1 Importing Projects . . . 225
11.1.1 Importing a Project from the Identity Vault . . . 226
11.1.2 Importing a Project from the File System . . . 231
11.1.3 Importing a Project from iManager . . . 233
11.1.4 Importing a Project from a Version Control Server . . . 234
11.2 Importing a Library, a Driver Set, or Driver from the Identity Vault . . . 236
11.2.1 Associating a Server to the Identity Vault . . . 237
11.2.2 Importing a Library from the Identity Vault . . . 238
11.2.3 Importing a Driver Set from the Identity Vault . . . 239
11.2.4 Importing a Driver from the Identity Vault . . . 242
11.3 Importing a Driver Configuration File . . . 245
11.3.1 Importing an Identity Manager Project from the File System . . . 246
11.3.2 Importing a Driver Configuration from a File in the Modeler View . . . 246
11.3.3 Importing from a File through the Outline View . . . 248
11.4 Importing Channels, Policies, and Schema Items from the Identity Vault . . . 250
11.4.1 Importing a Channel . . . 251
11.4.2 Importing a Policy . . . 254
11.4.3 Importing a Schema . . . 256
11.5 Using the Compare Feature When Importing . . . 259
11.5.1 Using Compare When Importing a Driver Object . . . 260
11.5.2 Using Compare on a Channel Object . . . 263
11.5.3 Using Compare On a Policy . . . 263
11.5.4 Matching Attributes with Designer Properties . . . 263
11.6 Error Messages and Solutions . . . 267

12 Documenting Projects 269

12.1 Creating a Document Style . . . 269
12.2 Editing a Document Style for Your Needs . . . 271
12.2.1 Editing a Style Template . . . 271
12.2.2 Editing Sections of a Style . . . 272
12.3 Generating a Document . . . 272
12.4 Using Your Style Template for Other Projects . . . 275
12.4.1 Document a Section of the Style . . . 275

12.4.2 Document Multiple Sections of the Style . . . 278
12.5 Customizing Styles to Include or Exclude Information . . . 278
12.5.1 Identity Vault Schema and Application Schema . . . 278
12.5.2 Using Project Configuration to Limit Information . . . 279
12.6 Advanced Editing of a Document Style . . . 281
12.6.1 What's In the Advanced Editing Mode . . . 281
12.6.2 A Walk-through Tutorial . . . 288
12.6.3 Selecting a Language for Generated Documents . . . 294
12.6.4 Double-Byte Font Support . . . 295

13 Using Entitlements 299

13.1 How Entitlements Work . . . 299
13.2 Designing Entitlements . . . 301
13.2.1 Terminology . . . 302
13.2.2 Entitlement Prerequisites . . . 302
13.2.3 Identity Manager Drivers with Preconfigurations that Support Entitlements . . . 302
13.2.4 Enabling Entitlements On Identity Manager Drivers . . . 303
13.3 Creating Entitlements through the Entitlement Wizard . . . 304
13.3.1 Valueless Entitlements . . . 305
13.3.2 Valued Entitlement that Queries an External Application . . . 307
13.3.3 Administrator-Defined Entitlements with Lists . . . 312
13.3.4 Administrator-Defined Entitlements without Lists . . . 314
13.4 Editing and Viewing Entitlements . . . 316
13.4.1 Entitlement XML Source and XML Tree Views . . . 317
13.4.2 Using the Novell Entitlement DTD . . . 320
13.5 Managing Entitlements . . . 321
13.6 Creating Entitlement Policies . . . 321

14 Scheduling Jobs 327

14.1 Creating a Job . . . 328
14.1.1 Copying a Job . . . 330
14.2 Editing a Job . . . 330
14.2.1 Job Editor Selections on the General Settings Page . . . 331
14.2.2 Job Editor Selections on the Job Parameters Page . . . 334
14.2.3 Job Editor Selections on the Scheduler Page . . . 336
14.2.4 Job Editor Selections on the Notification Settings Page . . . 338

15 Deploying and Exporting 341

15.1 Preparing to Deploy . . . 341
15.2 Deploying a Project to an Identity Vault . . . 342
15.3 Deploying a Driver Set to an Identity Vault . . . 342
15.3.1 eDir-to-eDir Deployments and SSL/TLS . . . 343
15.4 Deploying a Driver to an Identity Vault . . . 344
15.5 Deploying a Channel to an Identity Vault . . . 346
15.6 Deploying a Policy to an Identity Vault . . . 347
15.7 Using the Compare Feature When Deploying . . . 348
15.7.1 Using Compare when Deploying a Driver Object . . . 348
15.7.2 Using Compare Before Deploying a Channel Object . . . 351
15.7.3 Using Compare Before Deploying a Policy . . . 352
15.7.4 Matching Attributes with Designer Properties . . . 352
15.7.5 Comparing Driver Set and Driver Attributes . . . 352
15.7.6 Renaming and Deleting Deployed Objects . . . 353
15.8 Troubleshooting Deployed Objects . . . 354

15.9 Exporting a Project . . . 354
15.10 Exporting To a File . . . 357
15.10.1 The Export Context Menu . . . 357
15.10.2 Exporting Configuration Files from the Modeler View . . . 357
15.10.3 Exporting Configuration Files from the Outline View . . . 358

16 The Novell XML Editor 361

16.1 About the Novell XML Editor . . . 361
16.1.1 Creating XML Files . . . 361
16.1.2 Validating Files . . . 366
16.1.3 Outline View . . . 366
16.1.4 XPath Navigator . . . 366
16.2 Using the Source Editor . . . 366
16.3 Using the Tree Editor . . . 369
16.4 Attaching a Schema or DTD . . . 370
16.5 Setting XML Editor Preferences . . . 372

17 Tools 373

17.1 Converting Earlier Projects . . . 373
17.1.1 Converting Projects With the Project Converter Wizard . . . 373
17.1.2 Running Later Projects on Earlier Designer Versions . . . 376
17.2 Migrating Driver Configuration Data To a New Server . . . 376
17.2.1 Using the Server Migration Wizard To Migrate the Driver Set . . . 376
17.2.2 Migrating a Driver Set To a Server in a Different Tree . . . 378
17.2.3 Migrating Server Data for Each Driver . . . 380
17.3 Opening a Web Browser . . . 382
17.4 Launching iManager . . . 382
17.5 Checking Your Projects . . . 383
17.5.1 Checking a Project . . . 383
17.5.2 Customizing the Project Checker . . . 384
17.5.3 Items That Are Checked . . . 386
17.6 Managing Directory Objects . . . 389
17.6.1 Tool-Based Method . . . 389
17.6.2 Task-Based Method . . . 390
17.6.3 Browse, View, or Modify Object Attributes . . . 391
17.7 Configuring TLS for eDir-to-eDir Drivers . . . 392
17.7.1 Prerequisites . . . 392
17.7.2 Enabling TLS . . . 392
17.7.3 Creating Certificates . . . 394
17.8 Using DS Trace . . . 395
17.8.1 Viewing DS Trace Live . . . 396
17.8.2 Creating a DS Trace Log File . . . 398
17.8.3 Viewing a DS Trace Log File . . . 398
17.9 Working with Generic Resources . . . 399
17.9.1 Creating a Generic Resource Object . . . 399
17.9.2 Editing a Generic Resource Object . . . 401
17.10 Updating Designer . . . 401

18 Editing Icons for Drivers and Applications 403

18.1 Editing Driver Icons . . . 403
18.2 Editing Application Icons . . . 406

19 Version Control **411**

19.1 Installing a Subversion Server on Windows . . . 411
19.1.1 Downloading and Installing the Server . . . 412
19.1.2 Configuring the Server . . . 412
19.2 Checking In a Project To a Version Control Server . . . 413
19.3 Importing a Project from a Version Control Server . . . 417
19.4 Accessing the Version Control View . . . 419
19.4.1 Version Control Icons . . . 420
19.4.2 Version Control View Headings . . . 421
19.4.3 Version Control Options . . . 422
19.5 Comparing Revisions and Resolving Conflicts . . . 428
19.5.1 Comparing Revisions . . . 428
19.5.2 Resolving Conflicts . . . 431
19.5.3 The Modeler View Layout In a Team-Enabled Environment . . . 434
19.5.4 Provisioning Objects . . . 436
19.6 Version Control Best Practices . . . 437
19.6.1 Best Practices . . . 437
19.6.2 Best Practice Scenarios . . . 437
19.6.3 Subversion and Version Control Interaction Rules . . . 441

20 Setting Preferences **443**

20.1 Finding Preference Pages . . . 443
20.2 General . . . 443
20.2.1 Appearance . . . 445
20.2.2 Capabilities . . . 446
20.2.3 Compare/Patch . . . 447
20.2.4 Content Types . . . 447
20.2.5 Editors . . . 448
20.2.6 Keys . . . 450
20.2.7 Network Connections . . . 450
20.2.8 Perspectives . . . 450
20.2.9 Startup and Shutdown . . . 451
20.2.10 Web Browser . . . 451
20.2.11 Welcome . . . 451
20.3 Help . . . 452
20.3.1 Content . . . 452
20.4 Novell . . . 452
20.4.1 Designer . . . 453
20.4.2 Identity Manager . . . 456
20.4.3 Provisioning . . . 474
20.5 Web and XML . . . 474
20.5.1 CSS Files . . . 475
20.5.2 HTML Files . . . 478
20.5.3 JavaScript Files . . . 482
20.5.4 Task Tags . . . 486
20.5.5 XML Catalog . . . 486
20.5.6 XML Files . . . 488

21 Troubleshooting Designer **493**

21.1 Running the Project Checker . . . 493
21.2 Viewing the Error Log . . . 493
21.2.1 Browsing the File System . . . 493
21.2.2 Using Menus . . . 494
21.2.3 Event Details . . . 494

21.2.4 Customizing Filter Settings . . . 495
21.3 Turning On Trace Messages . . . 496
21.4 Checking Loaded Plug-Ins . . . 498
21.5 Deploying Identity Manager Objects . . . 498
21.5.1 Deployment Considerations . . . 499
21.5.2 An Example Deploy Error . . . 500
21.5.3 Error Messages and Solutions . . . 501
21.6 No F1 Help in Maximized Editors . . . 512
21.7 Freeing Heap Memory . . . 512
21.8 Reporting Bugs and Giving Feedback . . . 512

A Modeler Operations 515

A.1 Modeler Space Operations . . . 515
A.2 Identity Vault Operations . . . 516
A.3 Driver Set Operations . . . 519
A.4 Driver Operations . . . 521
A.5 Application Operations . . . 525
A.6 Submenus . . . 528
A.7 Keyboard Support . . . 529

B Document Generator Core Support Templates 531

B.1 dgSection.xsl . . . 531
B.2 dgFormat.xsl . . . 532
B.3 idmConfig.xsl . . . 535
B.4 idmUtil.xsl . . . 536

C Adding Applications and Drivers to the Palette 539

C.1 Definition Folders and Files . . . 540
C.1.1 Driver Configuration and Localization Files . . . 540
C.1.2 Palette Folders and Files . . . 541
C.1.3 The Notification Templates Folder . . . 546
C.1.4 The Themes Folder . . . 546
C.2 Adding to the Palette . . . 546
C.2.1 Copying Configuration Files . . . 547
C.2.2 Creating the Group . . . 547
C.2.3 Adding a Key_Value Pair . . . 548
C.2.4 Creating a Driver Definition . . . 549
C.2.5 Creating the Application . . . 552
C.2.6 Hooking Up the Custom Application . . . 555
C.3 Protecting Your Customized Files . . . 559

D Moving Data from Older Projects 561

D.1 Importing Data from a Live System . . . 561
D.2 Exporting Data from the Old Project To Configuration Files . . . 561
D.2.1 If Multiple Servers Are Associated with a Driver Set . . . 562
D.2.2 Customized E-Mail Templates . . . 562
D.3 Manually Configuring Information That Is Not Imported . . . 562

E Version Control with Subversion and Identity Manager Designer 565

E.1 Understanding Subversion . . . 565
E.1.1 How Revisions Work In Subversion . . . 565
E.1.2 Understanding Atomic Commits . . . 566
E.1.3 Where Subversion Stores the Project Data . . . 567
E.1.4 Moving an Existing Project . . . 567
E.2 Administering Your Subversion Server . . . 568
E.2.1 Server Specifications . . . 568
E.2.2 Network Protocols . . . 569
E.2.3 Authentication Schemes . . . 571
E.2.4 Using Client Certificates . . . 573
E.2.5 Configuring Subversion with Apache HTTP . . . 574
E.2.6 Proxy Server Configuration . . . 574
E.2.7 Subversion Server Backup . . . 575
E.3 Taking Full Advantage of Version Control . . . 576
E.3.1 When to Commit and When to Update . . . 576
E.3.2 Comments . . . 576
E.3.3 Creating and Using Tags . . . 577
E.3.4 Subversion Keyword Substitutions . . . 577
E.4 Glossary . . . 578

About This Guide

Designer for Identity Manager can help you design, test, document, and deploy Identity Manager solutions in a highly productive environment.

Newcomers can use wizards to build Identity Management solutions. Veterans and expert users can bypass the wizards and interact directly at any level of detail.

Audience

Designer for Identity Manager was created for the following audiences:

- Enterprise IT developers
- Consultants
- Sales engineers
- Architects or system designers
- System administrators

Designer is aimed at information technology professionals who:

- Have a strong understanding of directories, databases, and the information environment
- Act in the role of a designer or architect of identity-based solutions

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comment 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 *Designer 4.0 for Identity Manager 4.0 Administration Guide*, visit the Identity Manager Web site (http://www.novell.com/identity/access/identity_integration).

Additional Documentation

- Understanding Designer for Identity Manager (http://www.novell.com/documentation/designer40/designer_intro/data/front.html)
- Identity Manager 4.0 Installation Guide (http://www.novell.com/documentation/idm40/idm_install/data/front.html)
- Understanding Policies for Identity Manager 4.0 (http://www.novell.com/documentation/idm40/policy/data/bookinfo.html)
- Policies in Designer 4.0 (http://www.novell.com/documentation/idm40/policy_designer/data/bookinfo.html)
- Novell Credential Provisioning for Identity Manager 4.0 (http://www.novell.com/documentation/idm40/policy_credprov/data/bookinfo.html)

- Identity Manager 4.0 DTD Reference (http://www.novell.com/documentation/idm40/policy_dtd/data/bookinfo.html)
- Identity Manager 4.0 Driver Guides (http://www.novell.com/documentation/idm40drivers/index.html)

For more documentation concerning Identity Manager 4.0, see Identity Manager 4.0 Documentation Index (http://www.novell.com/documentation/idm40/index.html).

Installing Designer

1

- Section 1.1, “Requirements,” on page 17
- Section 1.2, “Installing,” on page 17
- Section 1.3, “Upgrading Designer 2.1x To Designer 3.x,” on page 21
- Section 1.4, “Using the Silent Install,” on page 21
- Section 1.5, “Uninstalling Designer,” on page 21

1.1 Requirements

- Section 1.1.1, “System Requirements,” on page 17
- Section 1.1.2, “Platform Requirements,” on page 17

1.1.1 System Requirements

- Minimum resolution is 1024 x 768. The recommended resolution for Designer is 1280 x 1024.
- 1024 MB RAM.
- 1 GB available disk space (recommended).
- 1 GHz processing speed.
- Gettext utilities (Linux only).

  When you install support packages for Designer, such as NICI, certain Linux core utilities are needed. The GNU gettext utilities provide a framework for internationalized and multilingual messages. Before installing Designer, make sure that you have installed this package. You can use YaST to check for dependencies and installed packages.

1.1.2 Platform Requirements

You can install Designer on the following:

- Windows Vista Business and Ultimate with SP1 (32-and 64-bit)
- Windows XP Professional with SP3 (32-bit)
- openSUSE 10.3 (32 and 64-bit)
- SUSE Linux Enterprise Desktop 10 with SP2 (32-and 64-bit)
- SUSE Linux Enterprise Server 10 with SP2 (32-and 64-bit)

1.2 Installing

- Section 1.2.1, “Installing Designer If Eclipse Is Not Already Installed,” on page 18
- Section 1.2.2, “Installing Designer If Eclipse Is Already Installed on Your System,” on page 20

1.2.1 Installing Designer If Eclipse Is Not Already Installed

- "Downloading the Files" on page 18
- "Installing on Windows" on page 18
- "Installing on Windows Vista" on page 19
- "Installing on Linux" on page 19

Because Designer runs in an Eclipse environment, Designer installs Eclipse on your workstation.

Downloading the Files

1. Download the correct `.iso` image or `.zip` file for your system.

   Downloads are available at the following locations:

   - Novell Downloads (http://download.novell.com/index.jsp)
   - Cool Solutions (http://www.novell.com/coolsolutions/dirxml/designer/)
2. If you downloaded a `.iso` image, burn that image file to a CD or DVD.

Installing on Windows

1. From the `designer_install` directory, run `install.exe`.
2. Select a language to use during the installation, then click *OK*.

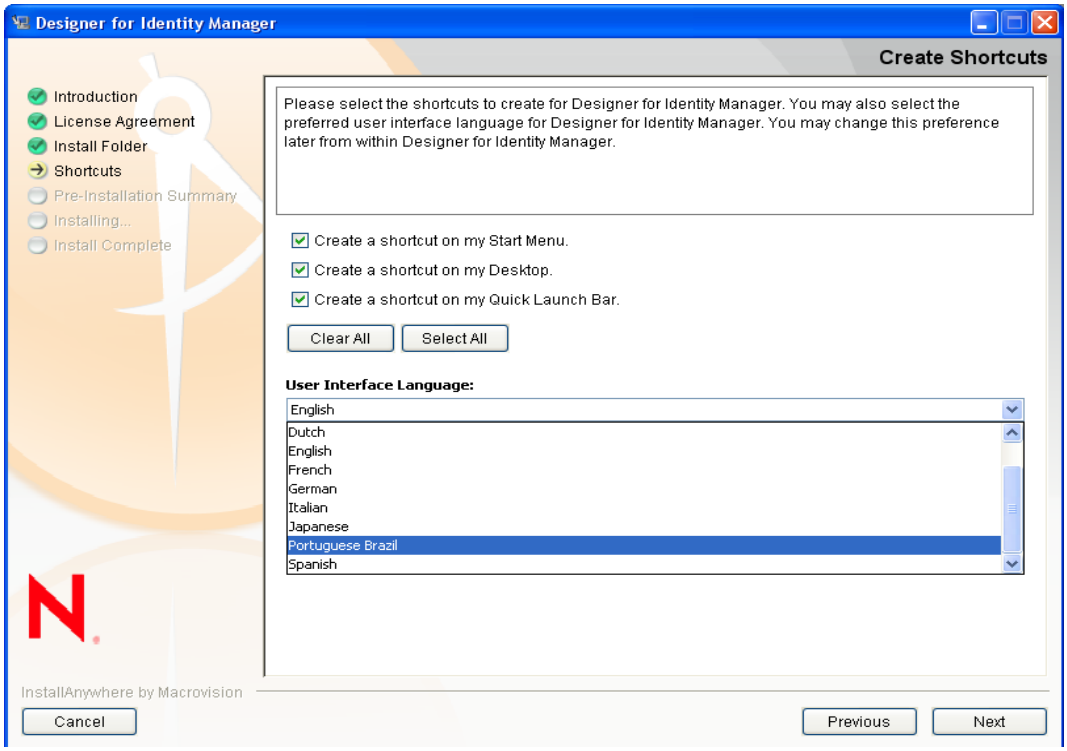
   The default language is English. You can select a different language from the drop-down list.

   The languages displayed in the language-selection dialog box are filtered, based on the default system locale. If the default system locale is English (or another Latin-based character set), only languages that use the Latin character set are displayed.

   If the default system locale is set to Japanese (or any double byte-locale), only the double-byte locales, such as Chinese, Korean, and Japanese, are displayed in the list. The exception to this is English, which is always in the list, regardless of the default system locale.
3. Review the Introduction, accept the license agreement, select a location for the Designer files, then select shortcuts to Designer.

4 Select the desired shortcuts for your Windows machine.

5 Select a language for the Designer user interface.



The default is the language that you selected to use for the installation. You can select a different language from the drop-down list.

After installation, you can change the language. See “Language” on page 453.

6 Review the summary of choices, then install Designer.

7 Review known issues in the Readme, then complete the installation.

8 To run Designer, click the *Designer for Identity Manager* icon on the desktop.

Installing on Windows Vista

Windows Vista has implemented a new User Account Control feature that prevents applications from running as Administrator unless you specifically allow it. If Designer does not install correctly or launch after installation on Windows Vista:

1 Right-click the Designer icon on the desktop.

2 Select *Run as Administrator*.

Installing on Linux

Installing NICI requires `root` privileges. The install program prompts you for the `root` password so that you can install these components.

To install on Linux:

1. From the Identity Manager CD or `.zip` file, run `/linux/designer_install/install.bin`.
2. To run Designer, click the *Designer* icon on the desktop.

Designer installs its own JRE during the installation program. This JRE does not impact Java installed for other uses on the workstation.

IMPORTANT: For updating your JRE, you must note that JRE 1.6 versions upto update 23 ship with CVE-2010-4476 security vulnerability (http://www.oracle.com/technetwork/topics/security/alert-cve-2010-4476-305811.html). This security vulnerability has been addressed in JRE 1.6.0-24 version. You must use the FPUpdater tool that Sun has recently released to update your JRE to JRE 1.6.0-24 version. The instructions for installing the latest JRE versions are available at the JRE Patch Download Site (http://www.oracle.com/technetwork/java/javase/fpupdater-tool-readme-305936.html).

1.2.2 Installing Designer If Eclipse Is Already Installed on Your System

---

NOTE: Because of the interaction between Designer and third-party Eclipse plug-ins, Novell does not support installing Designer into an existing Eclipse installation.

For additional information, see the CoolSolutions Web site (http://www.novell.com/coolsolutions/dirxml/designer/existinginstall_instructions.html).

---

1. Make sure that the stand-alone Designer is not installed on the system.
2. Download a platform-specific file from the eclipse.org Web site:

Platform	Filename
Linux	`wtp-all-in-one-sdk-R-1.5.0-200606281455-linux-gtk.tar.gz`
Windows	`wtp-all-in-one-sdk-R-1.5.0-200606281455-win32.zip`

3. Unzip into the directory of your choice.
4. From the `designer_install/features` directory, unzip all of the files in the folder. This step creates a separate `eclipse` directory with `features` and `plugins` subdirectories.
5. Copy the contents of the Designer `features` and `plugins` subdirectories into the main Eclipse `features` and `plugins` subdirectories.
6. Because of the memory usage of Designer, you will need to change your heap and stack settings for Eclipse. In the Eclipse directory, find the file `eclipse.ini` and add the following lines (or change existing lines to match):
   - -vmargs
   - -Xms512m
   - -Xmx1024m

1.3 Upgrading Designer 2.1x To Designer 3.x

To upgrade to Designer 3.x, follow the same procedure that is outlined in Section 1.2, “Installing,” on page 17. If you install Designer in its present location, you see the Designer Found message, asking if you want to upgrade. If you select *Yes*, Designer deletes the older version of Designer and installs Designer 3.x in its place.

When upgrading to Designer 3.x, take note of three items:

- Do not use Designer 2.1x workspaces for Designer 3.x. Designer stores projects and configuration information in a workspace and these are not compatible between Designer versions.

  In Designer 3.x, default workspaces are stored under the `C:\Document and Settings\Username\designer_workspace` directory for Windows XP, the `C:\users\Username\designer_workspace` directory for Windows Vista, and the `\home\username\designer_workspace` for Linux.
- Import all Designer 2.1x projects into Designer 4.x. This runs the Converter wizard, making the projects compatible to run in Designer 3.x. Be sure the *Copy project into the workspace* option is selected. For more information on the Project Converter, see Section 17.1, “Converting Earlier Projects,” on page 373.
- If you are running workflow provisioning and provisioning with roles, follow the installation or upgrade procedures as outlined in Identity Manager Roles Based Provisioning Module 3.7 (http://www.novell.com/documentation/idmrbpm37/index.html).

1.4 Using the Silent Install

You can install Designer by using scripts. This feature is a silent install.

To use the silent install, run the install program with `-i silent` option. The option is the same whether you install on Windows or Linux. For example:

- **Windows:** `install.exe -i silent`
- **Linux:** `./install -i silent`

The `-i silent` option uses all of the default values in the installation. To change the default values:

1. Issue the command `install -r`.

   The `-r` option creates `installer.properties` file.
2. Edit the `installer.properties` file to add the desired values.
3. Issue the command `install -i silent -f installer.properties`.

1.5 Uninstalling Designer

- “Uninstalling on Windows” on page 22
- “Uninstalling on Linux” on page 22

Uninstalling on Windows

1. In the Control Panel, select *Add/Remove Programs*.
2. Click *Designer for Identity Manager > Change/Remove > Uninstall > Yes*.

To easily uninstall on English-language workstations, select Uninstall from the Start menu. For example, on Windows, click *Start > All Programs > Novell Designer for Identity Manager > Uninstall*.

Uninstalling on Linux

1. Make sure that you have the correct privileges to uninstall.
2. Run `Uninstall_Designer_for_Identity_Manager`.

   This file is in [*`path you chose to install into`*]`/designer/UninstallDesigner`.

Creating a Project

2

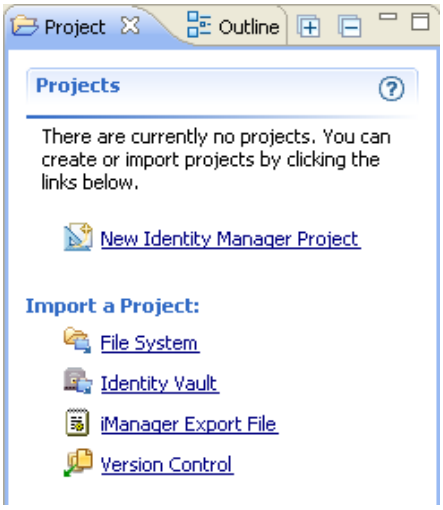
IMPORTANT: Projects created in Designer releases earlier than Designer 3 work in Designer 3 once they are converted. However, projects created in Designer 2 or 3 don't work in Designer 1.1 or earlier releases.

- Section 2.1, "When No Project Exists," on page 23
- Section 2.2, "When You Want to Create an Additional Project," on page 24
- Section 2.3, "When You Want to Import a Project," on page 26
- Section 2.4, "When You Want to Disable a Project," on page 26

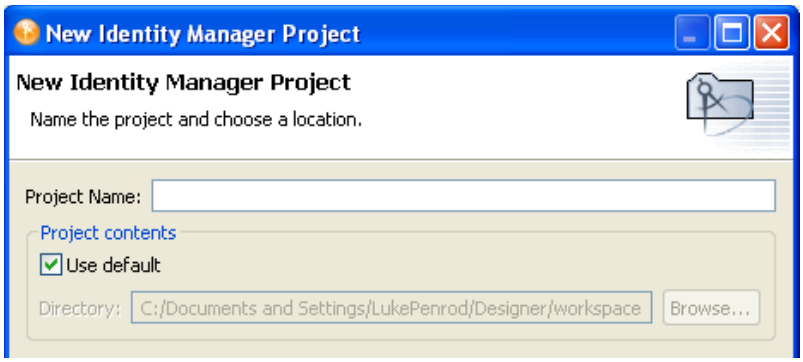
2.1 When No Project Exists

1 Make sure that the Designer perspective (in the upper right corner) is selected.

2 If you are just starting Designer and have no projects in the Project tab, you see the following window:



3 Click *New Identity Manager Project* to launch the Identity Manager Project Wizard.



4 Name the project, then click *Finish*.

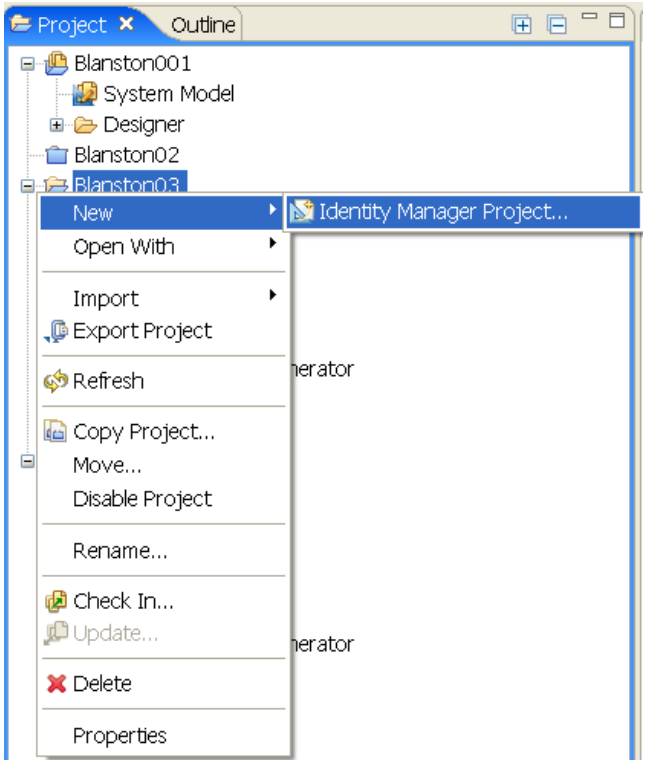
Ensure that you protect your work by periodically saving your project by backing up your project files and project data. You use the Version Control feature in Designer 3.x to automatically create versioned backups of your projects. You can also easily make a copy of your projects by performing the following:

1 Right-click a project in the Project view.

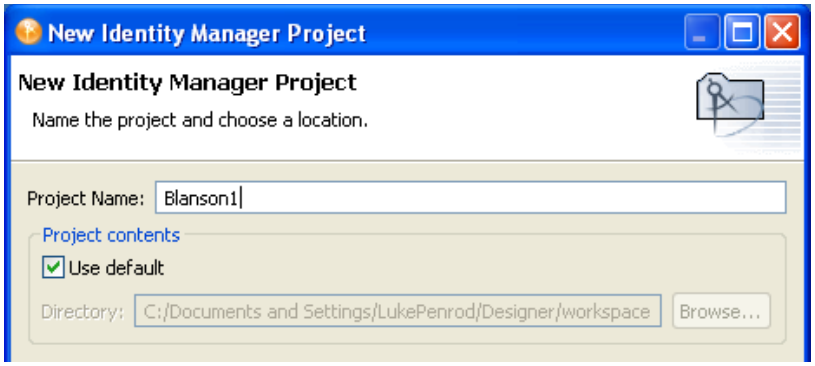
2 Select *Copy Project*.

2.2 When You Want to Create an Additional Project

1 Right-click in the Project view pane, then click *New* > *Identity Manager Project*.



2 In the Identity Manager Project Wizard, name the project.



Designer stores the project in a local directory. You specified this directory when you installed Designer. Typically, this default directory is `c:\Documents and Settings\Username\designer_workspace` for Windows XP and `c:\users\Username\designer_workspace` for Vista. To specify a different directory, deselect *Use Default*, then browse to and select the desired directory.

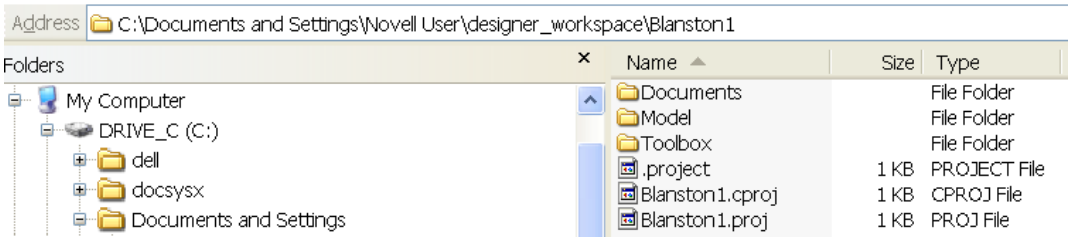
WARNING: Earlier Designer workspaces are not compatible with Designer 3.5 and above. Designer stores projects and configuration information in a workspace. These workspaces are not compatible from one version of Designer to another. You need to point Designer 3 to a new workspace, and not to a workspace used by a previous version of Designer.

If you have Designer 2.x or 3.5 Milestone projects, you can import the projects into Designer 3.5 (*File > Import > Project from File System*). Be sure *Copy project into the workspace* is selected. Importing the project runs the Converter Wizard, making the project compatible with Designer 3.5 architecture and placing it under your designated Designer 3 workspace directory (`designer_workspace` by default).

---

3 Click *Finish*.

The project is stored in a directory structure with the project name as the initial directory containing files with a `.proj` and a `.project` extension. In this example, the project is stored in the `c:\Documents and Settings\Novell User\designer_workspace\Blanston1` directory on a Windows XP workstation.



The project name appears in the Project view. When you select the System Model icon under the project name, Designer opens the Modeler (an editor) for the new project.

For information on saving a project, see “Section 17.1, “Converting Earlier Projects,” on page 373”.

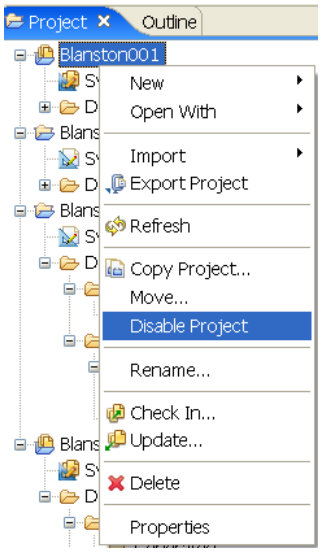
2.3 When You Want to Import a Project

To import a project from an Identity Vault or from the File System, see Chapter 11, “Importing into Designer,” on page 225.

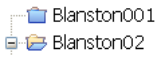
2.4 When You Want to Disable a Project

You can disable and enable projects from the Project view.

1 To disable a project, right-click a project in the Project view and select *Disable Project*.



Once disabled, the project is not accessible from any of the other views, including the Version Control view, and the project is converted to a placeholder in the Project view.



2 To enable the project, right-click the project placeholder in the Project view and select *Enable Project*. The project is again accessible in the other views.

Creating a Model

3

The Designer Modeler lets you create and manipulate a model of your Identity Manager environment within a Designer project. This section discusses the following topics:

- Section 3.1, “Basic Tasks,” on page 27
- Section 3.2, “Accessing the Modeler,” on page 28
- Section 3.3, “Selecting a Modeling Mode,” on page 29
- Section 3.4, “Working from the Palette,” on page 36
- Section 3.5, “Copying and Pasting,” on page 41
- Section 3.6, “Moving Items,” on page 45
- Section 3.7, “In-Line Editing,” on page 46
- Section 3.8, “Tooltips and Toolbar,” on page 46
- Section 3.9, “Organizing by Domain Groups,” on page 47
- Section 3.10, “Connecting Applications,” on page 54
- Section 3.11, “Aligning and Laying Out Components,” on page 61
- Section 3.12, “Editing Multiple Objects,” on page 68
- Section 3.13, “Modeling Active Directory Domain Controllers,” on page 68
- Section 3.14, “Saving Your Model,” on page 70

3.1 Basic Tasks

This section lists basic tasks for creating a model after you have created a project.

1. In Designer, select a project.

   If your project doesn’t appear in the Modeler, open the Project view (*Window* > *Show View* > *Project*), expand the project, then double-click *System Model*.

2. Drag an Identity Vault object from the palette to the Modeler.

   When you create an Identity Vault or server in Designer 3, the default Identity Manager engine version is 3.6. Designer assumes that the Identity Vault has 3.6 capabilities. You can successfully deploy and run 3.6 projects only on Identity Manager 3.6 servers.

   You can easily change the engine version by selecting a version from the Server DN field. However, selecting earlier engine versions takes out any later version capabilities and features from within Designer.

   Before you deploy a project, you must associate a server with the Identity Vault. You do this through the Identity Vault properties. See Section 4.3, “Configuring Identity Vaults,” on page 74.

   You can add multiple Identity Vaults.

3. Configure a driver set.

   Each Identity Vault contains a driver set. See Section 4.5, “Configuring Driver Sets,” on page 78.

4 Add applications.

Drag applications from the palette to the Modeler view. See Section 4.11, “Configuring Application Properties,” on page 120.

5 Configure drivers.

Driver connections are automatically drawn between the application and the driver set. See Section 4.7, “Configuring Drivers,” on page 87.

6 Develop and customize your model.

Develop according to what you planned in Planning Identity Solutions (http://www.novell.com/documentation/designer30/designer_intro/data/gsplanning.html) in the *Understanding Designer for Identity Manager*.

7 Save your model (design).

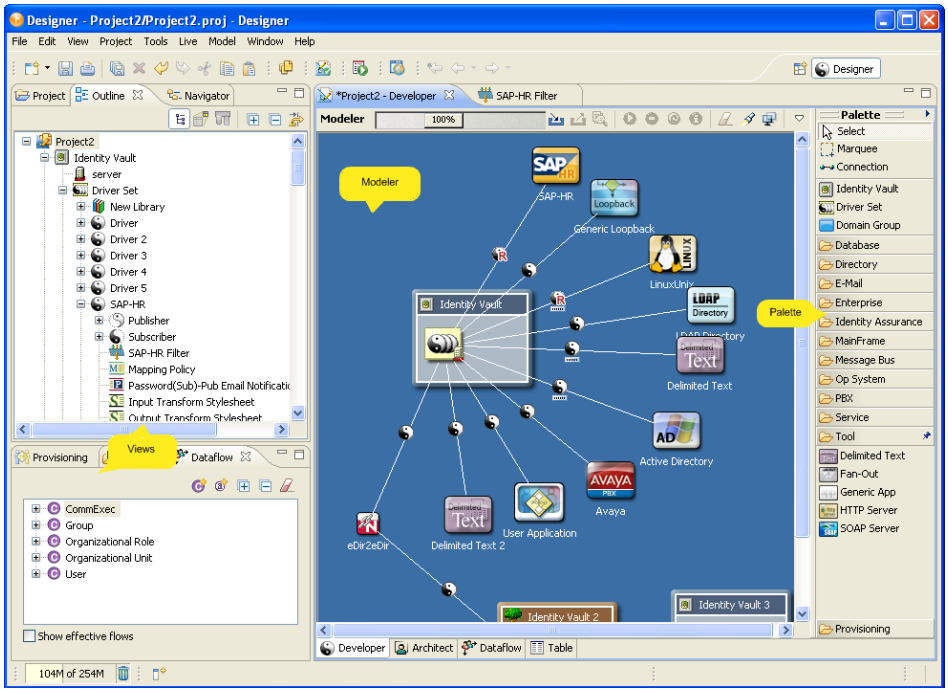
Do one of the following:

- From the main menu, select *File* > *Save* (or *Save All*).
- From the main menu, select *File* > *Close* > *Yes*.
- Click the *X* in the Modeler’s tab, then select *Yes*.

3.2 Accessing the Modeler

The Modeler space is the main working area. It is an editor where you design projects. It is the main work space and primary means of interacting with Designer. All other editors, views, and dialog boxes support and provide functionality for the Modeler.

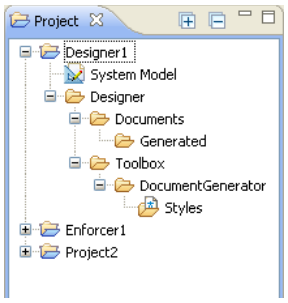
Figure 3-1 *Designer’s Modeler*



To get started, you create a project and drag items from the palette into the Modeler space. Then you arrange and configure the items.

If the Modeler doesn't display:

1. Expand a project in the Project view.



   If you haven't yet created a project, create one.

2. Double-click *System Model*.

3.3 Selecting a Modeling Mode

- Section 3.3.1, "Developer Mode," on page 30
- Section 3.3.2, "Architect Mode," on page 31
- Section 3.3.3, "Dataflow Mode," on page 35
- Section 3.3.4, "Table Mode," on page 36

The Modeler has tabs along the bottom, so that you can switch among different modeling modes. The modes have different advantages, depending on the task you're trying to do and the role that you are acting in.

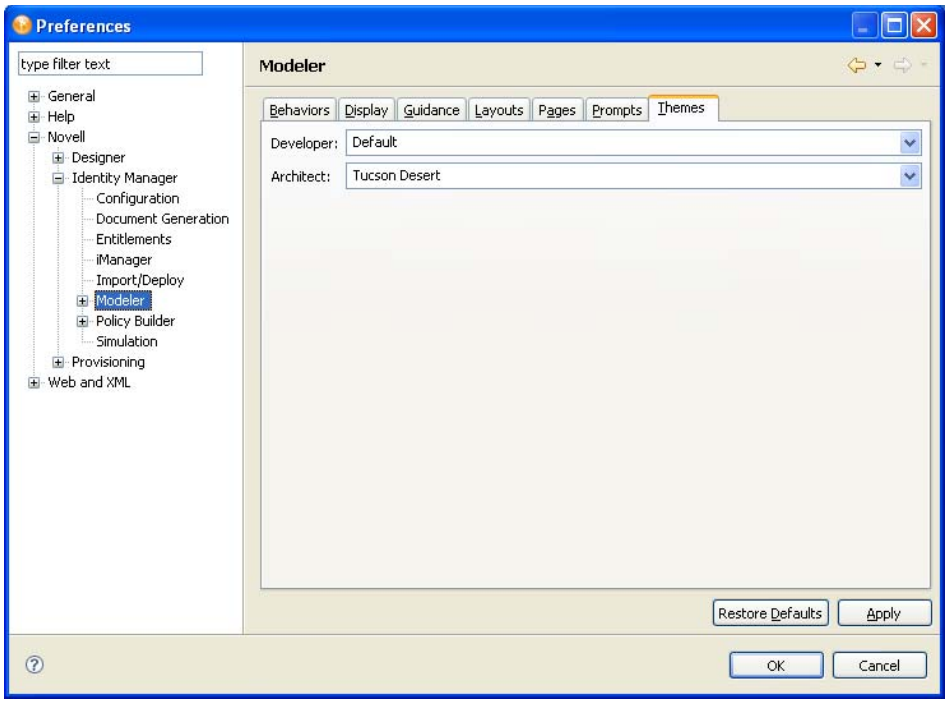
Figure 3-2 *Modeler Modes*

The modes are synchronized with each other with selection, data, and content. They are also synchronized with the Outline View and Thumbnail view.

As you switch modes in the Modeler editor, the editor tab at the top displays the mode that you are in. Also as you switch modes in the Modeler editor, Designer also remembers and restores to the Modeler page you were last on when you close and re-open a project. This helps you stay in the last mode you were last in.

By default, the theme preference is different for each mode. To configure each theme independently in the Modeler preferences:

1. Click *Window* > *Preferences* then select *Novell* > *Identity Manager* > *Modeler*.

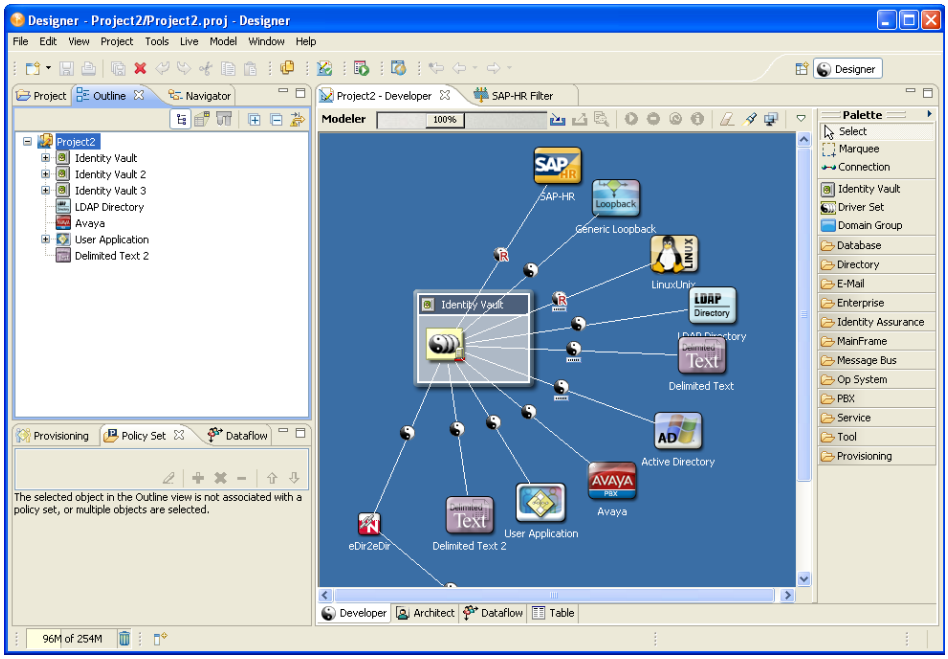


2 Click the *Themes* tab.

3 Select a theme, then click *OK*.

3.3.1 Developer Mode

Figure 3-3 *Developer Mode*



Use Developer mode to do all low-level operations with driver sets, drivers, policies, and applications. This mode lets you manage all of the visual elements and configuration details that you need to fully build and deploy an identity solution.

In Developer mode, the palette organizes the applications and systems into categories. You can customize them to show up as one alphabetical list by using the Modeler Preferences. See “Palette Page” on page 471.

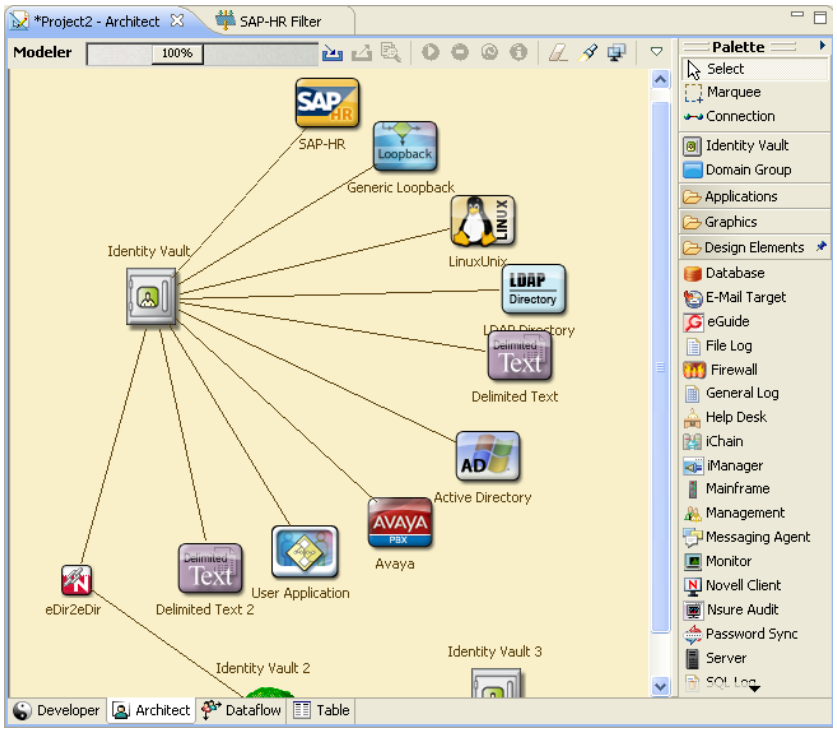
Working with Labels

Figure 3-4 *An Application's Label*

By default in both Developer and Architect modes, labels appear under application icons in the Modeler. They also appear above Identity Vaults in Architect mode.To configure these labels to not appear, use the Modeler Preferences. See “Modeler” on page 469.

3.3.2 Architect Mode

Figure 3-5 *Architect Mode*



Use the Architect mode to work at a design level for your projects. Because the design level doesn't show drivers, driver sets, or policies, you focus more on systems. This mode helps you do large-scale design, which is more intuitive to architects and business strategists.

It is quite likely that you will start in this mode when you begin each project. You will probably spend time putting together an accurate diagram of your enterprise as you consult with various people throughout your organization. As you do so, you should capture key information on each system, such as the owner, contact information, machine environment, software versions, and authentication credentials. As you go through this process, you will also define your project requirements, start thinking about your data, and capture that information in your project.

When the time is right, you can switch to the Developer mode and delve into the technical details of building a working solution. Depending on the size of your project and the makeup of your team, you could have architects and designers build high-level solutions with Designer in the Architect mode, and then send the project to identity developers who understand the details of writing policies and configuring systems. They can share the same project.

In Architect mode, you can connect any design element with any other design element, application, image, or Identity Vault. The connecting lines enable you to express any relationship, making Architect mode a general-purpose, high-level business modeler. The Architect-mode lines don't display when you switch to the Developer mode.

NOTE: When you add icons representing driver applications through the Architect mode, you will need to configure those drivers in the Developer mode. Once you have added the necessary drivers and switch to the Developer tab, right-click the line between the driver icon and the driver set, then select *Run Configuration Wizard*.

The design elements have connectivity information tied to them. You can use design elements to perform live operations or to remotely control other elements that are in your environment but are not necessarily included in your Identity Manager infrastructure.

When using the Architect mode, you should be familiar with the following:

- "The Palette in Architect Mode" on page 32
- "High-Level Data Flows in Architect Mode" on page 33
- "Tasks" on page 34

The Palette in Architect Mode

In Architect mode, the palette lists all applications in one folder and design elements in another folder. The Architect Modeler view now contains all of the graphical modeling tools that are present in the Developer Modeler view. This includes:

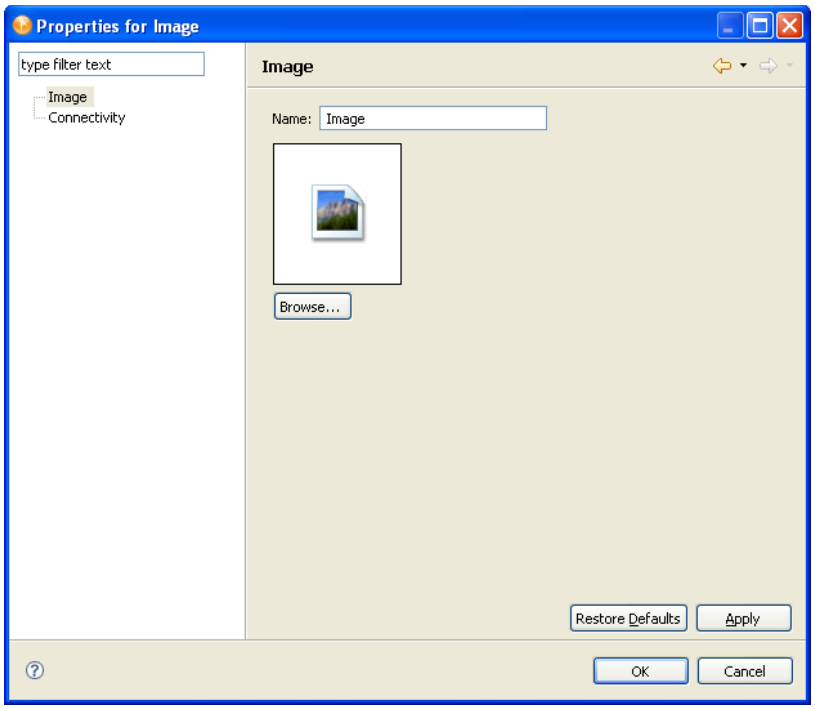
- Rulers
- Snap-in guides
- Alignment hints
- Grid
- Snap-in movement

The Graphics folder has an Image icon. When you drag this icon to the Modeler, Designer displays a generic graphic:

Figure 3-6 *The Image Icon*

To edit the properties of this icon:

1. Right-click the icon, then select *Properties*.



2. In the *Name* field, replace *Image* with a caption.
3. Browse to and select a replacement graphic, then click *OK*.

   You might need to reduce the size of the graphic before importing it.

After the image is in the Modeler, you can drag it, change it, connect lines to it, align or distribute it, or delete it.

High-Level Data Flows in Architect Mode

To set data flows in Architect mode:

1. Right-click the line between an application and an Identity Vault.
2. Select *Show Dataflow View*.
3. Right-click the line again and select *Dataflow*.
4. Specify synchronization and notification events, then click *OK*.

This option is used the same way as in Developer mode except that in Architect mode, Designer automatically configures all the details (schema, filters, and mapping policies) for you. You won't see the Data Flow Wizard for these details. Before deployment, you can edit the details by using Developer mode.

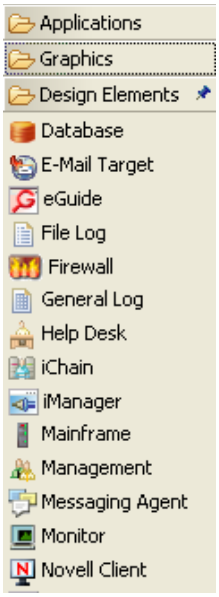
Tasks

You can perform the following tasks in Architect mode:

- Straighten connections (edges). See Section 3.11, “Aligning and Laying Out Components,” on page 61.
- View Password Sync icons and edit synchronization. See Section 7.6, “Integrating Passwords,” on page 191.
- Auto-connect eDir-to-eDir.
- When deleting the driver line, view a prompt to confirm drivers being deleted.
- Display design elements in your model.

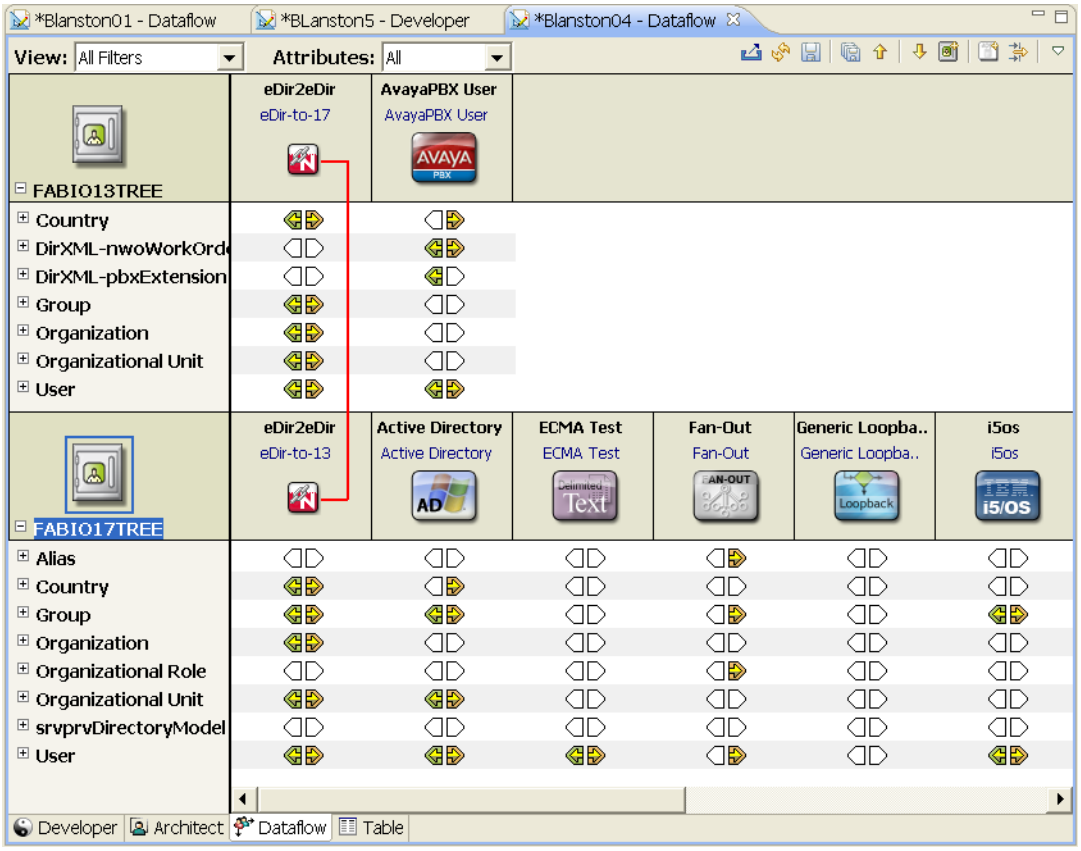
  Open the *Design Elements* folder on the palette, drag design elements onto the Modeler, and connect the design elements.

Figure 3-7 *Items in the Design Elements Folder*



3.3.3 Dataflow Mode

Figure 3-8 *Dataflow Mode*



The Dataflow mode launches the Dataflow Editor, so that you can see all of the filters that control how data flows between the connected systems and Identity Vaults. In the Dataflow Editor, you can right-click an eDir-to-eDir connection and have the option to remove the connection.

The Dataflow mode is synchronized with the Modeler and with the Outline view when you add, delete, change, or synchronize objects. Also, you can see how passwords flow from each server. See Chapter 7, “Managing the Flow of Data,” on page 167.

The Dataflow toolbar enables you to perform the following actions:

- Deploy driver filters for all drivers in the Dataflow view
- Refresh the Dataflow view’s UI screen
- Save the current Dataflow view to an HTML file. You can select a directory to put the file.
- Save all of the filtered views (Notify, Sync, Reset, Password Sync) to an HTML files. You can select a directory to put the files.
- Go up and down to the Identity Vaults
- Create a new Identity Vault
- Add an application driver for a connected system
- Filter Identity Vaults and application drivers out of the Dataflow view

The pull-down menu allows you to perform the following:

- Expand all containers
- Collapes all containers
- Launch Dataflow preferences
- Get help

The Architect and Modeler views contain the same pull-down menu with the same functionality.

3.3.4 Table Mode

Figure 3-9 *Global Table Editor*

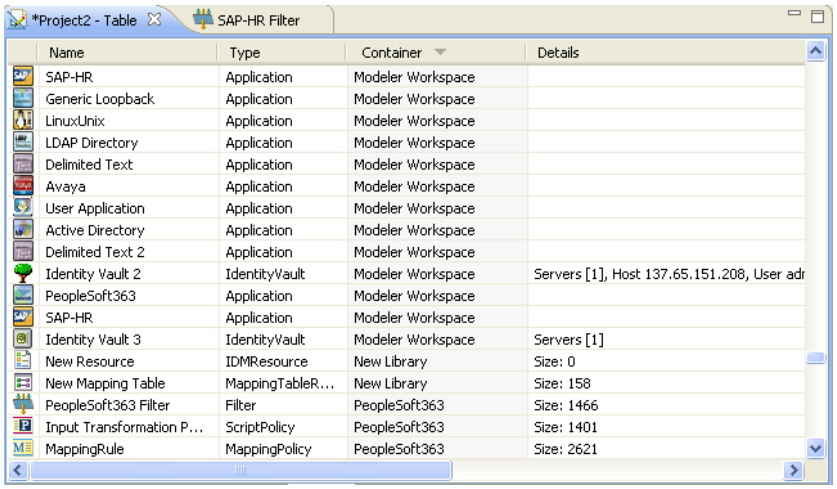


Table mode provides a Global Table editor, which lists all design elements in the project. You can scroll this table to quickly scan essential information, such as the element's type, the container where the element resides, and details, such as an element's size, or driver and serverinformation. You can efficiently find all items of a particular type and edit their settings.

To edit an entry in the table, double-click a line, or right-click a line and select either *Open With*, then an editor. You can also right-click a line, select *Open*, and Designer launches the editor that has been associated with the action. For example, drivers open their Properties page, policies open in the Policy Builder, etc.

When you select an entry in the table, Designer synchronizes the selection with the Outline view, so that you can view the selection's container.

To sort the lists, click a column header.

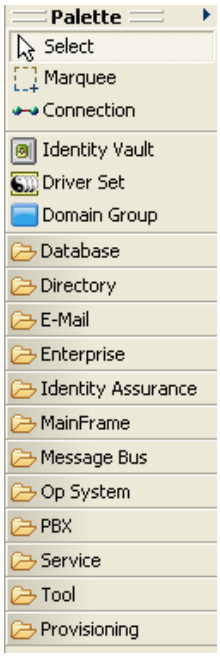
3.4 Working from the Palette

- Section 3.4.1, "About the Palette," on page 37
- Section 3.4.2, "Palette Operations," on page 37
- Section 3.4.3, "Using Generic Applications," on page 38
- Section 3.4.4, "Fly-Out Palette," on page 38
- Section 3.4.5, "Resizing the Palette," on page 39

- Section 3.4.6, “Docking the Palette,” on page 39
- Section 3.4.7, “Arranging Folders and Applications,” on page 40
- Section 3.4.8, “Changing the Layout,” on page 40
- Section 3.4.9, “Keyboard Support for the Palette,” on page 41

3.4.1 About the Palette

Figure 3-10 *Designer's Palette*



The palette is the source of all of the items that you add into the Modeler. To build a model, do one of the following:

- Drag and drop items from the palette to the Modeler space. When you drag and drop an application, it auto-connects to the closest driver set.
- Click an item in the palette, then click in the Modeler space where you want the item to go.

3.4.2 Palette Operations

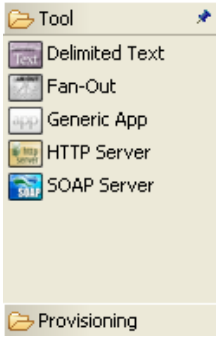
Table 3-1 *Palette Operations*

Operation	Description
Connection	Connects items in the Modeler space.
Identity Vault	Places an Identity Vault in the Modeler space.
Driver Set	Places an eDirectory Driver Set object in an Identity Vault. All applications that you want to connect use a Driver Set object as a hub between the two applications.

Operation	Description
Group	Lets you group and organize items in the Modeler space.
Folders	Applications are organized within folders or drawers. To open or close a folder, click it. To hold the folder in place and make sure that it doesn't fully collapse (even when you open other folders), click the pin. Once the Palette is full unpinned folders automatically close when you open another folder.
Applications	The various applications that you can connect are grouped into folders by type. You can drag and drop these applications to the Modeler space and begin editing them. The Modeler automatically adds a connecting line, which represents a driver.
Scrolling Arrows	Small directional arrows. If a folder has many items, or if the screen area is restricted, scrolling arrows appear. To scroll through he contents of a folder, click the arrows.

3.4.3 Using Generic Applications

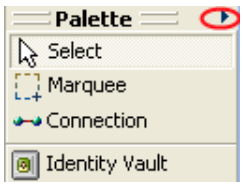
Figure 3-11 *The Generic App Option on the Palette*



Scenario: A Generic Application. Fridrik creates a project with his own items and graphics, in his own version of Designer. He transfers the project to you, but you are using a different version of Designer, which doesn't understand those items. Your version renders the transferred objects as Generic applications.

3.4.4 Fly-Out Palette

Figure 3-12 *The Palette's Control Arrow*



To hide the palette, click the small control arrow on the palette. The palette collapses.

To open the palette again and keep it open, click the arrow.

To temporarily open the palette again, hover the cursor over the collapsed palette, below the control arrow. The palette quickly expands. This is fly-out mode.

To change the palette from fly-out mode, click the control arrow again. The state persists and is restored the next time you run the application.

3.4.5 Resizing the Palette

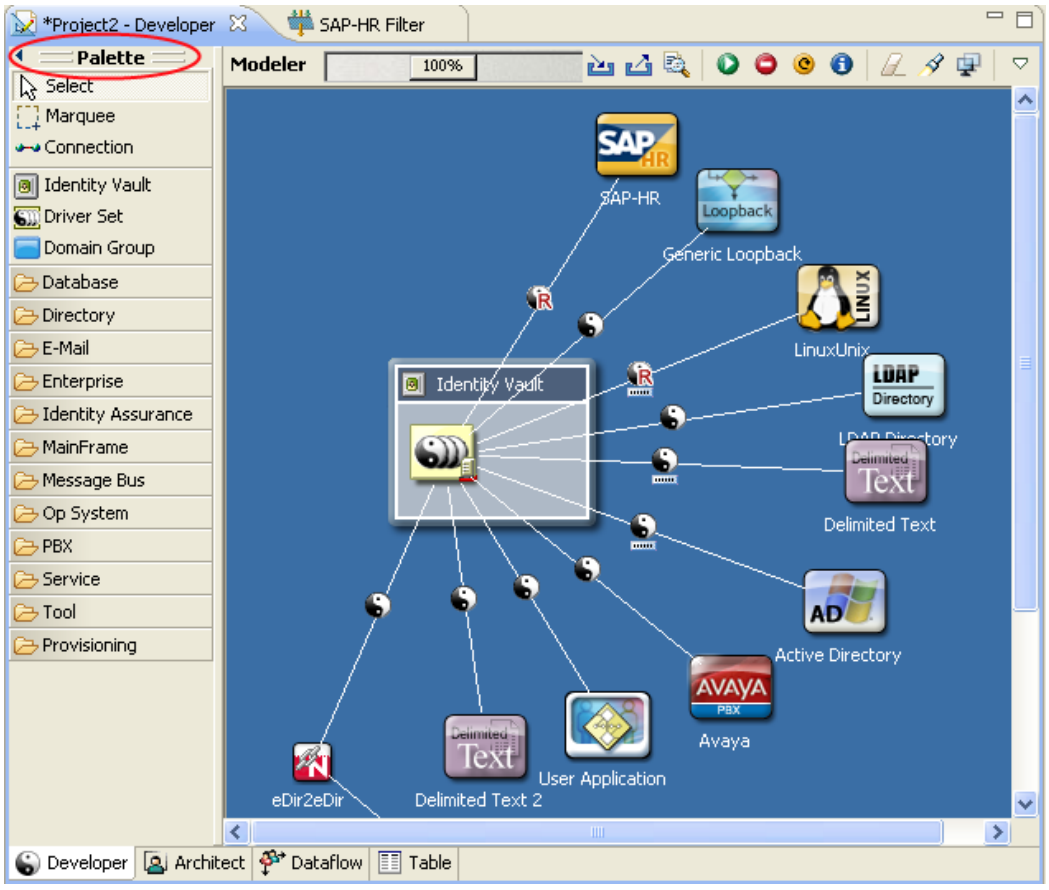
1. Click the palette's thick border that faces the Modeler space.
2. Drag the line.

   The size persists and is restored the next time you run the application.

3.4.6 Docking the Palette

To dock the palette on the left or right of the Modeler space:

1. Click the top palette header.



2. Drag the palette to the desired location.

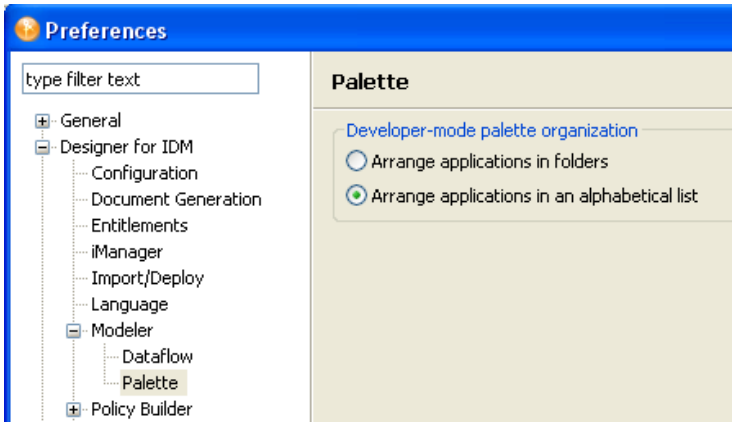
   The location persists and is restored the next time you run the application.

3.4.7 Arranging Folders and Applications

By default, applications are placed in folders.

To arrange applications alphabetically instead of in folders:

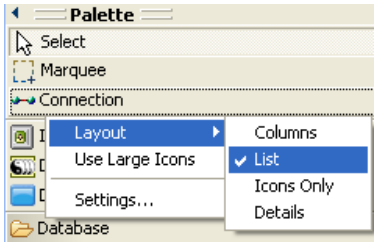
1 Click *Window* > *Preferences* > *Novell* > *Identity Manager* > *Modeler* > *Palette*.



2 Select *Arrange applications in alphabetical list*, then click *OK*.

3.4.8 Changing the Layout

1 Right-click the palette.



2 Select *Layout*.

3 Select an option.

Setting	Description
Layout: Columns	Displays folders and applications in columns.
Layout: List	Arranges folders and applications in a list.
Layout: Icons Only	Removes descriptive labels.
Layout: Details	Briefly describes palette items.
Use Large Icons	Toggles the size of icons used for applications.
Settings	Enables you to set the layout and icon size in one dialog box. Controls how folders (drawers) behave.

3.4.9 Keyboard Support for the Palette

Table 3-2 *Shortcut Keys for the Palette*

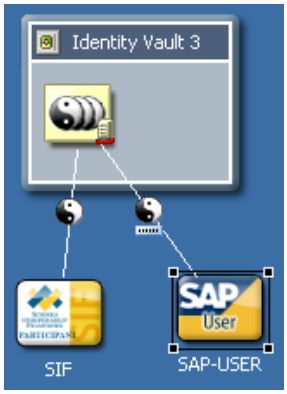
Keystroke	Description
Left-arrow	Collapses an open folder. The focus must be on the folder, not the application.
Right-arrow	Opens a collapsed folder. Moves into an open folder.
Up-arrow	Moves up to the next folder.
Down-arrow	Moves down to the next folder.

3.5 Copying and Pasting

- Section 3.5.1, “Copying Applications,” on page 41
- Section 3.5.2, “Copying a Driver Set,” on page 42
- Section 3.5.3, “Copying an Identity Vault,” on page 43
- Section 3.5.4, “Copying a Domain Group,” on page 44
- Section 3.5.5, “Copying between Editors,” on page 44

3.5.1 Copying Applications

Figure 3-13 *Applications to Copy*



You can copy and paste the following within the same editor or to another editor:

- Applications, including custom applications
- Disconnected applications
- Driver icons

1. Select an application or driver icon.
2. Press *Ctrl+C*, then *Ctrl+V*.

The copy and paste operations are also accessible from the Clipboard context menu. (Currently, they aren't accessible from the main menus.)

When you copy an application in the same editor, Designer copies all of the application's attributes, and copies all subelements. Therefore, all drivers that the application is connected to are copied, and all policies that the drivers contain are also copied. The new application defaults to connect to the same driver sets that the previous application connected to.

To copy an application to a different driver set (in the same editor or in another editor):

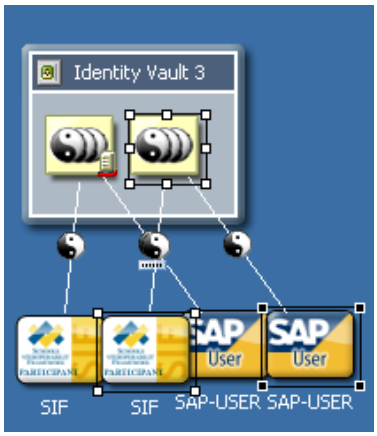
1. Select the application.
2. Press Ctrl+C.
3. Select the target driver set that the application will connect to.
4. Press Ctrl+V.

If you copy and paste an application without selecting a target driver set, Designer makes a copy and connects it to the current driver set.

You can select multiple applications and then copy and paste them.

3.5.2 Copying a Driver Set

Figure 3-14 *Driver Sets*



You can copy and paste driver sets within the same Identity Vault or to another Identity Vault in the same editor or in another editor.

1. Select a driver set.
2. Press *Ctrl+C*, then *Ctrl+V*.

When you copy a driver set in the same editor, Designer copies all of the attributes of the driver set, including the following:

- All drivers that the driver set is connected to
- All policies that the drivers contain
- All target applications

To copy to a different editor:

1. Select a driver set.
2. Press *Ctrl+C*.
3. Select the target Identity Vault in the other Modeler editor where you want the driver set to be copied to.
4. Press *Ctrl+V*.

By default, the new driver set is created in the same Identity Vault as the one that it was copied from. However, if you select another Identity Vault, the driver set is copied there.

After you copy and paste, you might need to move the pasted objects to a better location so that they don't cover up an existing object. To do this, leave the objects selected after you paste them, then move them. Another way to easily select objects:

1. Right-click a driver set.
2. Select *Select All Connected Applications*.
3. Move one of the selected applications.

   All connected applications move together.

When you copy a driver set, it has the same settings, except for the selected servers, which are blank. This exception occurs because the Identity Manager engine does not allow more than one driver set on an Identity Vault to be associated with the same server. Therefore, you need to set up the servers for the new driver set. If you copy an Identity Vault, Designer copies the driver sets. The new driver set has the same server settings set up for you.

You can select multiple driver sets and then copy and paste them. To copy and paste multi-driver connections, you must copy the driver set or Identity Vault that contains them. In Designer 2.0, if you copy the application that has a multi-driver connection, the application and only one of its drivers are copied.

3.5.3 Copying an Identity Vault

You can copy and paste Identity Vaults within the same editor, to another editor in the same Modeler space, or in a specific Domain Group.

1. Select an Identity Vault.
2. Press *Ctrl+C*.
3. Select nothing or the target Domain Group (in the same editor or another) where you want the Identity Vault to be copied to.

   If you select nothing, the new Identity Vault is copied to the right of the previous Identity Vault in the current editor.
4. Press *Ctrl+V*.

The new Identity Vault appears to the right of the previous Identity Vault and is the same size as the one that it is being copied from.

When you copy an Identity Vault, Designer copies all of the elements of the Identity Vault. The elements include servers, e-mail templates, driver sets, and connected applications.

You can select multiple Identity Vaults and then copy and paste them.

3.5.4 Copying a Domain Group

You can copy and paste Domain Groups within the same editor or to another editor in the same Modeler space or in a specific Domain Group.

1. Select a Domain Group.
2. Press *Ctrl+C*.
3. Select the location for the new Domain Group.

   If you select nothing, the new Domain Group is copied to the right of the previous Domain Group in the current editor.
4. Press *Ctrl+V*.

The new Domain Group appears to the right of the previous Domain Group, and is the same size as the one it was copied from.

When you copy a Domain Group in the same editor, Designer copies all of the attributes of the Domain Group. However, Designer doesn't copy all subelements.

You can select multiple Domain Groups and then copy and paste them.

3.5.5 Copying between Editors

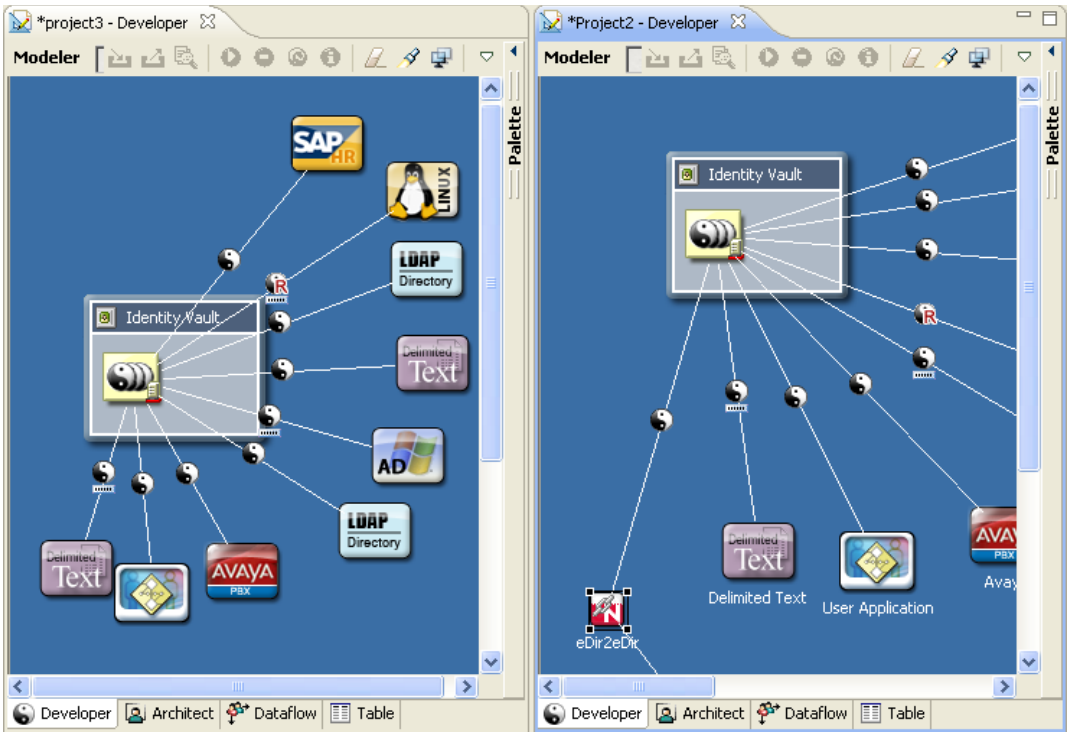
To easily copy and paste between two editors:

1. Using the Project view, open two projects.

   One project is active. The second project's tab displays at the top of the Modeler.
2. Close the palette by clicking the control arrow on the palette's title bar.

3. Click the second project's tab and drag it to the Modeler's right border.

   The tab changes to a folder icon until it arrives near the border, where the folder changes to an arrow.
4. Release the mouse.



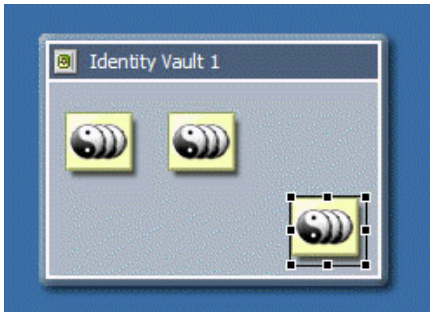
5 Copy items from one editor to the other.

3.6 Moving Items

After an item is in the Modeler space, you can move it by dragging it to a new location. The Modeler prevents you from placing objects where they don't belong. For example, you cannot move a driver set out of an Identity Vault to the Modeling space, or drop an application inside of an Identity Vault. You can always drag objects into a Domain Group, or drag a driver set from one vault into another.

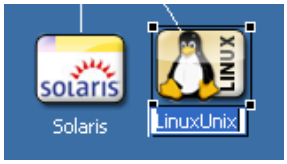
If you drag a driver set into an Identity Vault, the Identity Vault automatically grows or shrinks to fit the driver set, so you don't need to manually resize the vault. This behavior can be turned on or off in Preferences. See "Modeler" on page 469.

Figure 3-15 *Example Driver Sets in an Expanded Identity Vault*



3.7 In-Line Editing

Figure 3-16 *An In-Line Edit*



To edit the names of objects, do one of the following:

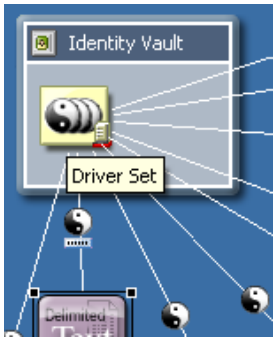
- Select the item, press F2, then edit the label.
- Double-click the item, then edit the *Name* field.

You can do an in-line edit for any type of item in the Modeler, including the driver lines.

3.8 Tooltips and Toolbar

As you mouse over objects in the Modeler, a tooltip appears with the name of the object.

Figure 3-17 *A Tooltip*



The Modeler also provides a toolbar.

Figure 3-18 *The Modeler Toolbar*

The Modeler toolbar enables you to quickly find often-used features:

- Search
- Find a driver's status (also available from the *Live* menu when you select a driver set or Identity Vault)
- Start, stop, or restart a driver (also available from the *Live* menu when you select a driver set or Identity Vault)
- Clear all items
- Save a snapshot of the model

The pull-down menu allows you to perform the following:

- Expand all containers
- Collapes all containers
- Launch Modeler preferences
- View demos on how to use the Designer
- Get help

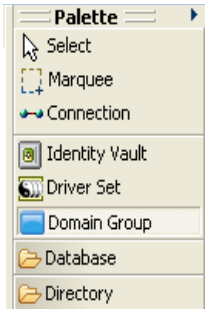
The Architect and Dataflow views contain the same pull-down menu with the same functionality.

3.9 Organizing by Domain Groups

- Section 3.9.1, “About Domain Groups,” on page 47
- Section 3.9.2, “Key Features,” on page 48
- Section 3.9.3, “Creating a Domain Group,” on page 48
- Section 3.9.4, “Minimizing (Collapsing) Domain Groups,” on page 50
- Section 3.9.5, “Restoring Domain Groups,” on page 51
- Section 3.9.6, “Maximizing Domain Groups,” on page 51
- Section 3.9.7, “Auto-Placement of Neighbors,” on page 51
- Section 3.9.8, “Grouping into a New Domain Group,” on page 51
- Section 3.9.9, “Ungrouping a Domain Group,” on page 52
- Section 3.9.10, “Clearing Contents,” on page 53
- Section 3.9.11, “Changing a Domain Group Icon,” on page 53
- Section 3.9.12, “Keyboard Support for Domain Groups,” on page 54

3.9.1 About Domain Groups

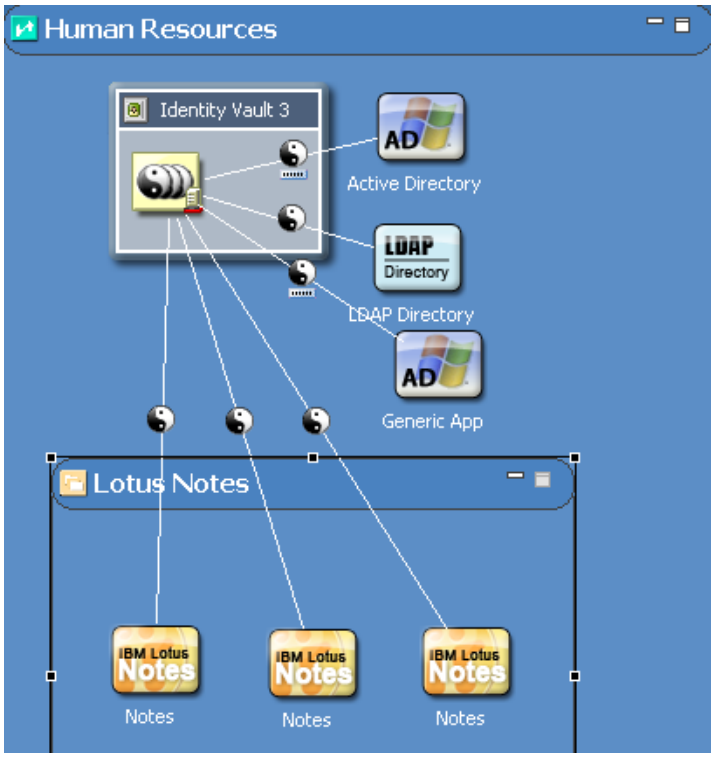
Figure 3-19 *The Domain Group Option on the Palette*



Domain Groups enable you to organize your model into logical groupings that help to keep your diagram clean. Domain Groups have no technical function, and they have no impact on how items and relationships are stored in the Identity Vault. This option is just a tool to help you better organize and view items in the Modeler.

Using Domain Groups is the key to modeling your entire enterprise, no matter how large it is. You can create a model that is manageable, useful, and logical, according to how you want to organize and diagram your enterprise.

Figure 3-20 *A Domain Group in the Modeler*



3.9.2 Key Features

- Change a group name through the Properties view.
- Drag and drop items in and out of groups.
- Minimize or restore groups.
- Move everything in a group.
- Remove everything in a group.
- Nest groups within groups (no limit).
- Resize groups. A minimum size is enforced.
- Ungroup. Remove the group but leave the children.

3.9.3 Creating a Domain Group

1. Drag and drop a Domain Group from the palette to the Modeler space.
2. Organize items inside Domain Group items.

To add another Domain Group, drag and drop one from the palette.

To add an Identity Vault, do one of the following:

- Drag an Identity Vault from the palette.
- Right-click in the Domain Group, then select *New* > *Add Identity Vault*.

  The Add Server to Identity Vault dialog box appears. If you select *Specify a Server*, Designer provides a dialog box that enables you to select an eDirectory server or specify a server manually.

To add a driver set:

1. Right-click inside an Identity Vault.
2. Select *Add Driver Set*.

To add an application:

1. Right-click a Driver Set object.
2. Select *Add Connected Application*.

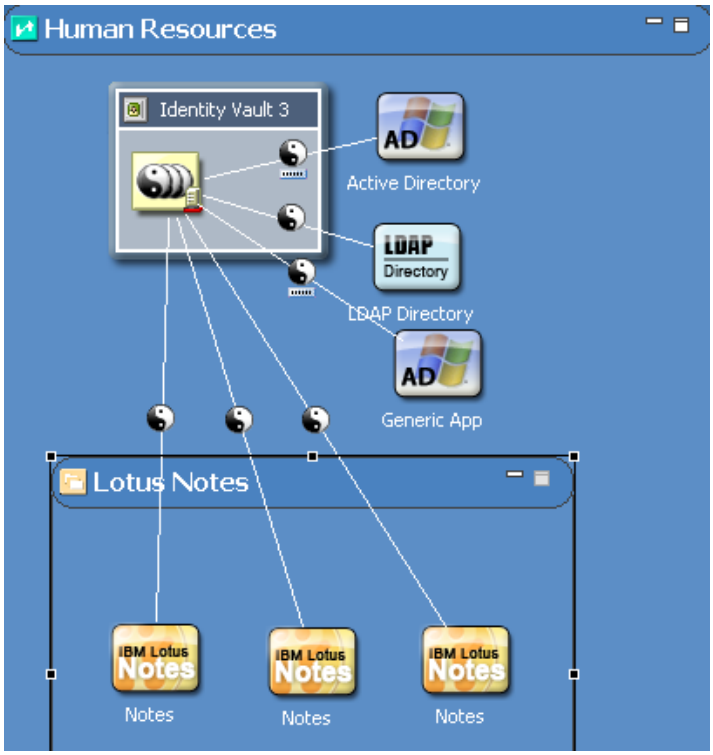
   The application is added to the right of the right-most connected application. If this is the first application, it is placed under the driver set.

The application defaults to a generic application type. To change the type:

1. Right-click the application, then select *Properties*.
2. Select a different application, then click *OK*.

When you add selected items to a Domain Group, the Domain Group expands.

Figure 3-21 *A Domain Group*



If you move an item to the edge of the Domain Group, the boundaries expand, so that the items remain inside the Domain Group. You can drag an item from the Domain Group to remove it from the group.

You can have nested domains. If you expand a nested domain, the outer (hosting) domain auto-increases in size. You aren't required to manually resize parent domains. By expanding, the hosting domain displays the nested domain, so that the nested domain isn't cut off.

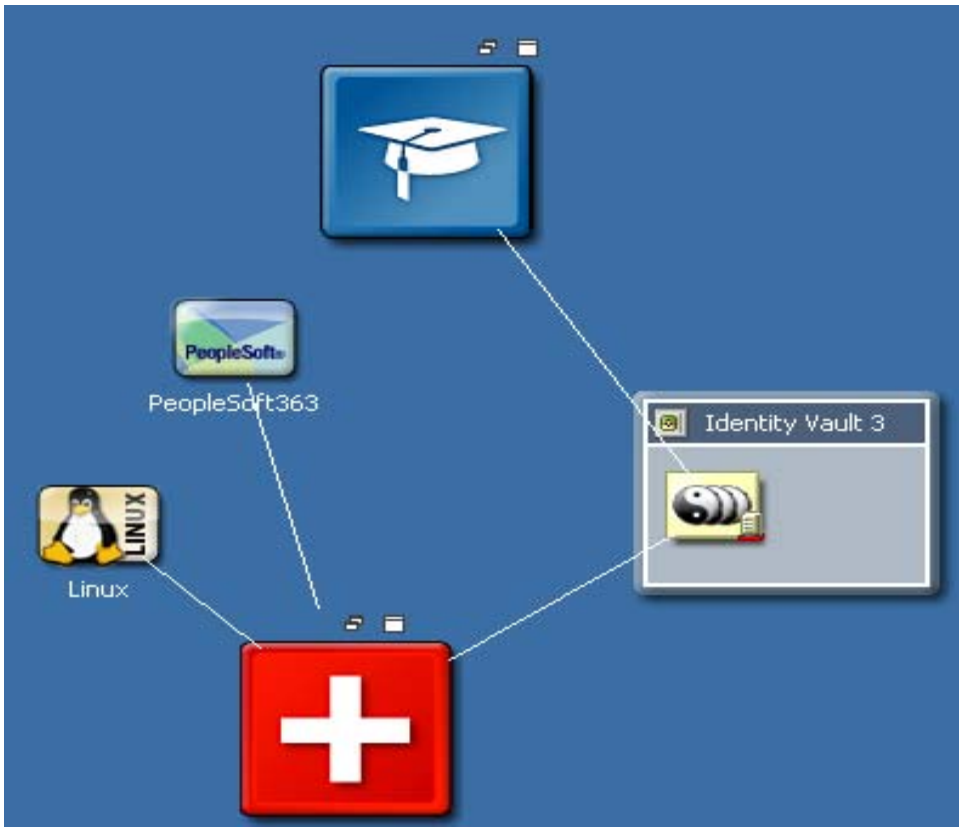
3.9.4 Minimizing (Collapsing) Domain Groups

To minimize a Domain Group, click the *Minimize* button. When a Domain Group is minimized, it defaults to a random icon. You can use Properties to change the icon. (See "Changing a Domain Group Icon" on page 53.) The icon and minimized state of the group are saved in the Project file.

When a group is minimized, you can't see its contents, nor can you drag new items into the group. However, you can move, rename, or delete it.

When you minimize a group, lines that were connected to items in the group now connect to the group. This functionality enables you to see that there is a relationship with items in the group and items outside the group. Depending on your objects, their relationships, and state of other related groups, multiple lines might collapse into one line.

Figure 3-22 *A Collapsed Group*



When you expand the group, the lines are moved back to the actual items they connect with. This functionality works for any level of nesting of groups.

3.9.5 Restoring Domain Groups

To restore the Domain Group to its original size, click the *Restore* button.

3.9.6 Maximizing Domain Groups

To maximize a Domain Group, click the *Maximize* button. The group expands to a much larger size. To return it to the original size, click the *Restore* button.

You can maximize only first-level groups. For inner groups, the Maximize function is disabled.

3.9.7 Auto-Placement of Neighbors

To push or pull the neighboring items when you expand or contract Domain Groups, hold down the Ctrl key while you expand or contract the Domain Group. Any item that is to the right or below a Domain Group is affected.

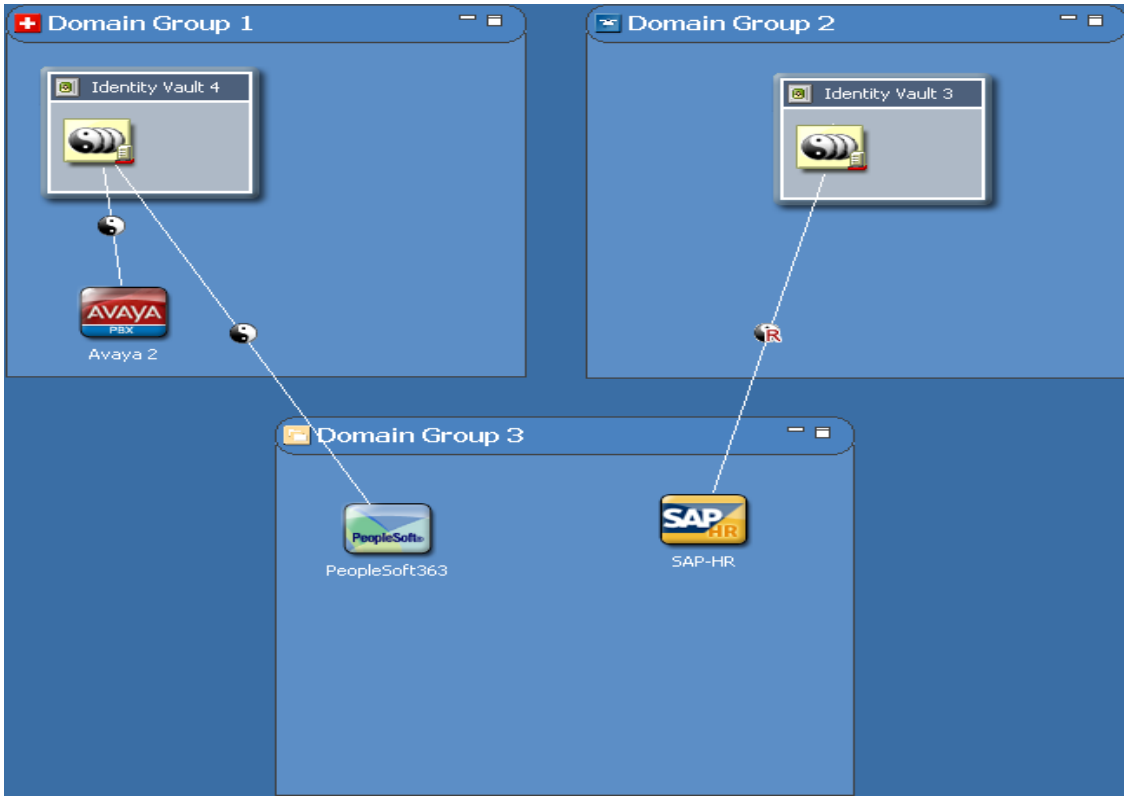
3.9.8 Grouping into a New Domain Group

1. In the Modeler, select multiple items.
2. Right-click, then select *Add to Group*.

The Modeler creates a new Domain Group and adds those items, preserving their relative spacing to each other. This process removes the items from wherever they previously existed and places them in the proper area in the new group.

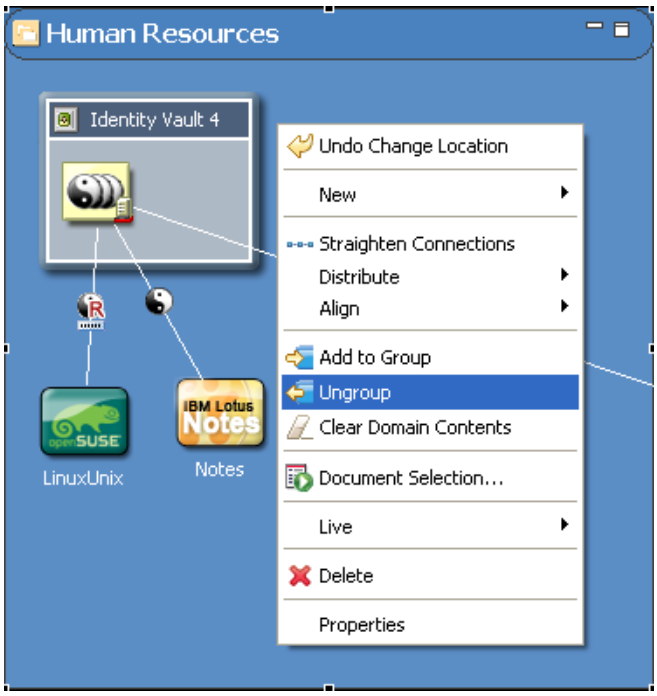
The following figure illustrates two Applications that have been added to a new Domain Group and removed from their previous groups.

Figure 3-23 *Grouping into a New Domain Group*



3.9.9 Ungrouping a Domain Group

Figure 3-24 *Ungrouping a Domain Group*

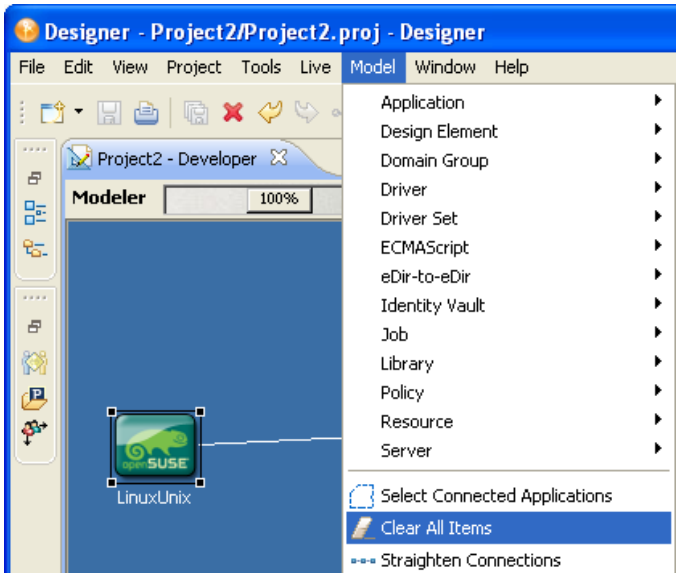


To ungroup a Domain Group, right-click it, then select *Ungroup*.

This process removes the Domain Group but leaves all contents where they are, so that they won't be deleted. This is just a way to ungroup the items. Depending on what level you are in the Modeler, the ungrouped items are automatically added to the host group or to the main Modeling space.

3.9.10 Clearing Contents

Figure 3-25 *Clear All Items*



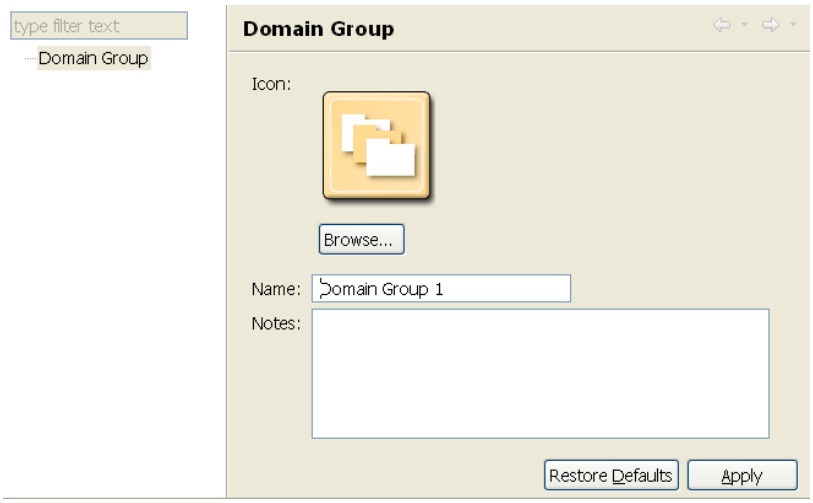
To remove all contents from the Modeler, click *Model*, then select *Clear All Items*.

To remove all contents from a Domain Group, right-click, then select *Clear Domain Contents*.

Designer prompts you before clearing the Modeler space.

3.9.11 Changing a Domain Group Icon

1 Right-click a Group Domain item in the Modeler, then select *Properties*.



2 Browse to and select an image (for example, `finance.png`).

   Icons for Domain Group components reside in the `Group` directory in the Modeler plug-in directory. By default, Designer opens the `Group` directory.

   Designer supports `.GIF`, `.JPEG`, `.PNG`, and Windows `.BMP` formats. You can add your own icons to the `Group` directory.

3 Click *Open*, then click *OK*.

The minimized 16x16 version of the image also now appears in the Domain Group title bar.

As you add Domain Group items, Designer randomly assigns icons from the `Group` directory to the new Domain Group.

3.9.12 Keyboard Support for Domain Groups

Table 3-3 *Shortcut Keys for Domain Groups*

Keystroke	Description
Alt+Down-arrow	Navigates into a Domain Group
Alt+Up-arrow	Navigates out of a Domain Group
Delete	Deletes the selected items

3.10 Connecting Applications

- Section 3.10.1, “Automatic Connections,” on page 55
- Section 3.10.2, “Connection Target Highlights,” on page 55
- Section 3.10.3, “Automatically Creating Objects,” on page 56
- Section 3.10.4, “Auto Redraw,” on page 56
- Section 3.10.5, “Manually Connecting,” on page 56
- Section 3.10.6, “eDir-to-eDir Connections,” on page 56

- Section 3.10.7, “Multiple Driver Connections,” on page 57
- Section 3.10.8, “Straightening Connections,” on page 58
- Section 3.10.9, “Reconnecting,” on page 60
- Section 3.10.10, “Driver Icons,” on page 60
- Section 3.10.11, “Selected Drivers,” on page 60
- Section 3.10.12, “Auto-Layout of Imported Objects,” on page 61
- Section 3.10.13, “Keyboard Support for Connections,” on page 61

3.10.1 Automatic Connections

When you drag an application into the Modeler space, and the Modeler contains a driver set, Designer automatically draws a connecting line between the Driver Set object and the application.

When you use the palette’s Connection function to connect an application to an Identity Vault, you can begin or end your driver line at the Identity Vault. The line automatically connects to a driver set in an Identity Vault.

If the Identity Vault contains more than one driver set, the Connection function connects the driver line to the first driver set. This functionality also works for multi-driver connections.

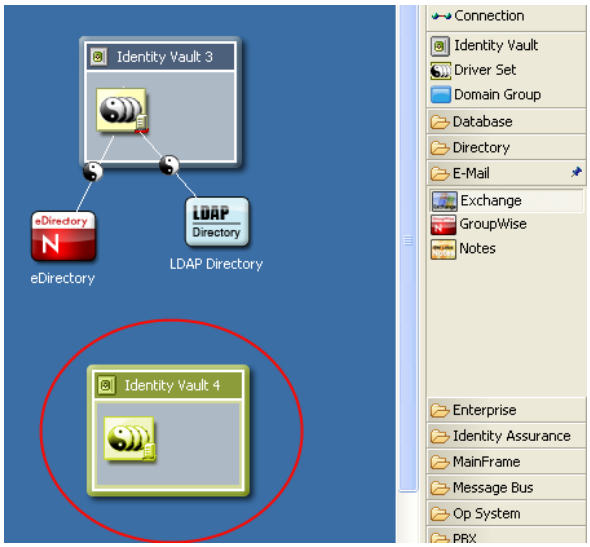
All multi-driver driver lines are bendable. You can lay them out so that the lines don’t overlap at any angle. Also, you can reconnect multi-driver connections.

If an Identity Vault has multi-driver connections in a Domain Group and you minimize that Domain Group, a single collapsed line represents all of the multi-driver connections.

3.10.2 Connection Target Highlights

When you drag an application across the Modeler space, the closest Identity Vault and closest driver set in that Identity Vault are highlighted. The highlights indicate the item that the application will connect with when you drop the application.

Figure 3-26 *Connected Objects*



3.10.3 Automatically Creating Objects

If you drop an application into the Modeler space, and that space has no Identity Vaults, Designer automatically creates an Identity Vault.

If you add a driver application in the Modeler by right-clicking in the Modeler, then selecting *New* > *Application*, the driver application is now added at the place where you right-clicked. This makes it easier to locate items in the view.

3.10.4 Auto Redraw

If you move items, lines are automatically redrawn.

3.10.5 Manually Connecting

To manually connect an application to a driver set:

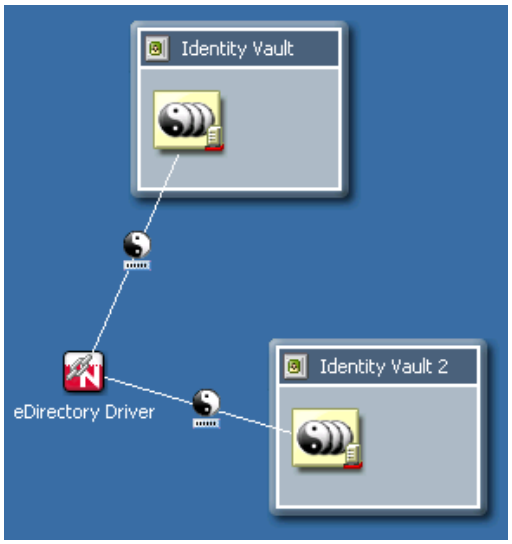
1. Click *Connection* in the palette.
2. Draw a line between the application and the driver set.

To reconnect an application, select the driver line, then drag one end of the line to another driver set or application.

The drag gesture gravitates the line towards the nearest connectable point. This functionality helps you know what you can connect to and where you can connect the item. If you try to connect to something that isn't allowed, the cursor usually indicates so, or nothing happens when you drop the item.

3.10.6 eDir-to-eDir Connections

Figure 3-27 *eDir-to-eDir Connections*



An eDir-to-eDir connection is a special type of connection. It is used frequently in Identity Manager environments. This connection is a way to configure two eDirectory drivers to communicate directly with each other. (No other drivers are able to communicate directly with any other type of driver.) This type of connection is most commonly used for synchronizing a local directory tree with a Metadirectory Identity Vault.

To create an eDir-to-eDir connection, do one of the following:

- Drag a line between two Identity Vaults
- Drag a line between two driver sets

When you connect a line between two eDirectory applications, the line automatically turns into an eDir-to-eDir connection. See the illustration in Section 7.2.6, “Viewing an eDir-to-eDir Driver,” on page 180.

To disconnect an eDir-to-eDir connection, right-click an eDir item, then select *Disconnect eDir-to-eDir*. Designer creates two new eDirectory applications and redirects each driver to its respective application. A new driver is not created. No data is lost. Designer keeps the same drivers.

If you delete one side of an eDir-to-eDir connection, Designer converts the remaining half into a regular driver connection to an eDirectory application.

3.10.7 Multiple Driver Connections

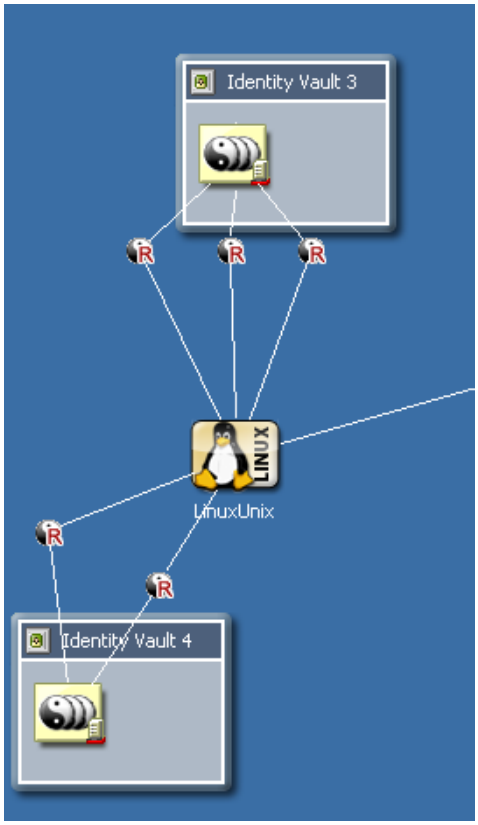
To connect more than one driver from a driver set to an application:

1. Select *Connection* in the palette.
2. Connect the driver set and the application again and again.

Each time you connect, a new line is added. All lines are bendable, so that the lines don’t overlap. To get the model to look optimal, you probably need to move the application slightly from its default position.

You can also connect more than one driver to a single application. This actually causes the application to act as a hub. Each driver can connect to and authenticate to the application or system the same or differently, depending on your needs. Each driver can access the same part of the application or system or different parts (for example, different tables in a database). The Modeler lets you diagram a layout according to your needs.

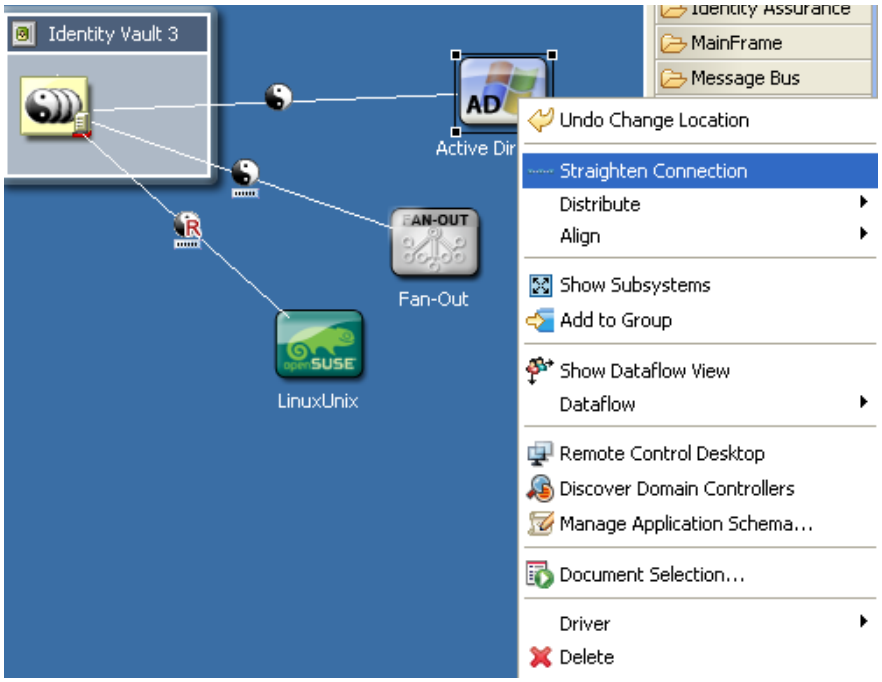
Figure 3-28 *Multi-Driver Connections*



3.10.8 Straightening Connections

To straighten connecting lines:

1. Press Ctrl, then select one or more items in the Modeler.



2 Right-click, then select *Straighten Connections*.

What is straightened depends on what you select:

Table 3-4 *Straightened Connections*

Selected Item	What Is Straightened
A driver	That driver's line
An application	The connecting driver's line
An Identity Vault	All lines that originate from that driver set in that Identity Vault
A Domain Group	Everything in the Domain Group
A project (selected by clicking the Modeler's background)	Everything in that project

Lines are straightened only if they are less than 20 pixels from a north, west, south, or east alignment. The intent of this operation is to quickly nudge lines that are almost straight, so that they become perfectly straight.

This nudging removes the tedium of meticulously dragging items into perfect alignment and being concerned with the pixels. If a line isn't almost straight, it is left alone. In fact, the Straighten Connection operation is disabled unless the selected items qualify to be straightened. If some of the selected items qualify but others don't, the operation is still enabled, but only eligible lines are straightened.

3.10.9 Reconnecting

To reconnect components, do one of the following:

- Drag the end of a line (driver) from one application to another.
- Drag the end of a line (driver) from one driver set to another.

3.10.10 Driver Icons

Table 3-5 *Driver Icons*

Icon	Description
	A driver. The entire line between a driver set object and an application represents a driver.
	A remote driver.
	A firewall. Indicates that the driver is communicating across a firewall.

The following figure illustrates these icons.

Figure 3-29 *Driver Icons*

To see, turn on, or turn off driver icons:

1. Right-click a driver line.
2. Select an option (for example, *Mark as Firewall*) to turn on or turn off.

3.10.11 Selected Drivers

As you move the mouse over a driver, the line thickens so that it is more obvious. You can click and interact with this line.

3.10.12 Auto-Layout of Imported Objects

When you import objects from the directory, they are automatically laid out, connected with lines, and assigned an icon that matches objects and relationships as closely as possible.

For example, if you import a Driver Set object, Designer imports all of the drivers and connects them with lines. Also, each driver points to an application icon. Application icons include the following:

- The exact Application icon (for example, Avaya or PeopleSoft)
- The image stored on the driver

  The image is embedded in a square application icon.
- A generic application icon

  If no image is stored on the driver, Designer supplies an icon for one of the following applications:

  - Generic
  - JDBC
  - LDAP
  - Delimited Text

  The auto-layout mechanism uses the layout topology that you have selected. The default is Fan Out - Bottom. You can customize this setting in Preferences. See "Modeler" on page 469.

3.10.13 Keyboard Support for Connections

Table 3-6 *Shortcut Keys for Connections*

Keystroke	Description
/	Navigates to the item's next connection
\	Navigates to the item's previous connection

3.11 Aligning and Laying Out Components

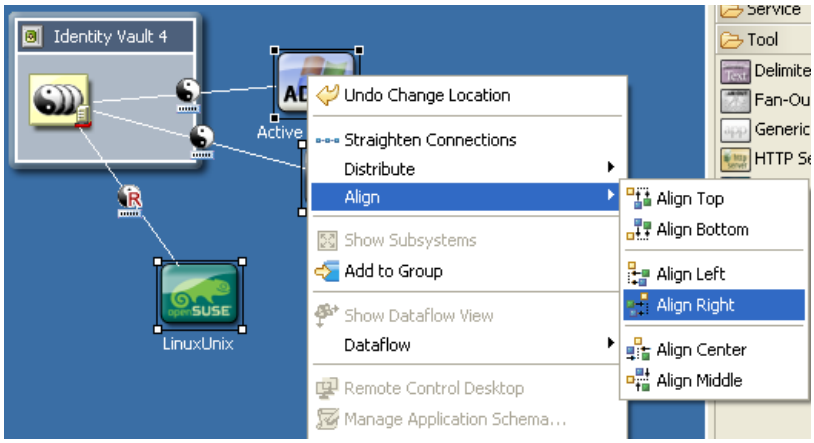
- Section 3.11.1, "Alignment Hints," on page 62
- Section 3.11.2, "Using Rulers," on page 63
- Section 3.11.3, "Using a Grid," on page 65
- Section 3.11.4, "Distributing Applications," on page 66
- Section 3.11.5, "Auto-Layouts," on page 67
- Section 3.11.6, "Layouts to Use for Imports," on page 67

Alignments place objects in the same horizontal or vertical plane. Alignments help you see relationships in your model. You can align or attach items to the left, center, or right of alignment guides.

When you move the guide, attached items move with it, staying attached in the same relative positions.

To align components:

1. Press Ctrl, then select more than one item.



2. Right-click, then select *Align*.
3. Select an alignment option.

You can also attach an item by dragging it to a guide. After you wait a moment, the guide line is highlighted, indicating that the item is attached.You can align within the same group but not across groups.

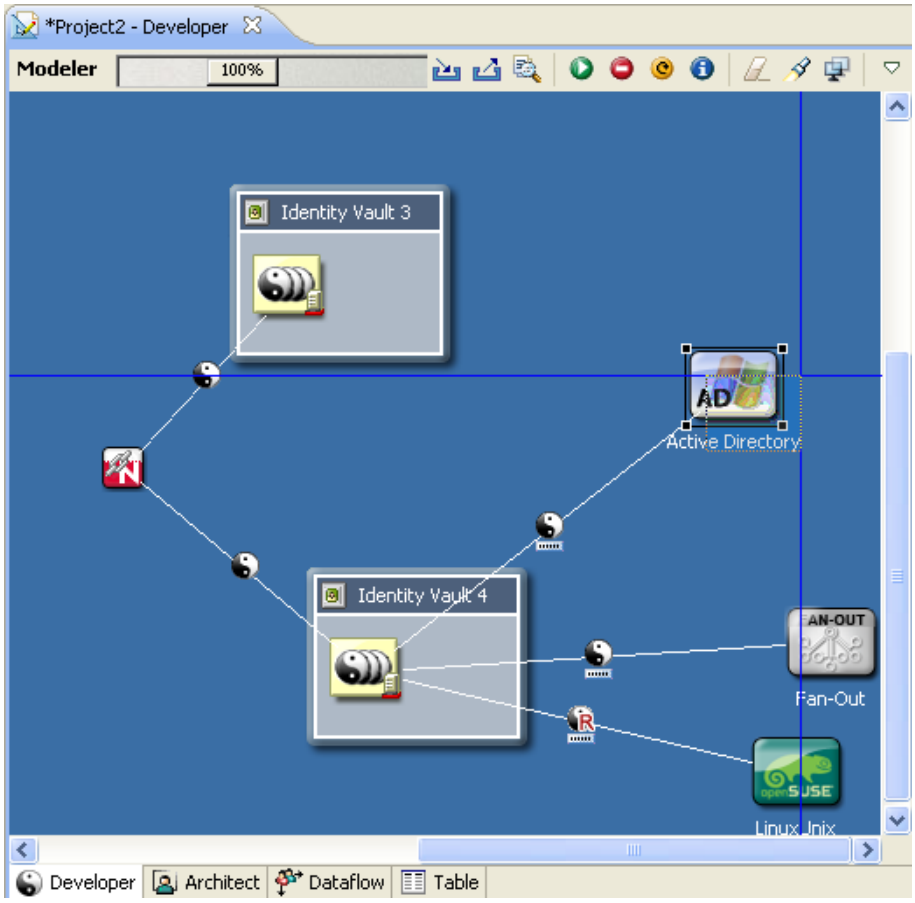
Guides that you set up are restored the next time that you run Designer. You don't need to re-create them.

Also, the alignments and attachments (left, center, or right) are stored in the project on a per-item basis, so that they are also restored.

3.11.1 Alignment Hints

Horizontal and vertical "hint" lines can be set to automatically show as you drag items into vertical or horizontal alignment with neighboring items by clicking *View* > *Alignment Hints*.

Figure 3-30 *Alignment Hints*



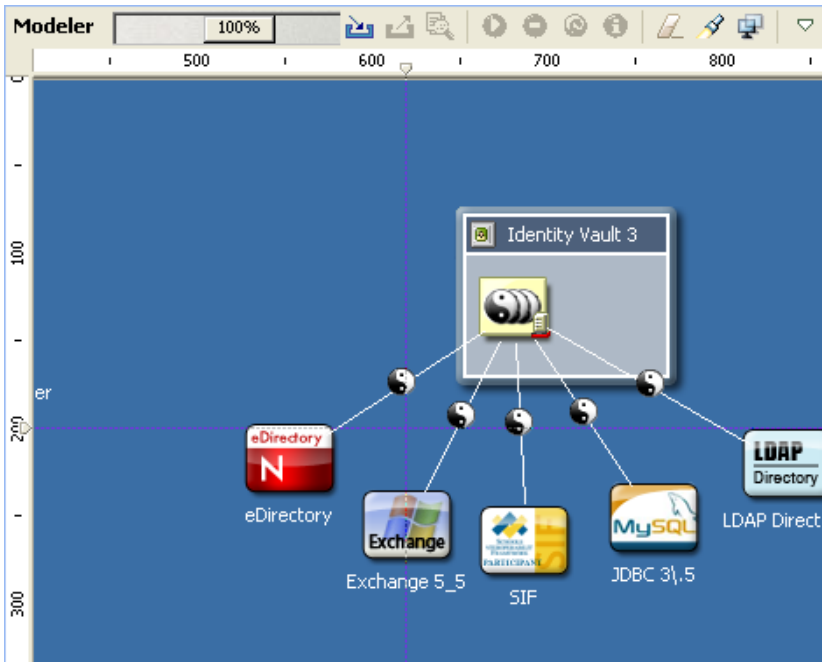
The Alignment Hints feature is off by default. To turn it on, click *View* > *Alignment Hints*.

Horizontal and vertical hint lines automatically show as you drag items into vertical or horizontal alignment with neighboring items.

3.11.2 Using Rulers

To turn on the horizontal and vertical rulers:

1. Click the Modeler space to make it active.
2. Click *View* > *Rulers*.



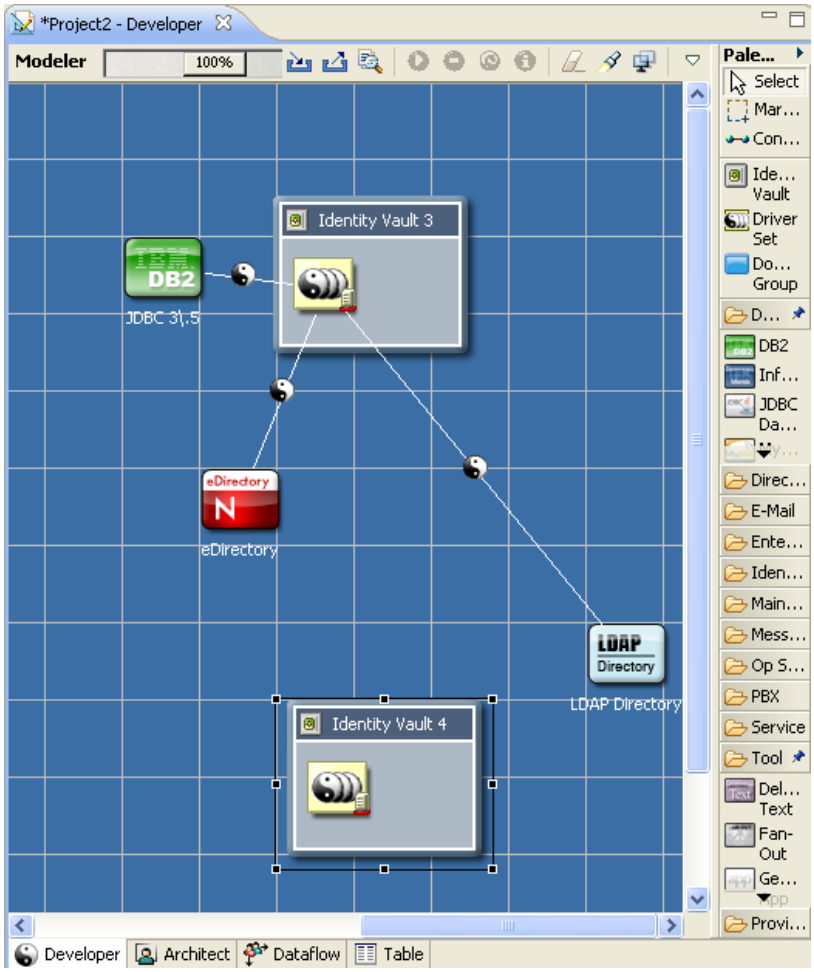
To create a guide (line), click either ruler.

To anchor items to a guide, drag the items in the model to the line.

To simultaneously move all anchored items, drag the line.

3.11.3 Using a Grid

Figure 3-31 *The Modeler's Grid*



When the grid is on, the snap-to-grid functionality is on.

To turn grid lines on and off:

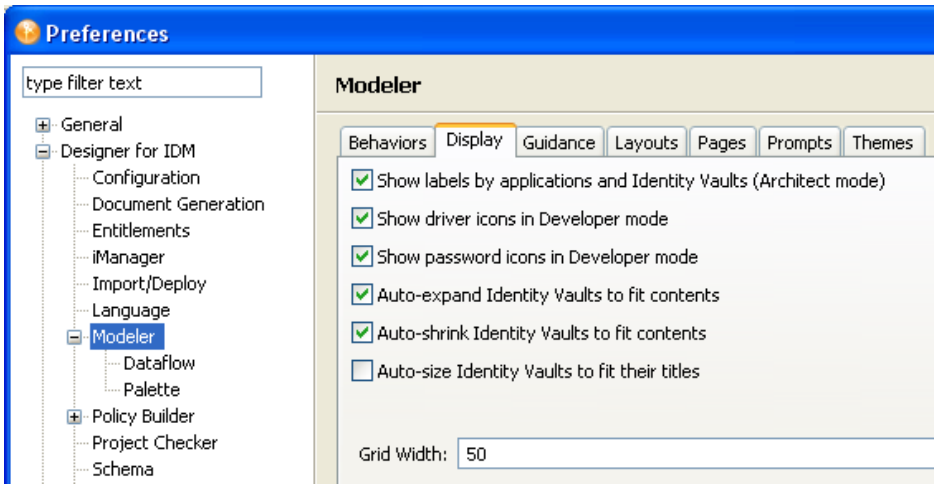
1. Click the Modeler, so that the Modeler is the active view.
2. Click *View* > *Grid*.

To coerce objects to not align with the grid, temporarily turn off snap-to-grid by holding down the Alt key. (Linux doesn't support this functionality.)

To constrain items to north-south or east-west coordinates, press Shift while dragging the items.

To change the grid size:

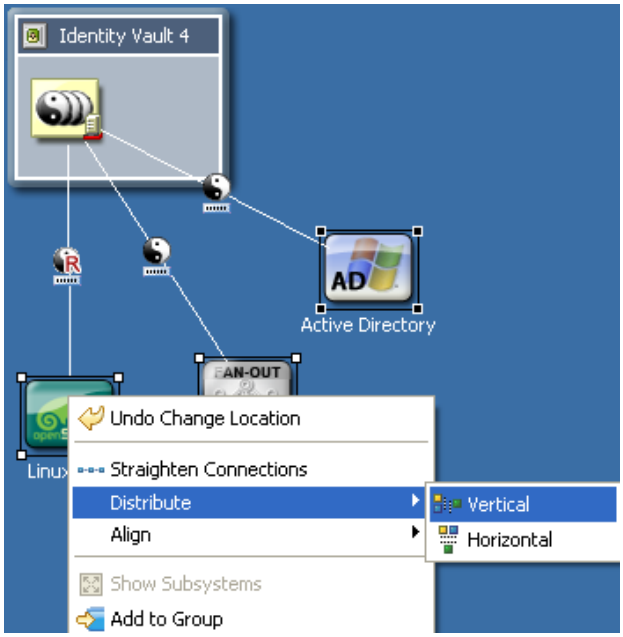
1. Click *Window* > *Preferences* > *Novell* > *Identity Manager* > *Modeler* > *Display*.



2 Type a value in the *Grid Width* field.

3.11.4 Distributing Applications

Figure 3-32 *Distributing Applications*



To equally distribute (space) applications horizontally or vertically:

1 Press Ctrl, then select three or more items.

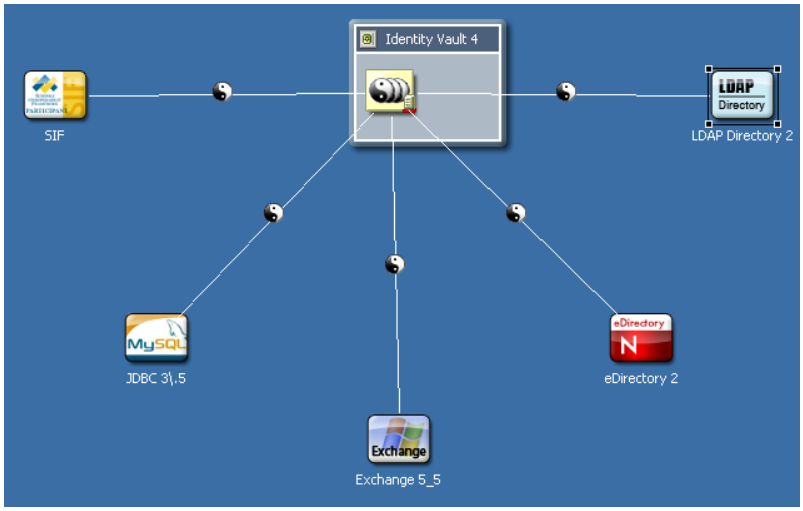
2 Right-click, then select *Distribute*.

3 Select a distribution (for example, *Vertical*).

3.11.5 Auto-Layouts

Designer ships with a number of predefined layout topologies: circle, half-circle, star, box, and different fan-out layouts.

Figure 3-33 *A Half-Circle Layout*



These layouts are set on a per-driver-set basis. Therefore, each driver set can have its own layout.

To select a layout:

1. Right-click a driver set, then select *Arrange Applications*.
2. Select an arrangement (for example, *Fan Out - Left*).

   If your model has an incorrect layout, the layout options are dimmed.

After you set a layout, applications that you connect will automatically snap into that layout. Certain connected objects (for example, multi-driver connections, eDir-to-eDir connections, and applications that are connected but reside in a different Domain Group) are ignored. They aren't included in the layout, and they don't disturb it.

An option on the Arrange Applications submenu on the Modeler's context menu enables you to expand or contract the layout arrangement. This option makes all spokes of the layout longer or shorter when you drag a slider.

3.11.6 Layouts to Use for Imports

To specify what layout to use on new driver sets that you import:

1. Select *Window* > *Preferences* > *Novell* > *Identity Manager*.
2. Click *Modeler* > *Layouts*.
3. Select an arrangement (for example, *Half Circle*), then click *OK*.

3.12 Editing Multiple Objects

You can open multiple objects and edit them at the same time. These objects must be of the same type (for example, policies).

To find out whether you can edit an object, right-click it. If *Edit* displays among the menu items, you can edit that object.

1. In the Outline view, expand the project that contains the objects that you want to edit.
2. Select the objects.
3. Right-click, then select *Edit*.
4. Edit the objects.

   You can copy and paste from one editor to another. Data must be of the same type.

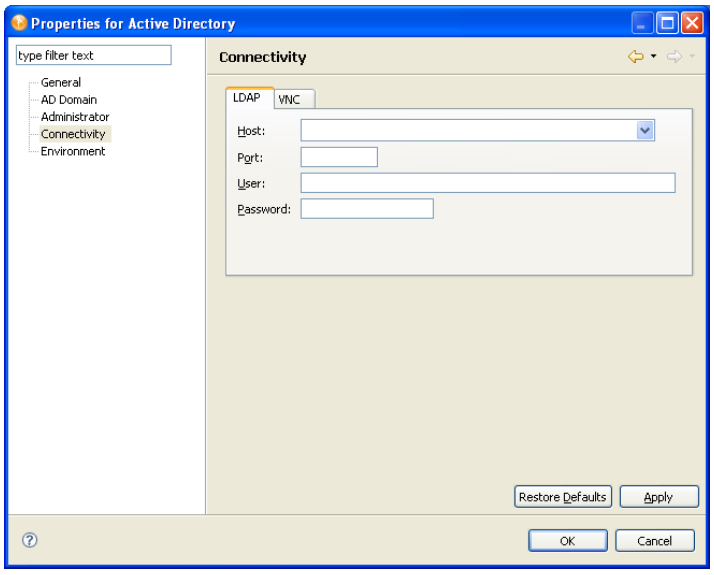
3.13 Modeling Active Directory Domain Controllers

- Section 3.13.1, “Configuring a Connection,” on page 68
- Section 3.13.2, “Discovering Controllers,” on page 69
- Section 3.13.3, “Information about Domain Controllers,” on page 69

3.13.1 Configuring a Connection

You can configure an LDAP connection to an Active Directory system so that you can discover its Domain Controllers.

1. Right-click the Active Directory application, then select *Properties* > *Connectivity*.

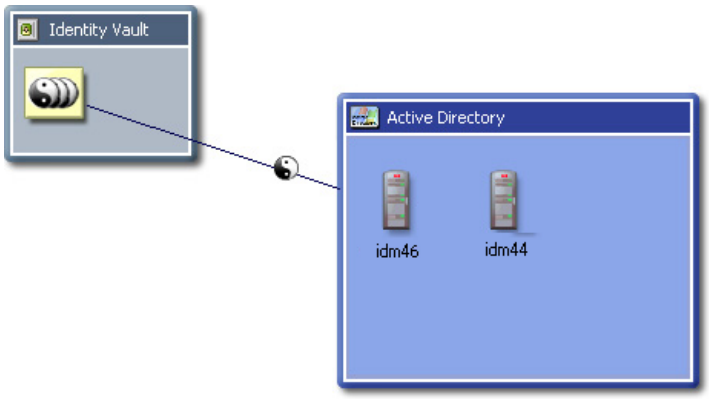


2. Complete the LDAP authentication information.

   As you tab from the *Host* field to the *User* field, Designer automatically builds a full user context. You can modify this context.

3.13.2 Discovering Controllers

1. Right-click the Active Directory application.
2. Select *Discover Domain Controllers*.

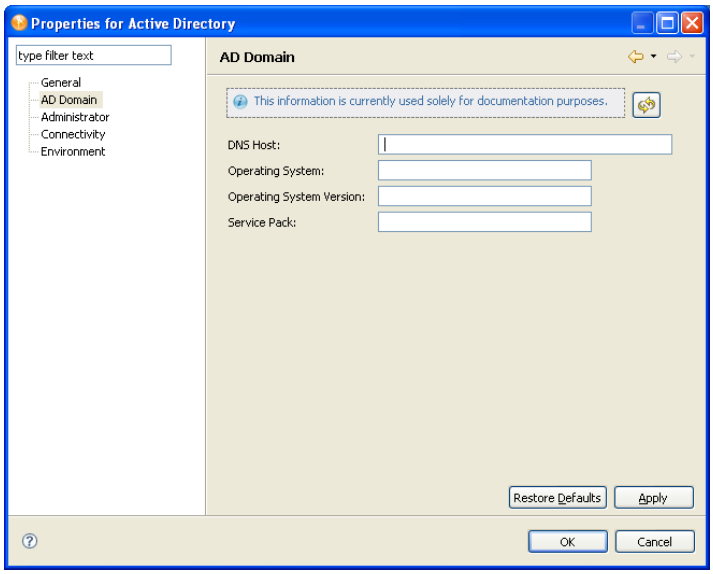


If Designer finds any controllers, it lays them out and expands the Active Directory application as a container.

3.13.3 Information about Domain Controllers

Information about each controller is loaded into the Modeler. To view this information, edit the Domain Controller object and select the AD Domain page.

Figure 3-34 *The AD Domain Page*



If the LDAP connection information is filled out, you can reread the information from that system by clicking the *Refresh* button.

3.14 Saving Your Model

To save your model, do one of the following:

- From the main menu, select *File* > *Save* (or *Save All*).
- From the main menu, select *File* > *Close* > *Yes*.
- Click the *X* in the Modeler's tab, then select *Yes*.

For more information, see The Project View (http://www.novell.com/documentation/designer30/designer_intro/data/projprojectview.html) in the *Understanding Designer for Identity Manager.*

Configuring Objects in Designer

4

Designer allows you to easily view, configure, and modify settings for Identity Vaults, driver sets, drivers, and connected systems.

- Section 4.1, “Viewing Object Properties,” on page 71
- Section 4.2, “Configuring a Domain Group,” on page 73
- Section 4.3, “Configuring Identity Vaults,” on page 74
- Section 4.4, “Configuring Servers,” on page 77
- Section 4.5, “Configuring Driver Sets,” on page 78
- Section 4.6, “Configuring Libraries,” on page 86
- Section 4.7, “Configuring Drivers,” on page 87
- Section 4.8, “Configuring Jobs,” on page 116
- Section 4.9, “Configuring ID Policy Containers,” on page 117
- Section 4.10, “Configuring ID Policies,” on page 118
- Section 4.11, “Configuring Application Properties,” on page 120
- Section 4.12, “Configuring Policies,” on page 125
- Section 4.13, “Adding Prompts to a Driver Configuration File,” on page 126
- Section 4.14, “Editing a Notification Template Name,” on page 127
- Section 4.15, “Synchronizing Passwords,” on page 127

4.1 Viewing Object Properties

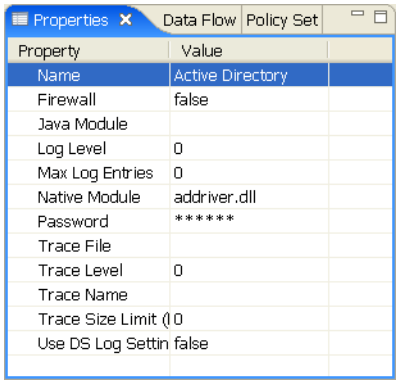
To quickly view or edit properties of items (for example, an Identity Vault or a driver), you can use the Properties view or a Properties dialog box.

- Section 4.1.1, “Properties View,” on page 71
- Section 4.1.2, “Properties Dialog Box,” on page 72
- Section 4.1.3, “Operations Relating to Properties,” on page 73

4.1.1 Properties View

If the Properties view is open when you select an item in the Modeler, information about that item displays in the Properties view. You can then quickly view or edit information.

Figure 4-1 *The Properties View*



To open the Properties view, click *Window* > *Show View* > *Other* > *General* > *Properties*. For additional information, see The Properties View (http://www.novell.com/documentation/designer35/designer_intro/data/projpropertiesview.html) in the *Understanding Designer for Identity Manager*.

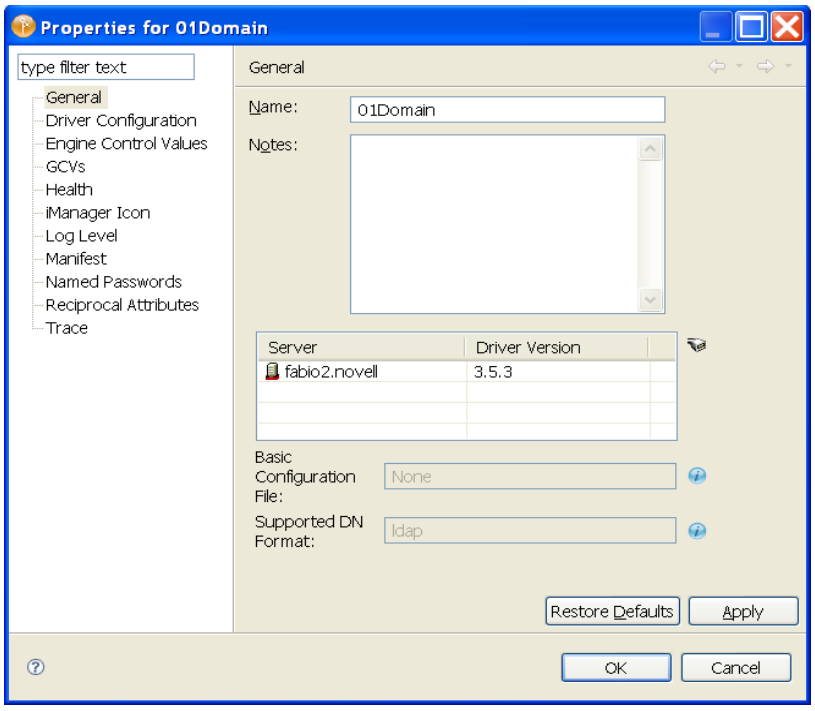
4.1.2 Properties Dialog Box

The list of property pages in the Properties dialog box is organized alphabetically across Designer with the exception og the *General* page, similar to that of Eclipse.

To view or edit properties of items:

1. Open the Properties dialog box by doing one of the following:
   - Double-click an item in the Modeler or in the Outline view.
   - Right-click an item (for example, an Identity Vault) in the Modeler or Outline view, then select *Properties*.
   - Select an item, then press *Enter*.
   - Select an item, then select *File* > *Properties*.
   - Select an item, then select *Model* > [object] > *Properties*.

   The following figure illustrates a driver's properties page:



2 Edit settings, then click *OK* to save.

4.1.3 Operations Relating to Properties

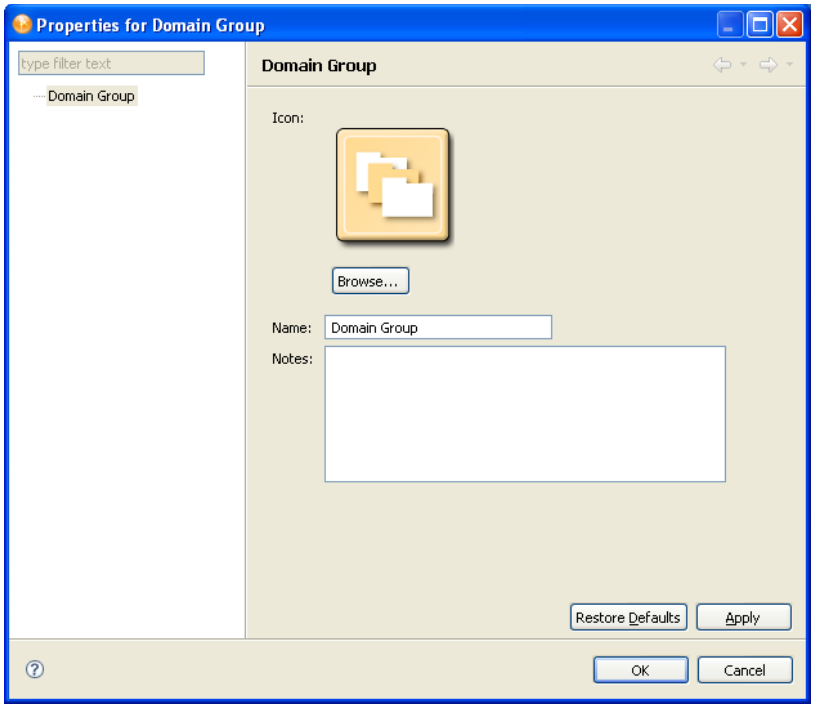
Table 4-1 *Operations Relating to Properties*

Operation	Description
Open the Properties view	Click *Window* > *Show View* > *Other* > *General* > *Properties*.
Open the Properties dialog box	Double-click an item, or right-click, then select *Properties*.
Edit settings	You can edit the settings of any item selected in the Modeler or Outline view.
View a server's properties	In the Outline view, right-click the server icon, then select *Properties*.
Save to memory or disk	When you click *Apply* or *OK* in a properties dialog box, changes are committed to memory. However, changes are not saved to disk unless you select *File* > *Save*.

4.2 Configuring a Domain Group

To view or change a domain group's settings, double-click the domain group.

Figure 4-2 *The Domain Group Property Page*



1. To change the domain group's icon, click *Browse*, then navigate to and select an image file.

   By default, the Browse button opens the `icons/group` folder in the `com.novell.designer.core` plug-in. The default image selected is `administrative.png`. To select a different image, double-click it.

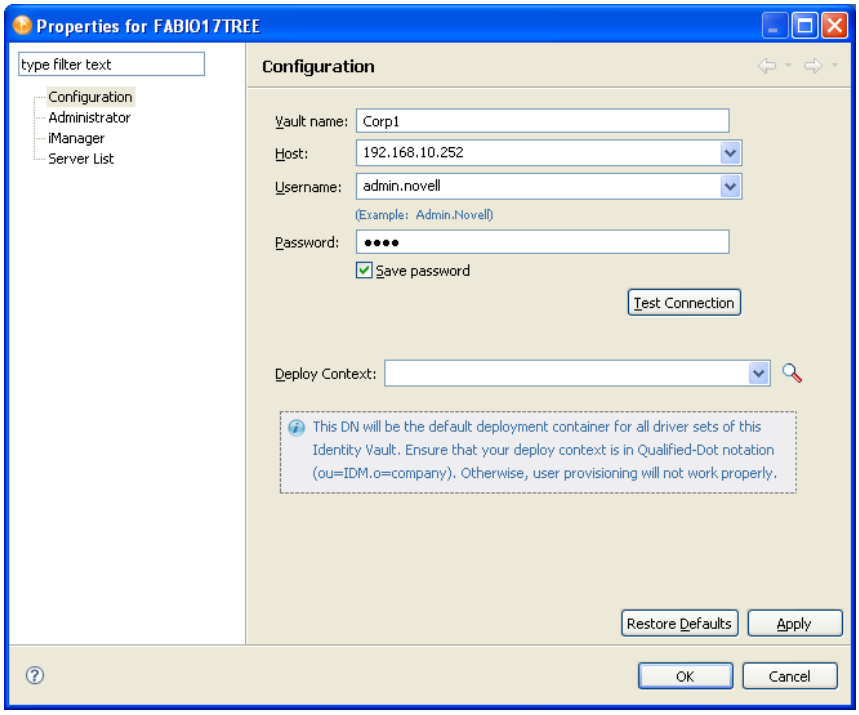
2. Click *Apply*.
3. To change the name of the domain group, edit the *Name* field.
4. Add details in the *Notes* pane.
5. Click *OK*.

   The image (for example, `administrative.png`) appears to the left of the domain name in the Modeler.

4.3 Configuring Identity Vaults

To view or change an Identity Vault's settings, double-click the ID vault object in the Outline view or the Modeler.

Figure 4-3 *Identity Vault Properties*



The Identity Vault Properties page has three options. In addition, you can configure a hostname in the `hosts` file.

- Section 4.3.1, “Configuration Settings,” on page 75
- Section 4.3.2, “Administrator,” on page 76
- Section 4.3.3, “Server List,” on page 76
- Section 4.3.4, “iManager,” on page 76
- Section 4.3.5, “Local Hostname,” on page 77

4.3.1 Configuration Settings

Table 4-2 *Configuration Settings for an Identity Vault*

Field	Description
Vault name	The name of the Identity Vault object. The default is Identity Vault.
Host	The eDirectory host where you plan to log in and deploy.
Username	The eDirectory username that has sufficient rights to make changes to objects associated with this deployment.
Password	The password for the above user.

Field	Description
Save password	Saves the password permanently, so you are authenticated into this Identity Vault each time you open Designer. If you use this option, the password is saved locally in Designer's file system and is not secure. If you do not select this option, the password is remembered only until you close Designer.
Test connection	Selecting this button allows the user to create, or, if a connection is unresponsive, to re-create a connection to the Identity Vault. If a connection has not been established to the Identity Vault, the button displays *Test connection*. After a connection is established, the button displays *Refresh connection*.
Deploy context	The default DN container assigned to all driver sets that are associated with this Identity Vault. If you specify a DN container on the Driver Set object, that setting takes precedence over the default setting.

4.3.2 Administrator

The *Administrator* option is divided into three sections. Entering information in these sections is optional.

- **Personal Information:** Lets you enter information specific to the Identity Vault, such as Name, Title, Department, and Location.
- **Contact Information:** Lets you enter information such as Email, Phone, Cell Phone, Pager, and Fax.
- **Notes:** Allows you to type any reminders you might need for future reference.

4.3.3 Server List

The *Server List* option displays the servers that are associated with the selected Identity Vault. You can add, edit, or remove the server entries.

NOTE: If you select the option to allow a default server to be created, that server shows up as Default Server.default_container in the list. You cannot deploy a driver set into an existing eDirectory tree if you have Default Server.default_container in the Server List. You must first remove this reference and add a Metadirectory server in an eDirectory tree.

4.3.4 iManager

The *iManager* option displays the URL that Designer uses to launch the Novell iManager administrative tool. You can modify this URL as needed.

To launch iManager from Designer, select *Tools* > *iManager*.

4.3.5 Local Hostname

If desired, Designer supports designating a hostname for your Identity Vault by adding an entry to the `hosts` file of your local OS. After assigning a hostname to the Host address of your Identity Vault, you can use the hostname instead of an IP address or DNS name to access the Identity Vault.

For example, if your Identity Vault has a host address of 192.168.100.254, you can associate the name *ID-VAULT* to that address in your local `hosts` file. Then, in Designer, you can refer to the Identity Vault by the name *ID-VAULT* instead of using the IP address.

For more information about using your local hosts file, consult your operating system's documentation.

4.4 Configuring Servers

1. Right-click the server icon in the Outline view.
2. Select *Properties*.

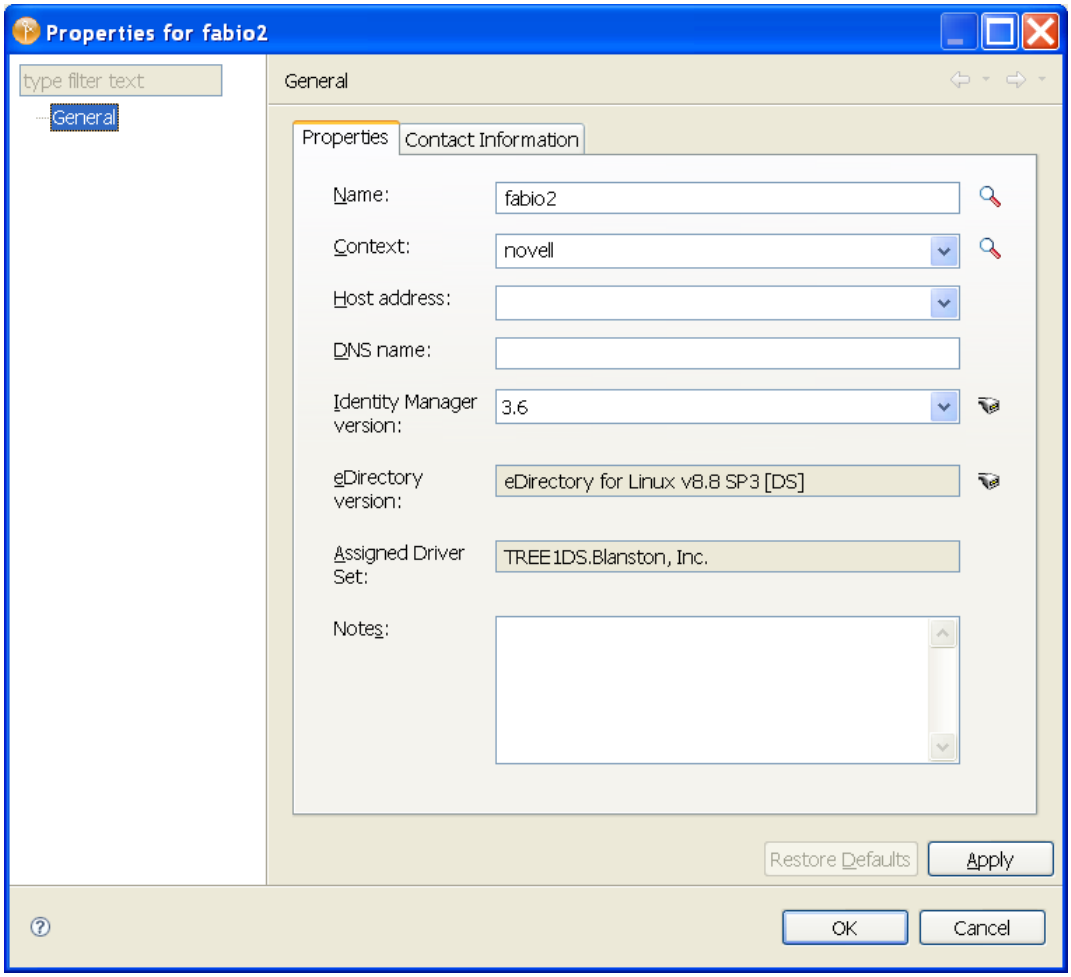


Table 4-3 on page 78 lists settings for the Server Properties page:

Table 4-3 *Settings for the Server Properties Page*

Field	Description
Name	The name of the Server object. The Identity Vault lists the server. You can browse to and select the server.
Context	The server's context. The Identity Vault assigns the context. You can browse to and select the context.
Host address	The server's IP address.
DNS name	The domain name or complete directory context name.
Identity Manager version	Section 5.2, "Changing the Identity Manager Version," on page 130The version of Identity Manager that is running on the server. The default is Identity Manager 3.6. You can change the version by using the drop-down list. See .
eDirectory version	The version of eDirectory that the server is using.
Assigned Driver Set	The driver set the server is assigned to.
Notes	Information that you want to specify, to help you maintain the server.

Use the Contact Information tab to provide information on the person to contact and other items of interest concerning the server.

4.5 Configuring Driver Sets

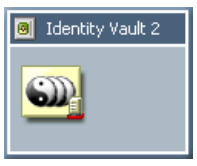
A driver set is a container that holds Identity Manager drivers. Only one driver set can be active on a server at a time. As a result, all active drivers must be grouped into the same driver set. To view or change settings, double-click a driver set in the Modeler.

- Section 4.5.1, "Driver Set General Options," on page 78
- Section 4.5.2, "Driver Set Global Configuration Values," on page 80
- Section 4.5.3, "Java Environment Parameters," on page 80
- Section 4.5.4, "Driver Set Log Levels," on page 81
- Section 4.5.5, "Driver Set Named Passwords," on page 82
- Section 4.5.6, "Driver Set Server List," on page 82
- Section 4.5.7, "Driver Set Trace," on page 83

4.5.1 Driver Set General Options

When you create an Identity Vault, a driver set is added to the vault by default.

Figure 4-4 *A Driver Set in an Identity Vault*



You can add other driver sets by dragging the Driver Set object from the palette to the Modeler.

From the *General* page, you can specify or change driver set values.

Figure 4-5 *Driver Set Properties*

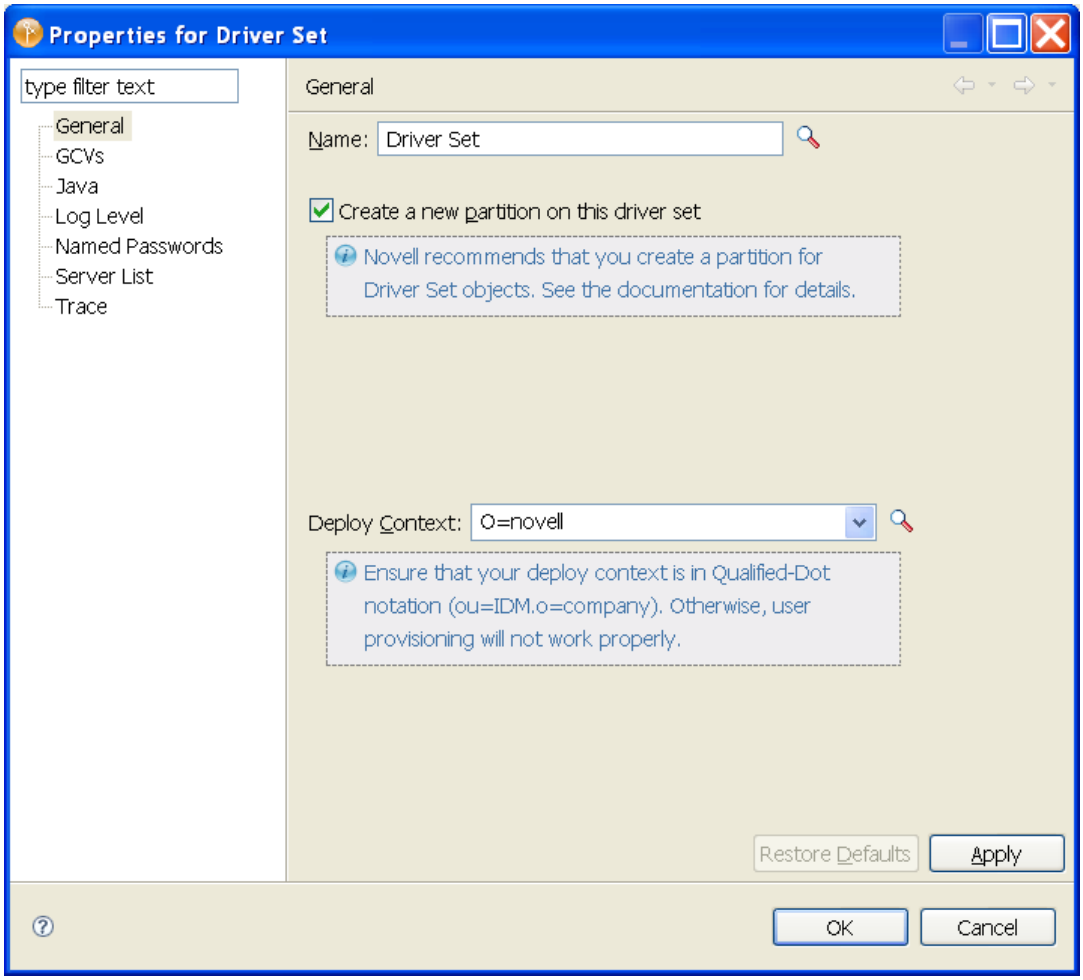


Table 4-4 *Driver Set Settings*

Field	Description
Name	The name of the Driver Set object (for example, DriverSet1.)
Create a new partition on this driver set	We recommend that you select this option. For details, see "Planning the Technical Details of your Identity Manager Implementation" (http://www.novell.com/documentation/idm/install/data/alorukq.html#alorukq) in the *XYZ Guide*.

Field	Description
Deploy context	The Identity Vault assigns the default DN container value to all driver sets. If you specify a DN container here on the Driver Set object, that setting takes precedence over the Identity Vault setting. You can manually enter this value or browse for it.

4.5.2 Driver Set Global Configuration Values

Global configuration values (GCVs) are settings that are similar to driver parameters. Global configuration values can be specified for a driver set as well as an individual driver. If a driver does not have a GCV, the driver inherits the value for that GCV from the driver set.

GCVs allow you to specify settings for Identity Manager features such as password synchronization and driver heartbeat, as well as settings that are specific to the function of an individual driver configuration. Some GCVs are provided with the drivers, but you can also add your own. You can refer to these values in a policy to help you customize your driver configuration.

To view or change the driver set's GCV settings, double-click the driver set. From the Global Configuration Values page, you can add, edit, or remove values, or edit the XML file for the driver set. To select a value, click on the value or the control field to the right of the value’s name. Use the Add, Edit, Remove, and Edit XML buttons at the bottom of the page.

For more information on global configuration values, see Section 4.7.2, “Driver Global Configuration Values,” on page 89.

4.5.3 Java Environment Parameters

The *Java Environment Paramaters* enable you to configure the Java virtual machine (JVM) on the Metadirectory server asociated with the driver set. The parameters apply only if the driver set is associated with a Metadirectory server that is version 3.6 or newer.

Table 4-5 *Java Environment Parameters Settings*

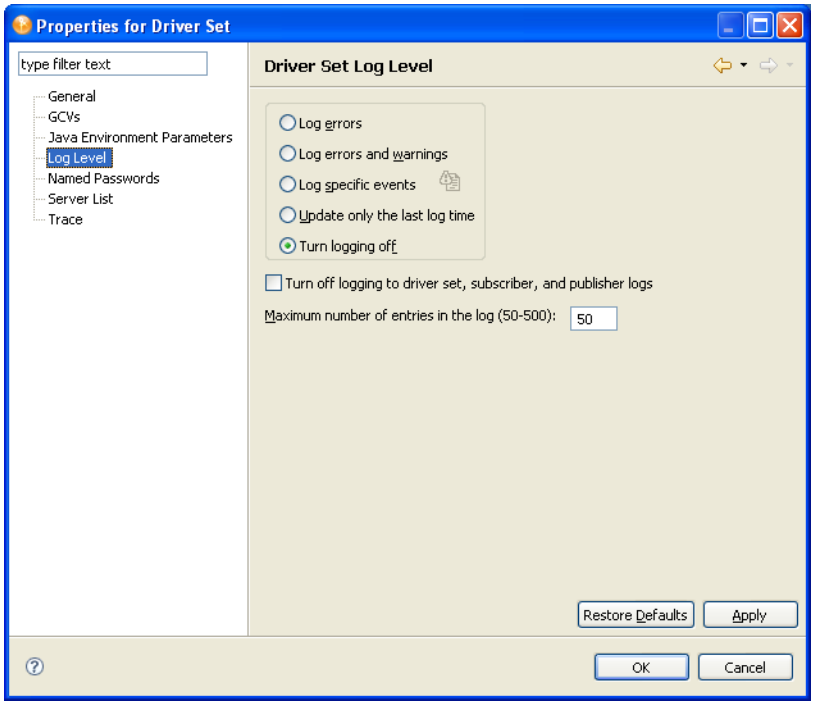
Field	Description
Classpath Additions	Specifies additional paths for the JVM to search for package (`.jar`) and class (`.class`) files. Using this parameter is the same as using the `java -classpath` command. When entering multiple class paths, separate them with a semi colon (;) for a Windows JVM and a colon (:) for Unix/Linux JVMs.
JVM Options	Specifies additional options to use with the JVM. Refer to your JVM documentation for valid options.

Field	Description
Initial Heap Size	Specifies the initial (minimum) heap size available to the JVM. Increasing the initial hap size can improve startup time and performance. Enter a numeric value followed by g, m, or k (case insensitive). If no letter size is specified, the size defaults to bytes. Using this parameter is the same as using the `java -Xms` command. Refer to your JVM documentation for information about the JVMs default initial heap size.
Maximum Heap Size	Specifies the maximum heap size available to the JVM. Enter a numeric value followed by g, m, or k (case insensitive). If no letter size is specified, the size defaults to bytes. Using this parameter is the same as using the `java -Xmx` command. Refer to your JVM documentation for information about the JVMs default maximum heap size.

4.5.4 Driver Set Log Levels

The *Driver Set Log Level* options enable you to view high-level information. For lower-level information, use the *Trace* option.

Figure 4-6 *Driver Set Options for Logging*



By default, logging is turned off. To track errors, messages, or events, change the default.

1. Double-click the driver set.
2. Select *Driver Set Log Level.*
3. Select a logging option.

The log option that you select determines which messages are available in the log.

4 To configure audit instrumentation, select *Log specific events*, click the event selector button, select events, then click *OK*.

   The *Update only the last log time* option updates the time stamp to indicate the last activity of the driver.

5 Specify the number of entries in the log.

   The default is 50 entries (lines) in the log. If you want a longer history, increase the number.

6 Save changes by clicking *OK*.

The driver set log contains messages from the engine when it tries to start or stop drivers. To view the log, use iManager. Select the *Status Log* icon above the Identity Vault in the *Identity Manager Overview*.

4.5.5 Driver Set Named Passwords

The Named Passwords property page allows you to manage (add, edit, delete) named passwords for the selected driver set. When defined in the driver set, named passwords are available to all drivers in the driver set.

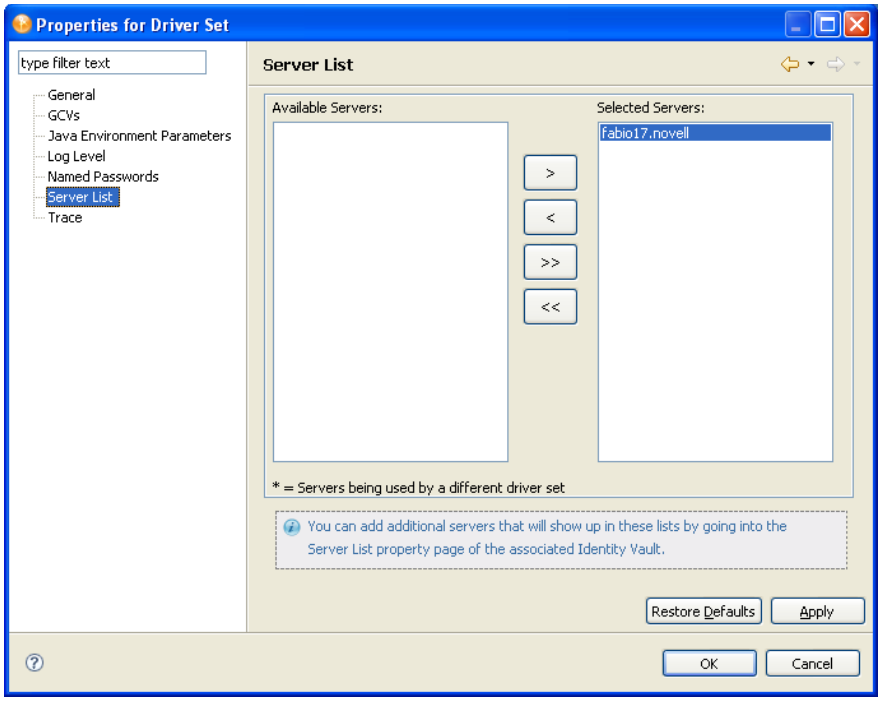
NOTE: If you create a named password of the same name in both the driver set and a driver in the driver set, the named password settings in the driver take precedence.

You can define named passwords on both drivers and driver sets. For more information about named passwords, see Section 4.7.6, “Driver Named Passwords,” on page 99.

4.5.6 Driver Set Server List

After adding one or more servers to the Identity Vault, you can view or change the driver set’s server association.

Figure 4-7 *Server List*



Select a server in the *Available Servers* list, then use the arrows to move the server to the *Selected Server* list. If a server is not in the *Available Servers* list, you must first add it by editing the Identity Vault properties. See Section 4.3, “Configuring Identity Vaults,” on page 74.

4.5.7 Driver Set Trace

Although a driver set has nothing to trace, you can add a trace level to a driver set. The *Trace* setting specifies a trace level used with all drivers associated with the driver set.

With the trace set, DS Trace displays Identity Manager and DirXML events as the engine processes the events. The trace level affects each driver in the driver set. Use the trace level for troubleshooting issues with the drivers when they are deployed. DS Trace displays the output of the specified trace level.

WARNING: You should use the trace level only for testing or for troubleshooting driver issues. Setting a driver trace level on a production driver can cause Identity Manager server to process events slowly.

To set a driver set’s trace characteristics:

1. In the Outline view or Modeler, right-click the driver set, then select *Properties*.
2. In the driver properties, select *Trace* in the left navigation area.
3. On the Trace page, specify the trace settings for the driver set, then click *OK*.

Table 4-6 *Driver Set Trace Settings*

Field	Description
Trace level	The IDM engine supports the following trace levels: ◆ Trace level 0: Displays fatal messages, errors, warnings and successes. ◆ Trace levels 1: Displays informational messages in addition to the information from Trace level 0. ◆ Trace level 2: Displays contents of XML documents in addition to the information from Trace level 1. ◆ Trace level 3: Displays policy information in addition to the information from Trace level 2.
XSL Trace Level	DS Trace displays XSL events. Set this trace level only when troubleshooting XSL style sheets. If you do not want to see XSL information, set the level to 0.
Java Debug Port	Allows developers to attach a Java debugger.
Trace File	When a value is set in this field, all Java information for the driver is written to file. The value for this field is the path for that file. As long as the file is specified, Java information is written to this file. If you do not need to debug Java, leave this field blank.
Trace File Encoding	The trace file uses the system's default encoding. You can specify another encoding if desired. This option is only available if the driver set is associated with a server that is version 3.6 or newer.
Trace File Size Limit	Sets a limit for the Java trace file. Select *Unlimited* to allow the file to grow to fill the disk.

The following methods help you capture and save Identity Manager trace information.

- "Windows" on page 84
- "UNIX" on page 85
- "iMonitor" on page 85

Windows

Open the Control Panel, select *NDS Services*, then click *DS Trace.DLM* > *Start*. A window named NDS Server Trace Utility opens.

To set the filters to capture the DirXML trace information:

1. Click *Edit* > *Options* > *Clear All*.
2. Click the boxes next to *DirXML* and *DirXML Drivers*, then click *OK*.

To save the information to a file:

1. Click *File* > *New*.

   A dialog box prompts for a filename.

2. Enter a filename with the extension of `.log`.
3. To stop capturing information, click *File* > *Close*.

   The file is saved.

UNIX

Use the `ndstrace` command at the console to display the Identity Manager events. The `exit` command quits the trace utility.

Table 4-7 *ndstrace Commands*

Command	Description
`Set ndstrace=nodebug`	Turns off all trace flags.
`Set ndstrace on`	Displays trace messages to the console.
`Set ndstrace file on`	Captures trace message to the `ndstrace.log` file in the `/var/nds` directory.
`Set ndstrace file off`	Stops capturing trace messages to the file.
`Set ndstrace=+dxml`	Displays the Identity Manager events
`Set ndstrace=+dvrs`	Displays the Identity Manager driver events.

iMonitor

Use iMonitor to get DS Trace information from a Web browser.

Table 4-8 *Platforms and Commands for Web Browsers*

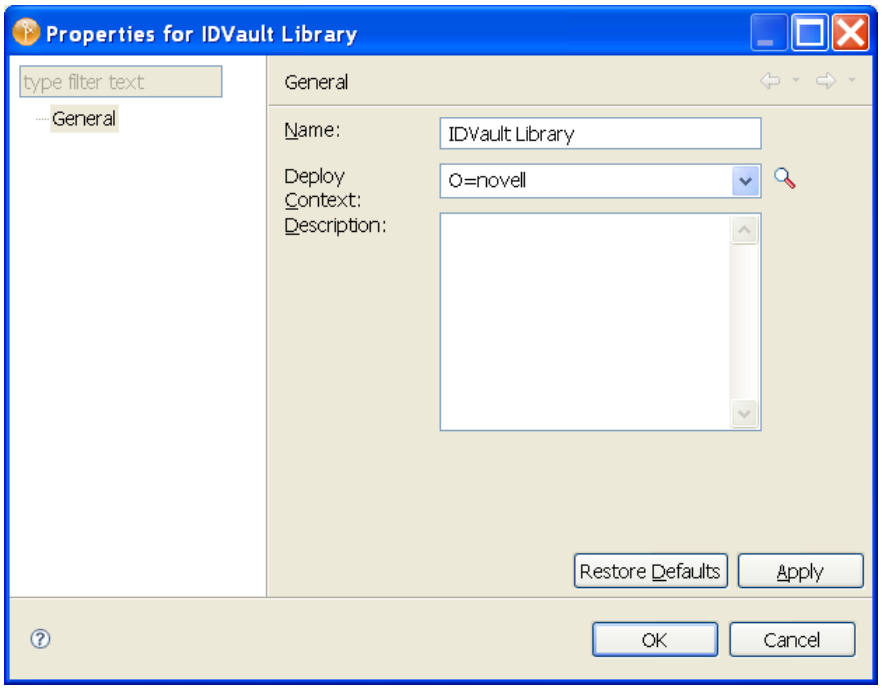
Platform	Command
Windows	`ndsimon.dlm`
Linux/Solaris/AIX/HP-UX	`ndsimonitor`

1. Access iMonitor from http://server_ip:8008/nds (the default port).
2. Click *Trace Configuration*.
3. Click *Clear All*.
4. Click *DirXML* and *DirXML Drivers*.
5. Click *Trace On*, then click *Trace History*.
6. Click the *Current* document icon to view the live trace.

4.6 Configuring Libraries

The Library object is a repository of commonly used policies that can be referenced from multiple locations. You can place a policy in the library that every driver in the driver set can reference. You can find the Library object in the Outline view.

1 Right-click the Library object and select *Properties* to view its Properties page.



The following table lists settings for the Library Properties page:

Field	Description
Name	The name of the library. You can modify the name to be more descriptive, especially if you have more than one library in a tree. For example, you might have one library at the Identity Vault level containing policies that are generic to most drivers, and another library at the Driver Set level containing policies that are specific to that driver set.
Deploy Context	The Identity Vault assigns the default DN container value to a library created or deployed at the Identity Vault level. If you specify a DN container here on the Library object, that container setting takes precedence over the Identity Vault setting. You can manually enter this value or browse to and select the context. Libraries created under the driver set do not have the Deploy Context option.
Description	This field allows you to type a description of the selected library.

2 After you have modified the fields, select *Apply* to apply the new settings, then click *OK* to close the Properties page.

For more information on what you can add to a library, see Library Objects (http://www.novell.com/documentation/idm36/policy_designer/data/rlibrary.html) in the *Policies in Designer 3.5*.

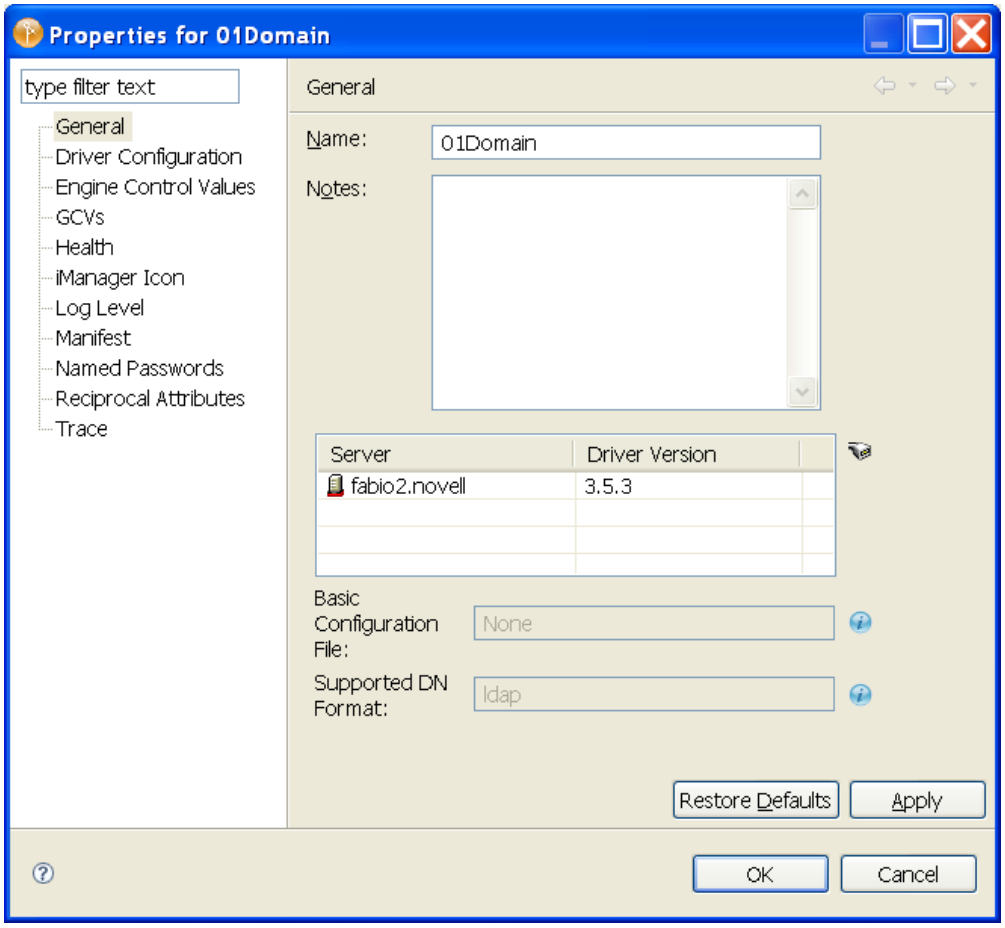
4.7 Configuring Drivers

A driver provides the connection between an application and the Identity Vault. The driver is the connector that enables data synchronization and sharing between systems. To view or change settings, double-click a driver or driver line in the Modeler.

- Section 4.7.1, “Driver General Settings,” on page 88
- Section 4.7.2, “Driver Global Configuration Values,” on page 89
- Section 4.7.3, “Driver Log Level,” on page 95
- Section 4.7.4, “Driver Configuration,” on page 96
- Section 4.7.5, “Driver Manifest,” on page 99
- Section 4.7.6, “Driver Named Passwords,” on page 99
- Section 4.7.7, “Driver iManager Icon,” on page 100
- Section 4.7.8, “Driver Trace Levels,” on page 100
- Section 4.7.9, “Engine Control Values,” on page 103
- Section 4.7.10, “Driver Health Configuration,” on page 105
- Section 4.7.11, “Reciprocal Attributes,” on page 113

4.7.1 Driver General Settings

Figure 4-8 *The Driver's General Property Page*



NOTE: Driver versions vary for each driver.

Table 4-9 *The Driver's General Settings*

Field	Description
Name	Displays the driver name, which you can change.
Notes	Enables you to type notes about your driver implementation.
Server/Driver Version	Displays the server name to which driver is associated. The driver version only shows if the driver is running. Driver versions vary for each driver.

Field	Description
Basic configuration file	Displays the configuration filename that this driver uses. Contains the filename of the configuration file that was used during import. To view the path to this file, click the information icon next to the filename. You might want to view the file to find out version information. If you haven't yet run the import wizard, this field is set to *None*.
Supported DN format	Displays the format (for example, LDAP) that is supported for each driver. This DN information is important for policy building and simulation. For additional details, click the information icon next to the format field.

The driver property page is dynamic. Labels and descriptions are dynamically read from the driver configuration file for the Authentication ID and connection information

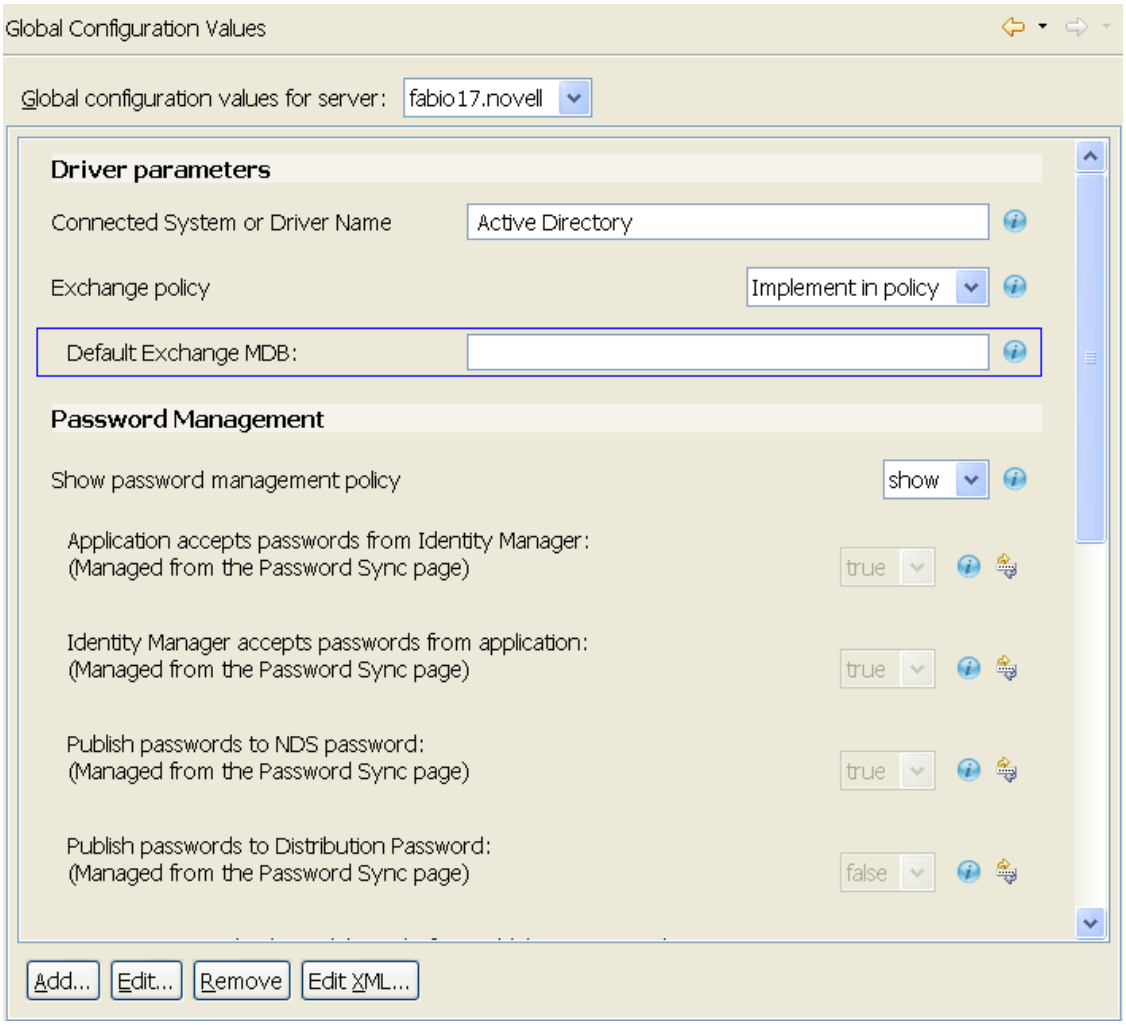
4.7.2 Driver Global Configuration Values

Global configuration values (GCVs) are settings that are similar to driver parameters. GCVs can be specified for an individual driver as well as a driver set. If a driver does not have a GCV, the driver inherits the value for that GCV from the driver set.

GCVs allow you to specify settings for Identity Manager features such as password synchronization and driver heartbeat, as well as settings that are specific to the function of an individual driver configuration. Some GCVs are provided with the drivers, but you can also add your own. You can refer to these values in a policy to help you customize your driver configuration.

To edit the driver set's GCV settings, double-click the Driver Set object in the Modeler view. From the Global Configuration Values page, you can add, edit, remove, or edit the XML for GCVs.

Figure 4-9 *The Global Configuration Values Page*



To view or change the driver's GCV settings, double-click the driver. From the Global Configuration Values page, you can add, edit, or remove values, or edit the XML file for the driver. To select a value, click on the value or the control field to the right of the value's name. Use the *Add*, *Edit*, *Remove*, and *Edit XML* buttons at the bottom of the page.

You can add, edit, and remove GCVs on the Global Configuration Values page, except for those values found under the Password Management heading. Password values are accessed through the Password Synchronization page; click the *Launch Password Sync Dialog* icon to the right of the control field and the *Information* icon.

In Identity Manager 3.6, GCVs play a much bigger role in configuring drivers. The two required options for configuring a driver are *Driver Configuration* and *GCVs*. However, because each driver contains different values and parameters, you need to consult the driver manual for specific values. Go to the Identity Manager Drivers Web site (http://www.novell.com/documentation/idm36drivers/index.html), then select the manual for the driver you are configuring.

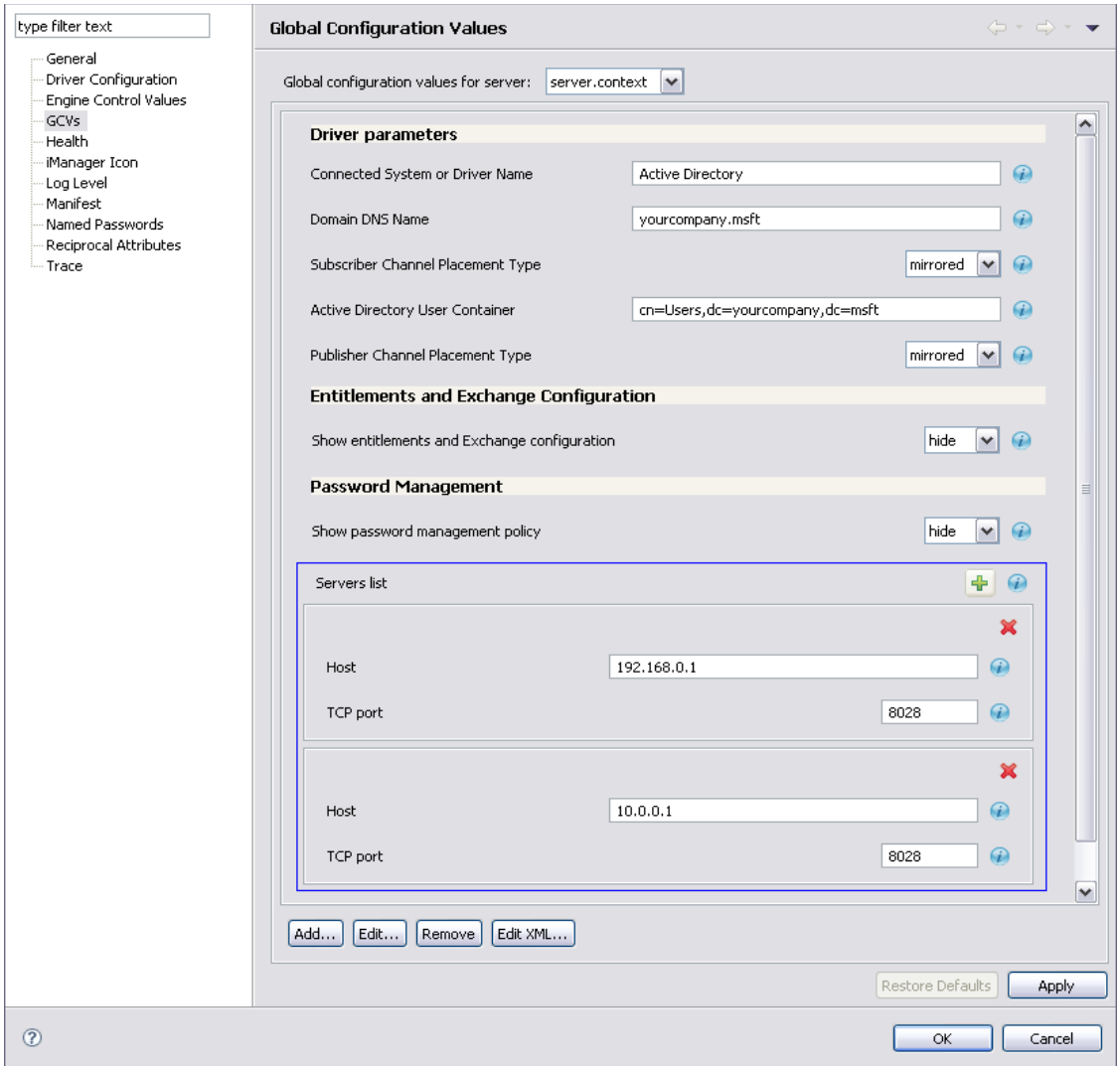
- "Structured Global Configuration Values" on page 91
- "Adding or Editing a Structured GCV Element" on page 92

Structured Global Configuration Values

Structured global configuration values allow you to place GCVs into a group. You do this by creating a template containing the GCVs that you want in the group, along with their default values. Then, you create instances of the template with the values you want to change. You can create instances by selecting a Structured GCV and clicking *Add*, then modifying the values.

NOTE: Structured GCV is supported with Identity Manager 3.6.1 and above.

Figure 4-10 *Structured Global Configuration Values Page*



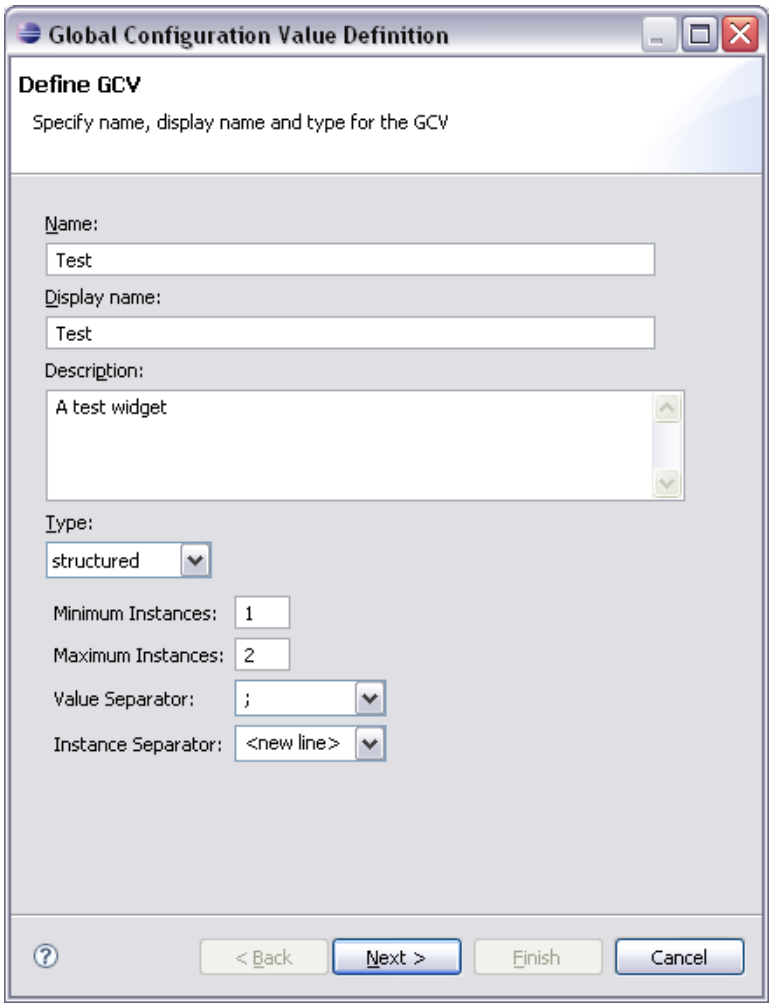
Icon	Description
	Adds an instance to the Structured GCV template.
	Deletes an instance from the Structured GCV template.

Save the changes by clicking *Apply*, then click *OK*.

Adding or Editing a Structured GCV Element

1 To open the Global Configuration Values Definition Wizard, click *Add/Edit* on the Global Configuration Values page.

Figure 4-11 *Define GCV Page*



2 Specify the values for different fields in the Define GCV page:

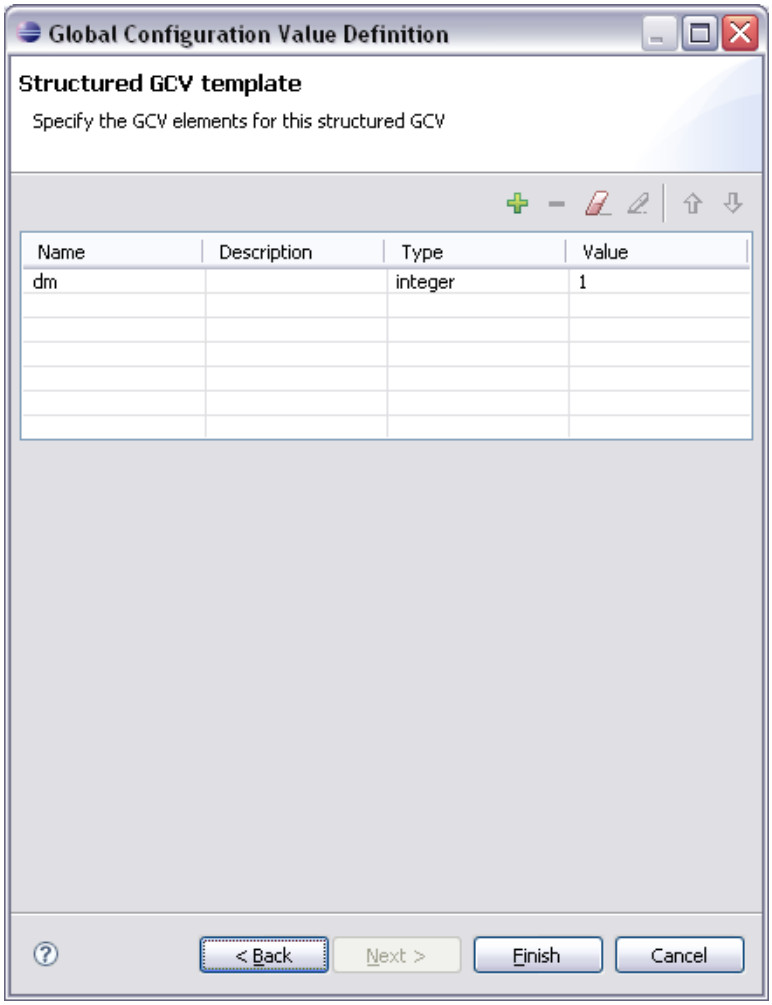
Field	Description
Name	Specify the name for the GCV. The name should be unique for each GCV.
Display name	Specify the display name of the GCV. It can be same as the name of the GCV.
Description	Provide a brief description of the GCV.
Type	Select *structured* from the *Type* drop down list.

Field	Description
Minimum Instances	When you add an instance, the GCVs in the instance take the default values. However, you can define the number of instances a GCV should contain. The UI enforces the restrictions. Specify the value for minumum number of instances you want. By default, the minimum number of instances is '0'.
Maximum Instances	When you add an instance, the GCVs in the instance take the default values. However, you can define the number of instances a GCV should contain. Specify the value for maximum number of instances you want. By default, the maximum number of instances is 'unlimited'.
Value Separator	Define a value separator for a GCV. A value separator acts as a separator between the instance value strings when the structured value is output as a single string. For example: If "one", "two", and "three" are the instance items, and ";" is the value separator, the following is the unstructured output of a single instance: "one;two;three"
Instance Separator	Define an instance separator for a GCV. An instance separator acts as a separator between the instances when the structured value is output as a single string. For example: If "one;two;three" and "four;five;six" are the instance strings and "
" is the instance separator (a linefeed character), the following is the unstructured output: "one;two;three four;five;six"

3 Click *Next*.

The Structured GCV template page displays a list of GCV elements in a table.

Figure 4-12 *Structured GCV Template Page*



4 Use the toolbar to add, edit, or remove GCV elements from the template table.

A GCV element is displayed as a row in the table. You can specify a value for it by clicking the cell in the *Value* column. Other fields of the table are automatically populated.

Icon	Description
	Adds a GCV element to the Structured GCV template.
	Allows you to edit a GCV element in the Structured GCV template.
	Removes a GCV element from the Structured GCV template.
	Deletes all GCV elements from the Structured GCV template. All elements are deleted in one operation.
	Moves the selected row up.
	Moves the selected row down.

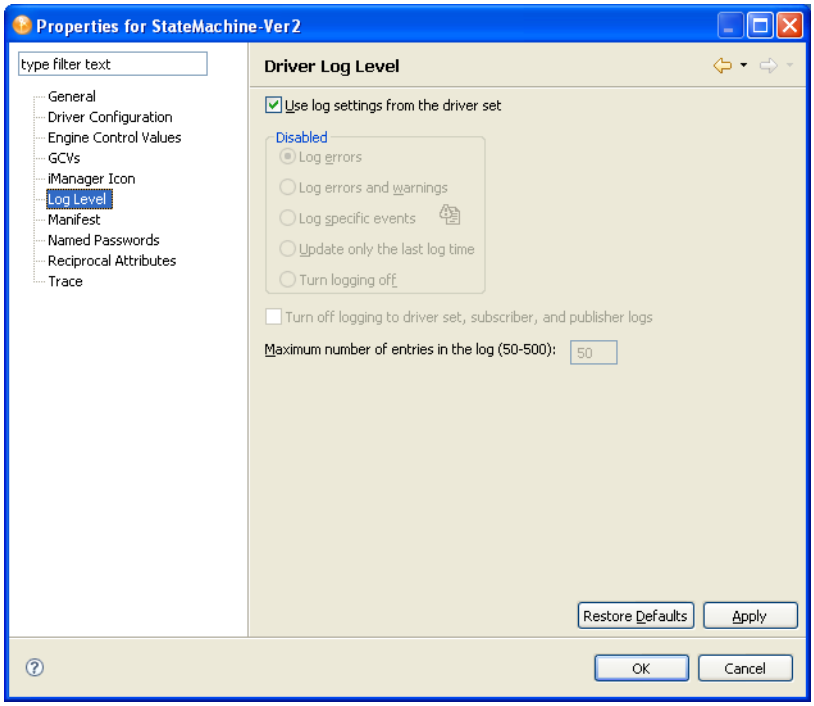
5 Click *Finish* to complete the Structured GCV configuration.

GCVs added to the table are actually the elements of the template. When a GCV element is removed from the template, it is also removed from all the instances of the structured GCVs. In addition, any Edit operation changes the GCV elements in all the instances of the structured GCVs.

4.7.3 Driver Log Level

The Driver Log Level options enable you to view high-level information. For lower-level information, use the Trace option. See Section 4.7.8, “Driver Trace Levels,” on page 100.

Figure 4-13 *Settings for the Driver Log Level*



By default, logging inherits the setting from the driver set. To change the default:

1 Double-click the driver.

2 Select *Driver Log Level*.

3 Select a logging option.

The option that you select determines which information is available in the log.

To configure the audit instrumentation, select *Log specific events*, click the event selector button, select events, then click *OK*.

4 Specify the number of entries in the log.

The default is 50 entries (lines) in the log. If you want a longer history, increase the number.

5 Save changes by clicking *OK*.

The driver log contains messages from the driver. The messages are related to operations that the driver performed or tried to perform. To view the log, use iManager. Select the log icon on the Driver object in the Identity Manager Overview.

4.7.4 Driver Configuration

Identity Manager 3.6 Driver Configuration Wizard adds all policies when the driver is imported, and are not lost if you did not select an option in the Import Information Request forms when you initially import or create a driver. If this is a driver configuration file that came with Identity Manager 3.6, you can change the driver's values through the *Driver Configuration* option in the driver's Properties page.

The two required options for every driver are *Driver Configuration* and *GCVs*. With the *Driver Configuration* option selected, fill out the required values and parameters that are necessary to have the driver run in your network environment. However, because each driver contains different values and parameters, you need to consult the driver manual for specific values. Go to the Identity Manager Drivers web site (http://www.novell.com/documentation/idm36drivers/index.html), then select the manual for the driver you are configuring.

- "Driver Module" on page 96
- "Authentication" on page 97
- "Startup Option" on page 98
- "Driver Parameters" on page 98
- "ECMAScript" on page 99

Driver Module

Table 4-10 *Driver Module Settings*

Field	Description
Java: Name of the Java class	Specify the name of the Java class that will be instantiated for the shim component of the driver. This class can be located in the `classes` directory as a class file, or in the `lib` directory as a `.jar` file.
Native: Name of the DLL	Specify the name of the `.dll` file that will be instantiated for the application shim component of the driver.
Connect to Remote Loader	Select this option if you want to connect the driver to the Metadirectory engine that uses the Remote Loader.
Driver object password: Set Password	Set a password for the Driver object. If you are using the Remote Loader, you must enter a password on this page or the remote driver cannot run. The Remote Loader uses this password to authenticate itself to the remote driver.

Field	Description
Remote Loader client configuration for documentation: Include in documentation	Enables you to document your Remote Loader configuration for the driver. From the drop-down list, select a name that you specified on the driver's documentation property page. To use this option, see Section 4.7.9, "Engine Control Values," on page 103.

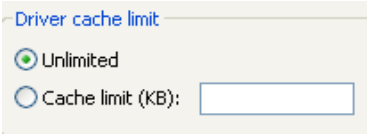
Authentication

Table 4-11 *Authentication Settings*

Field	Description
Authentication information for server	The server that the driver is associated with.
Authentication ID	Specify the application user ID. This ID is used to pass Identity Vault subscription information to the application. If you have enabled SSL/TLS for eDirectory drivers, this option is dimmed.
Connection Information	Specify the address or name and port of the server that the application shim should communicate with.
Set Password	Enables you to set or change an application password (for example, Active Directory).
Remove Password	Deletes the password to the application.
Host name	Specifies the address or name of the machine where the Remote Loader runs. For example, enter `hostname=192.168.0.1`. If you don't specify this communication parameter, this value defaults to *localhost.*
Port	Specifies the port that the Remote Loader uses to accept connections from the remote interface shim. For example, enter port=8090. If you don't specify this communication parameter, this value defaults to 8090.
KMO	Specifies the Key Name of the Key Material Object containing the keys and certificate used for SSL. For example, enter `kmo=remote driver cert`. If you don't specify this communication parameter, no value is stored for this parameter. SSL won't be available.
Other parameters	Provides reference information. It is included when you document your entire project.

Driver Cache Limit

Figure 4-14 *Options for the Driver Cache*



The driver cache is a file that holds Identity Vault events until a driver can process them.

This file can become very large in the following situations:

- If events occur at a steady rate that is faster than Identity Manager can process them over a long period of time.
- If the driver is shut down for long period of time but is not disabled.

By default, the driver cache (file) size is limited only by available disk space. This is the recommended setting.

The only reason to set some other limit is to protect against accidentally filling up the disk. The number that you use depends on the difference between projected amount of available disk space without anything in the cache and the amount of free disk space that you want to ensure will always be left available, divided by the number of drivers on the server.

The primary reason that the cache file becomes very large is if the driver is left not running over a long period of time. In this case, the recommendation is to disable the driver rather than set a cache limit. After the limit is reached, all the cached events are discarded.

Startup Option

Table 4-12 *Startup Settings*

Setting	Description
Auto start	The driver starts automatically when the Metadirectory engine loads.
Manual	You must start the driver manually from the driver state location.
Disabled	Disables the driver.
Do not automatically synchronize the driver	If you don't select this option, a driver that has been deployed but disabled resynchronizes on startup. If you select this option, a driver that has been deployed but disabled does not resynchronize.

Driver Parameters

From this tab, you can enter common driver options, Subscriber and Publisher channel options, as well as edit XML. Because the Driver Parameters options are different for each driver, refer to the Identity Manager Drivers web site (http://www.novell.com/documentation/idm36drivers/index.html) for configuration information on the driver you have selected.

ECMAScript

Displays an ordered list of ECMAScript resource files that are loaded when the driver starts. The ECMAScript files contain extension functions that can be used in policies. For more information, see Using ECMAScript in Policies (http://www.novell.com/documentation/idm40/policy_designer/data/ecmaoverview.html) in the *Policies in Designer 4.0*.

4.7.5 Driver Manifest

The driver manifest is like a resume for the driver. The driver manifest states what the driver supports, and includes a few configuration settings. The driver developer should provide the driver manifest. Usually a network administrator does not need to edit the driver manifest.

For more information, see the developer documentation for Identity Manager drivers.

4.7.6 Driver Named Passwords

The Named Passwords property page allows you to manage (add, edit, delete) named passwords for the selected driver. You can define named passwords on both drivers and driver sets.

Named Passwords let you store multiple passwords securely by referring to each password by a key, or name. When you refer to the named password in a driver policy, you use the name only, not the password value. Then, when the driver needs the password value to execute the policy, it requests the password value from the Metadirectory engine. This method lets you avoid revealing the password value in the code for a driver policy.

The following example shows how a Named Password can be referenced in a driver policy on the Subscriber channel in XSLT:

```
<xsl:value-of
select="query:getNamedPassword($srcQueryProcessor,'mynamedpassword')"
xmlns:query="http://www.novell.com/java/
com.novell.nds.dirxml.driver.XdsQueryProcessor/>
```

You can store and retrieve named passwords for any driver without making changes to the driver shim.

As a security measure, in addition to using named passwords, you should contro

l access to all Identity Manager objects in eDirectory.

NOTE: A driver developer can also customize a driver to use named passwords in other ways, such as retrieving Named Passwords when the driver starts up, instead of requesting them from the Metadirectory engine each time they are needed.

For example, the Identity Manager Driver for Lotus Notes driver has been customized to support additional ways of using named passwords, and examples of those methods are included in the sample driver configurations. For more information, see the Identity Manager driver guides (http://www.novell.com/documentation/idm35/index.html).

4.7.7 Driver iManager Icon

You can see and edit the iManager icons that each driver uses. This is important because iManager renders driver icons in a particular way. However, those icons don't appear in Designer. Conversely, Designer's application icons don't appear in iManager's user interface.

To help bridge that gap, you can view the iManager icon to be used in Designer:

1. In the Modeler, right-click a driver (for example, eDirectory), then select *Properties*.
2. In the left navigation area, select *iManager Icon*.

   Designer displays an icon. It is associated with the driver in Designer, unless a different one was imported and stored on the driver.

For information about editing or changing icons, see Chapter 18, “Editing Icons for Drivers and Applications,” on page 403.

4.7.8 Driver Trace Levels

You can add a trace to your driver. With the driver trace level set, DS Trace displays driver-related Identity Manager events, at the level of detail specified by the driver trace level, as the engine processes the events. The driver trace level affects only the driver or driver set where it is set.

WARNING: You should use the trace level only for testing or for troubleshooting driver issues. Setting a driver trace level on a production driver can cause Identity Manager server to process events slowly.

To set a driver’s trace characteristics:

1. In the Outline view or Modeler, right-click the driver, then select *Properties*.
2. In the driver properties, select *Trace* in the left-side navigation.
3. On the Trace page, specify the driver’s trace settings, then click *OK*.

Field	Description
Trace level	The IDM engine supports the following trace levels: ◆ Trace level 0: Displays fatal messages, errors, warnings and successes. ◆ Trace levels 1: Displays informational messages in addition to the information from Trace level 0. ◆ Trace level 2: Displays contents of XML documents in addition to the information from Trace level 1. ◆ Trace level 3: Displays policy information in addition to the information from Trace level 2. Consult the driver documentation for additional trace options that might be available. **NOTE:** You can also set the driver trace level in Designer by right-clicking a driver (in the Outline or Modeler views) and selecting *Live > Set Driver Trace Level*. This immediately deploys the trace level to the selected driver. To update the driver trace level in your project as well, select *Update local model*.
Trace level: Use setting from the driver set	If you select this option, all trace levels set at the driver set take precedence over any driver settings. Otherwise, the driver settings are effective.
Trace file	Specify a filename and location where the Identity Manager information is written for the selected driver. When a value is set in this field, all Java information for the driver is written to file. As long as the file is specified, Java information is written to this file. If you do not need to debug Java, leave this field blank.
*Trace file: Use setting from the driver se*t	If you select this option, all trace levels set at the driver set level take precedence over any driver settings. Otherwise, settings at the driver level are effective.
Trace File Encoding	The trace file uses the system's default encoding. You can specify another encoding if desired. This option is only available if the driver is associated with a server that is version 3.6 or newer.
Trace file size limit	Allows you to set a limit for the Java trace file. Select *Unlimited* to allow the file to grow to fill the disk. **NOTE:** The trace file is created in multiple files. Identity Manager automatically divides the maximum file size by ten and creates ten separate files. The combined size of these files equals the maximum trace file size.
Trace file size limit: Use setting from the driver set	If you select this option, all trace levels set at the driver set level take precedence over any driver settings. Otherwise, settings at the driver level are effective.

Field	Description
Trace name	Helps you track trace messages. The name that you specify here appears with the driver trace messages. Use a trace name if the driver name is very long.

The following methods help you capture and save Identity Manager trace information.

Windows

Open the Control Panel, select *NDS Services*, then click *DS Trace.DLM > Start*. A window named NDS Server Trace Utility opens.

To set the filters to capture the DirXML trace information:

1. Click *Edit > Options > Clear All*.
2. Click the boxes next to *DirXML* and *DirXML Drivers*, then click *OK*.

To save the information to a file:

1. Click *File > New*.

   A dialog box prompts for a filename.
2. Enter a filename with the extension of `.log`.
3. To stop capturing information, click *File > Close*.

   The file is saved.

UNIX

Use the `ndstrace` command at the console to display the Identity Manager events. The `exit` command quits the trace utility.

Table 4-13 *ndstrace Commands*

Command	Description
`Set ndstrace=nodebug`	Turns off all trace flags.
`Set ndstrace on`	Displays trace messages to the console.
`Set ndstrace file on`	Captures trace message to the `ndstrace.log` file in the `/var/nds` directory.
`Set ndstrace file off`	Stops capturing trace messages to the file.
`Set ndstrace=+dxml`	Displays the Identity Manager events
`Set ndstrace=+dvrs`	Displays the Identity Manager driver events.

iMonitor

Use iMonitor to get DS Trace information from a Web browser.

Table 4-14 *Platforms and Commands for Web Browsers*

Platform	Command
Windows	`ndsimon.dlm`
Linux/Solaris/AIX/HP-UX	`ndsimonitor`

1. Access iMonitor from http://server_ip:8008/nds (the default port).
2. Click *Trace Configuration*.
3. Click *Clear All*.
4. Click *DirXML* and *DirXML Drivers*.
5. Click *Trace On*, then click *Trace History*.
6. Click the *Current* document icon to view the live trace.

4.7.9 Engine Control Values

The engine control values enable you to change certain default behaviors of the Metadirectory engine can be changed. You can access the values only if a server is associated with the Driver Set object. The values are populated based on the Identity Manager version of the servers that are associated with the driver set (servers can be associated through the *Engine Controls for Server* entry).

Changing a version of an Identity Manager server affects the engine controls for all drivers in a driver set that is associated with the server. When the Identity Manager version is changed, the engine controls for all associated drivers are updated to match the specified version. During the update process, all current settings for existing engine controls are merged into the new engine controls. If the engine controls are not valid for the version of the selected server, they are removed as options.

In Designer:

1. In the Modeler, right-click the driver line.
2. Select *Properties > Engine Control Values*.
3. Click the tooltip icon to the right of the *Engine Controls for Server* field. If a server is associated with the Identity Vault, and if you are authenticated, the engine control values display in the large pane.

Table 4-15 *Engine Control Values*

Field	Description
Subscriber channel retry interval in seconds	The Subscriber channel retry interval controls how frequently the Metadirectory engine retries the processing of a cached transaction after the application shim's Subscriber object returns a retry status.
Qualified form for DN-syntax attribute values	The qualified specification for DN-syntax attribute values controls whether values for DN-syntax attribute values are presented in unqualified slash form or qualified slash form. A True setting means the values are presented in qualified form.

Field	Description
Qualified form from rename events	The qualified form for rename events controls whether the new-name portion of rename events coming from the Identity Vault is presented to the Subscriber channel with type qualifiers. For example, CN=. A True setting means the names are presented in qualified form.
Maximum eDirectory replication wait time in seconds	The maximum eDirectory replication wait time controls the maximum time that the Metadirectory engine waits for a particular change to replicate between the local replica and a remote replica. This only affects operations where the Metadirectory engine is required to contact a remote eDirectory server in the same tree to perform an operation and might need to wait until some change has replicated to or from the remote server before the operation can be completed (for example, object moves when the Identity Manager server does not hold the master replica of the moved object; file system rights operations for Users created from a template.)
Use non-compliant backwards-compatible mode for XSLT	This control sets the XSLT processor used by the Metadirectory engine to a backward-compatible mode. The backwards-compatible mode causes the XSLT processor to use one or more behaviors that are not XPath 1.0 and XSLT 1.0 standards-compliant. This is done for backwards-compatibility with existing DirXML style sheets that depend on the non-standard behaviors. For example, the behavior of the XPath "!=" operator when one operand is a node set and the other operand is other than a node set is incorrect in DirXML releases up to and including Identity Manager 2.0. This behavior has been corrected; however, the corrected behavior is disabled by default through this control in favor of backwards-compatibility with existing DirXML style sheets.
Maximum application objects to migrate at once	This control is used to limit the number of application objects that the Metadirectory engine requests from an application during a single query that is performed as part of a Migrate Objects from Application operation. If java.lang.OutOfMemoryError errors are encountered during a Migrate from Application operation, this number should be set lower than the default. The default is 50. **NOTE:** This control does not limit the number of application objects that can be migrated; it merely limits the batch size.
Set creatorsName on objects created in Identity Vault	This control is used by the Identity Manager engine to determine if the creatorsName attribute should be set to the DN of this driver on all objects created in the Identity Vault by this driver. Setting the creatorsName attribute allows for easily identifying objects created by this driver, but also carries a performance penalty. If not set, the creatorsName attribute defaults to the DN of the NCP™ Server object that is hosting the driver.
Write pending associations	This control determines whether the Identity Manager engine writes a pending association on an object during Subscriber channel processing. Writing a pending association confers little or no benefit but does incur a performance penalty. Nevertheless, the option exists to turn it on for backwards compatibility.

Field	Description
Use password event values	This control determines the source of the value reported for the nspmDistributionPassword attribute for Subscriber channel Add and Modify events. Setting the control to False means that the current value of the nspmDistributionPassword is obtained and reported as the value of the attribute event. This means that only the current password value is available. This is the default behavior. Setting the control to True means that the value recorded with the eDirectory event is decrypted and is reported as the value of the attribute event. This means that both the old password value (if it exists) and the replacement password value at the time of the event are available. This is useful for synchronizing passwords to certain applications that require the old password to enable setting a new password.
Enable password synchronization status reporting	This control determines whether the Identity Manager engine reports the status of Subscriber channel password change events. Reporting the status of Subscriber channel password change events allows applications such as the Identity Manager User Application to monitor the synchronization progress of a password change that should be synchronized to the connected application.

4.7.10 Driver Health Configuration

This properties page is only available if the Identity Manager server version is set to 3.6 or newer.

The Driver Health Configuration allows you to monitor a driver's state of health (green, yellow, or red), and to specify the actions to perform in response to each of these health states.

To do so, you define the conditions (criteria) that determine each of the health states, and the associated actions to perform whenever the driver's health state changes. For example, if the driver's health changes from a green state to a yellow state (based on the conditions you've established), you can perform such actions as restarting the driver, shutting down the driver, and sending an e-mail to the person designated to resolve issues with the driver.

You can also define custom driver states that are independent of the standard green, yellow and red. Whenever the driver meets the conditions for the custom statet, Designer performs the associated actions.

To monitor a driver's health state using the Driver Health Configuration, you must complete the following tasks:

- "Creating a Driver Health Configuration" on page 106
- "Modifying the Health State Conditions" on page 107
- "Creating a Driver Health Job" on page 109

Additionally, you can perform the following tasks to further configure the Driver Health Check environment:

- "Modifying the Health State Actions" on page 110

- “Creating a Custom State” on page 111
- “Modifying the Driver Health Job Settings” on page 112

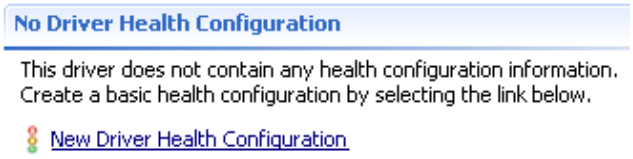
NOTE: Monitoring driver health is applicable only to deployed drivers. Designer does not indicate driver health in the Modeler or any other pre-deployment interface. Use iManager to monitor the health of deployed drivers. For more information about driver health monitoring in iManager, see Monitoring Driver Health (http://www.novell.com/documentation/idm36/idm_common_driver/data/bdygx01.html) in the *Identity Manager 3.6 Common Driver Guide* (http://www.novell.com/documentation/idm36/idm_common_driver/data/front.html).

Creating a Driver Health Configuration

The health configuration of drivers at version 3.6 (or newer) is configured automatically. Skip this section if your drivers are version 3.6 (or newer). If you have drivers older than version 3.6, you must create the health configuration for each driver you want to monitor.

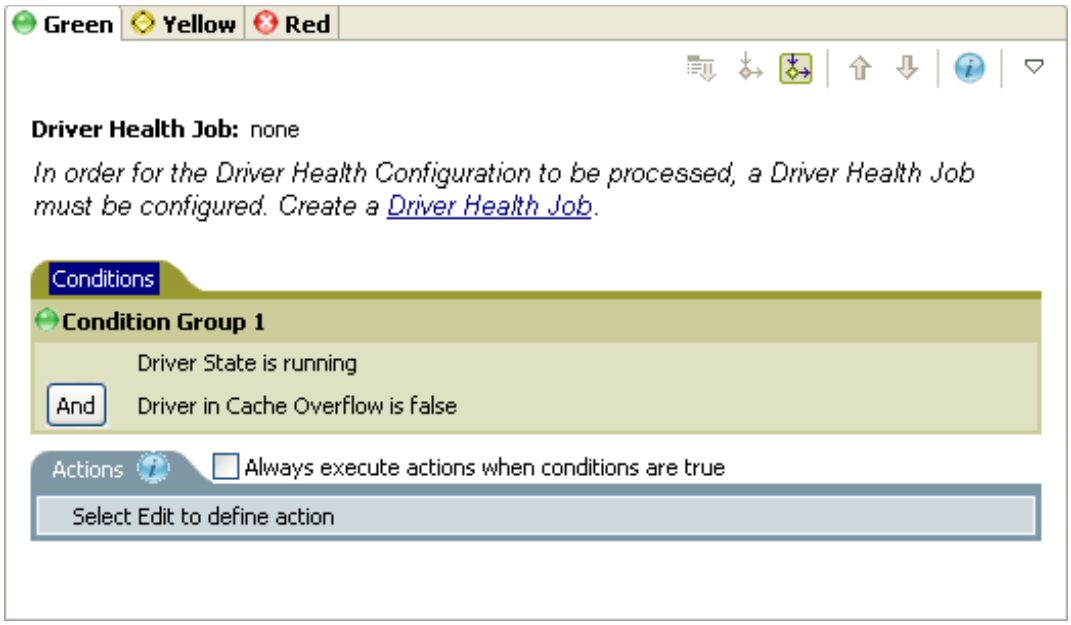
To create a driver health configuration:

1. In the Modeler or Outline view, right-click the driver, then select Properties.
2. In the left-side navigation, select *Health*.



3. Select *New Driver Health Configuration*.

   Designer creates a basic health configuration with sample conditions for the green and yellow states (none for red).



4 Continue with "Modifying the Health State Conditions" on page 107.

Modifying the Health State Conditions

The driver health configuration lets you define the conditions that determine each health state. The green state contains conditions intended to represent a healthy driver, while a red state represents an unhealthy driver, having failed conditions for both green and yellow states.

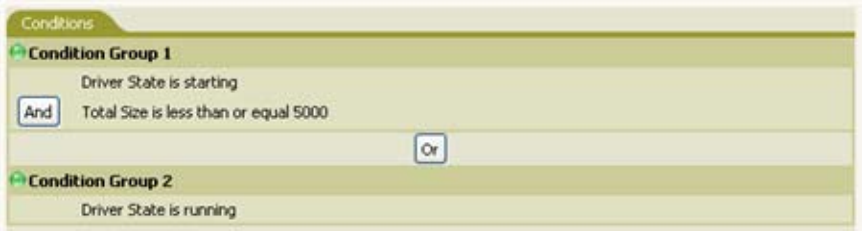
The Driver Health job evaluates the conditions for the green state first. If the driver fails to meet the green conditions, it evaluates the yellow conditions. If the driver fails to meet the yellow conditions, it is automatically assigned a red state.

To modify the conditions for a state:

1 In the Modeler or Outline view, right-click the driver where you want to modify the health check configuration, then select *Properties*.

2 In the left-side navigation, select *Health*.

3 Click the state tab (Green or Yellow) that you want to modify.



The tab displays the current conditions for the health state. Conditions are organized into groups, with logical operators (either AND or OR), to link each condition and condition group.

Table 4-16 describes the conditions that the Driver Health job can evaluate.

Table 4-16 *Driver Health Check Conditions*

Condition	Description
Driver State	Running, stopped, starting, not running, or shutting down. For example, one of the default conditions for the green health state is Driver State is running.
Driver in Cache Overflow	The state of the cache used for holding driver transactions. If the driver is in cache overflow, all available cache has been used. For example, the default condition for the green health state is Driver in Cache Overflow is false and the default for the yellow health state is Driver in Cache Overflow is true.
Newest	The age of the newest transaction in the cache.
Oldest	The age of the oldest transaction in the cache.
Total Size	The size of the cache in bytes.
Unprocessed Size	The size of all unprocessed transactions in the cache.
Unprocessed Transactions	The number of unprocessed transactions in the cache. You can specify all transactions types or specific transaction types (such as adds, removes, or renames).
Transaction History	The number of transactions processed at various points in the subscriber or publisher channel over a given period of time. This condition uses multiple elements in the following format: <*transaction type*> <*transaction location and time period* > <*relational operator*> <*transaction number*>. ◆ <*transaction type*>: Specifies the type of transaction being evaluated. For example, adds, removes, renames, and so forth. ◆ <*transaction location and time period*>: Specifies the point in the subscriber or publisher channel and the time period being evaluated. For example, you might evaluate the total number of transactions processed as publisher reported events over the last 48 hours. The time period cannot exceed the Transaction Data Duration setting, which is configurable in the Driver Health job. For more information, see "Modifying the Driver Health Job Settings" on page 112. ◆ <*relational operator*>: Specifies the relationship between the identified transactions and the <*transaction number*> (equal to, less than, greater than, and so forth.) ◆ <*transaction number*>: Specifies the number of transactions being used in the evaluation. For example: `<number of adds> <as publisher commands> <over the last 10 minutes> <is less than> <1000>`

Condition	Description

Available History	The amount of transaction history data that is available for evaluation. This condition helps ensure that a Transactions History condition does not cause the current state to fail because it does not have enough transaction history data collected for the time period being evaluated.
	
	For example, assume that you want to use the Transactions History condition to evaluate the number of “Add as Publisher” commands over the last 48 hours. However, you don't want the condition to fail if there is less than 48 hours of data. You could create condition groups similar to the following:
	
	`Group1`
	`Available History <is less than> <48 hours>`
	
	`or`
	
	`Group2`
	`Available History <is greater than or equal to> <48 hours> and`
	`Transactions History <number of adds> <as publisher commands> <over the last 48 hours> <is less than> <1000>`
	
	The state evaluates to true if either condition group is true.
	
	The state evaluates to false if both conditions evaluate to false.

4 Modify the condition criteria as desired.

- To add a new group, select the *Conditions* tab, then click *Append Condition Group* .
- To add a condition, select an existing condition group, then click *Append Condition* .
- To reorder condition groups or individual conditions, select the condition group or condition, then click *Move Up* or *Move Down* . You can also use these buttons to move a condition from one group to another.
- Cut, copy and paste a condition group or condition to the clipboard by right-clicking the item, then selecting the appropriate clipboard action.

5 Click *Apply* to save your changes without closing the Properties page.

6 If you want to change the actions associated with the conditions you’ve set, continue with “Modifying the Health State Actions” on page 110.

Creating a Driver Health Job

The driver health job is available only for Identity Manager server version 3.6 or newer.

The Driver Health job executes periodically to evaluate the health of a driver configured for health checks. The job evaluates the conditions defined for each of the driver’s health states, then assigns the driver the appropriate state. The job also executes any actions associated with the assigned state.

If a Driver Health job does not exist, the Driver Health Configuration page displays a *New Driver* link from which you can configure the Driver Health job. If a Driver Health job already exists, the Driver Health Configuration page does not display this prompt.

To create a Driver Health job:

1 In the Modeler or Outline view, right-click the driver, then select Properties.

2 In the left-side navigation, select *Health*.

3 Click *Driver Health Job* to open the Job dialog box. Select the appropriate job, then click *OK*.

Follow the prompts to import the configuration file for the Driver Health job. Refer to the following information for details:

- **Where to place the driver:** Place the job in the same driver set as the driver. The correct driver set is selected by default. You can only have one driver health job per driver set.
- **Import a configuration:** Import the configuration from the server. In the Show field, select Identity Manager 3.6 configurations, then select the Driver Health job in the Configurations field.
- **Email server:** Select the e-mail server that you want used for any actions that initiate e-mail. If you have not defined additional e-mail servers, select the Default Notification Collection server.
- **Servers:** If the driver set is associated with only one server, that server is selected and cannot be changed. If the driver set is associated with multiple servers, select the server where you want to run the job.

After creating the Driver Health job, you can modify job settings as needed. For example, you can configure how often the job runs, which drivers use the job, and how much data the job maintains to support transaction history. For more information, see "Modifying the Driver Health Job Settings" on page 112.

Modifying the Health State Actions

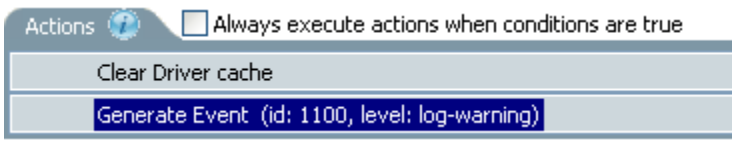
The Driver Health Configuration lets you define the actions that the Driver Health job performs when the driver health state changes. For example, if the state changes from green to yellow, you can shut down or restart the driver, generate an event, or start a workflow.

The Driver Health job performs a health state's actions only once each time the conditions are met; as long as the driver state remains the same, the actions do not repeat. If the driver state changes because its conditions are no longer met, the Driver Health job performs the state's actions again the next time its conditions are met.

1 In the Modeler or Outline view, right-click the driver where you want to modify the health check configuration, then select *Properties*.

2 In the left-side navigation, select *Health*.

3 Select the state tab (Green or Yellow) that you want to modify.



The tab displays the current actions for the health state. If no action is assigned, the Driver Health Configuration displays *Define new action here* in the *Actions* tab.

4 Select the *Actions* tab, then click *Append Action* to add an action to the health state.

Select an action from the dropdown list.Table 4-17 describes the actions that the Driver Health job can perform.

NOTE: Some actions require additional information before they will execute.

Table 4-17 *Driver Health Check Actions*

Action	Description
Clear Driver Cache	Removes all transactions, including unprocessed transactions, from the cache.
Execute ECMAScript	Executes an existing ECMAScript. Specify the DirXML-Resource object that contains the ECMAScript.
Generate Event	Generates an event that can be used by Novell Audit and Novell Sentinel.
On Error	If an action fails, tells Designer what to do with the remaining actions, the current health state, and the Driver Health job.
Restart Driver	Restarts the driver (stop, then start)
Send Email	Sends an e-mail to one or more recipients. The template you want used in the e-mail message body must already exist.
Start Driver	Starts the driver.
Start Workflow	Starts a provisioning workflow.
Stop Driver	Stops the driver.
Write Trace Message	Writes a message to the driver's log file, using the message parameters specified in the action.

5 If you want the actions executed every time the conditions evalute to true, click *Always execute actions when conditions are true*.

By default, actions are performed only one time while a driver's health state remains the same; regardless of the number of times the conditions are evaluated, as long as the health state remains in effect (true), the actions are not repeated. For example, when the driver's health state changes from red to green, the green state's actions are executed. The next time the conditions are evaluated, if the health state is still green, the actions are not repeated.

Selecting the *Always execute actions when conditions are true* setting causes the actions to be repeated each time the condition evaluates to true. For example, if the driver's health state repeatedly evaluates to green without changing to another state, the green state's actions are repeated after each evaluation.

6 Click *Apply* to save your changes without closing the Properties page.

Creating a Custom State

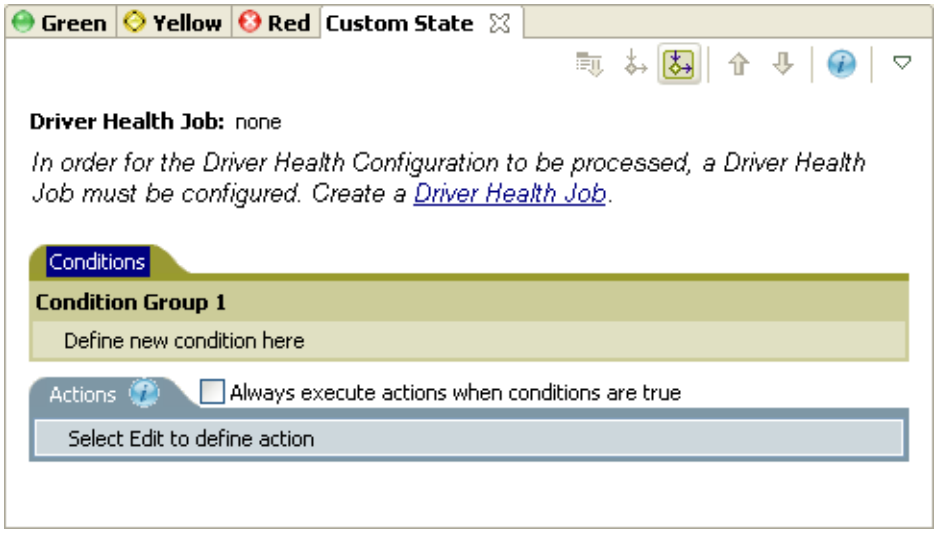
The Driver Health Configuration lets you create one or more custom states to perform actions independent of the driver's current health state (green, yellow, red). If the driver meets the custom state's conditions, the Driver Health job performs its actions.

As with the standard driver health states (green, yellow, red), the Driver Health job performs a custom state's actions only once each time the conditions are met; as long as the driver state remains the same, the actions do not repeat. If the driver state changes because the custom state's conditions are no longer met, the Driver Health job performs the custom state's actions again the next time its conditions are met.

1 In the Modeler or Outline view, right-click the driver where you want to create a custom state, then select *Properties*.

2 In the left-side navigation, select *Health*.

3 Select the pulldown menu ▽, then select *New Custom State*.



4 Define the conditions and actions for the custom state, then click *Apply* to save the changes without closing the Properties page.

For information about defining state conditions, see "Modifying the Health State Conditions" on page 107. For information about defining state actions, see "Modifying the Health State Actions" on page 110.

Modifying the Driver Health Job Settings

NOTE: The driver health job is available only for Identity Manager server version 3.6.

The Driver Health job evaluates the conditions for the health states and assigns the driver the appropriate state. The job also executes any actions associated with the assigned state.

As with all driver jobs, there are several settings that you can modify to optimize the job's performance for your environment, including how often the job runs, which drivers use the job, and how much data the job maintains to support transaction history.

To modify the job settings:

1 In the Modeler or Outline view, open the driver set object where the driver health job is stored.

2 Right-click the appropriate job object, then select *Edit*.

3 Change the desired settings on the following tabs, then click *OK* to save your changes:

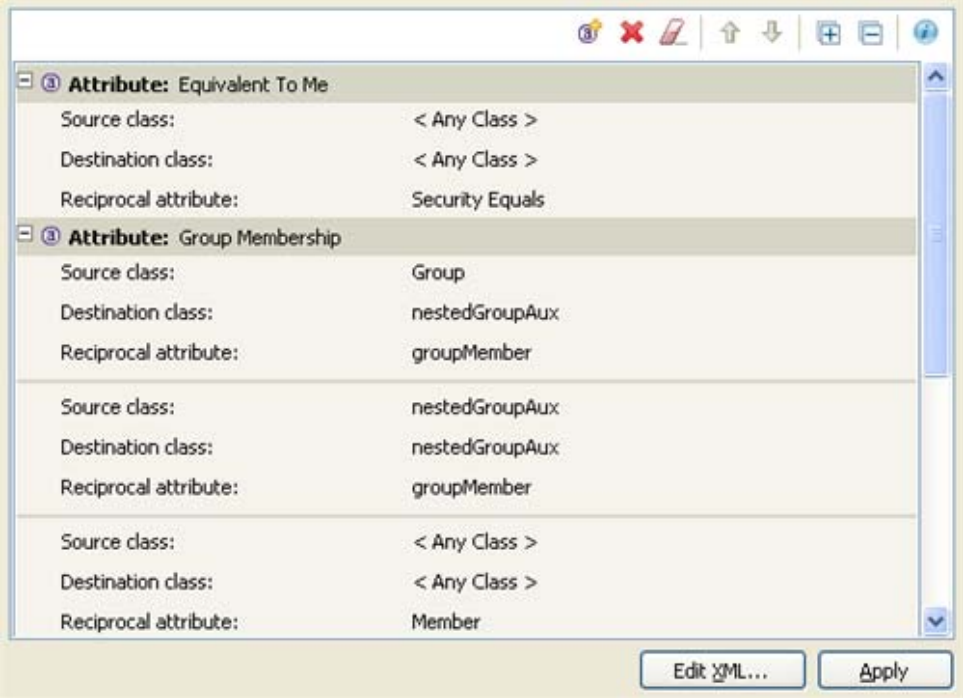
Tab	Description
Schedule	The Driver Health job is a continuously running job, meaning that it does not stop unless a health state action shuts it down or you shut it down manually. The job must run continuously to be able to support transaction data collection for use in Transactions History conditions. If the job does stop, it is restarted based on the schedule. The default schedule checks every minute to see if the job is running. If the job is not running, it is started.
Scope	By default, the job applies to all drivers in the driver set. This means that you only need one Driver Health job per driver set. However, you can create multiple Driver Health jobs for different drivers within the same driver set. For example, you might have some drivers whose health you want updated more frequently than other drivers, in which case you would need at least two Driver Health jobs.
Parameters	You can change any of the following job parameters: ◆ **Login ID:** This defaults to the login ID that was used when creating the driver job. You should only change this if you want the driver to authenticate using different credentials. ◆ **Login password:** This is the password required for the login ID that you supplied in the Login ID field. ◆ **Polling interval:** Determines how often the job evaluates the conditions for the health states, assigns the driver the appropriate state, executes any actions associated with the assigned state, and stores the driver's transaction data. The default polling interval is one minute. ◆ **Polling interval units:** Specifies the time unit (minutes, hours, days, weeks) for the number specified in the Polling interval setting. ◆ **Duration transaction data is kept:** Specifies how long a driver's transaction data is kept. The default retains a transaction for two weeks before being deleted. Longer transaction durations require more memory. For example, to store transaction data for one driver every minute (*Polling interval*) for two weeks requires approximately 15 MB of memory. ◆ **Duration units:** Specifies the time unit (minutes, hours, days, weeks) for the number specified in the Duration transaction data is kept setting.

4.7.11 Reciprocal Attributes

NOTE: This properties page is only available if the Identity Manager server version is set to 3.6.

The Reciprocal Attributes propoerties lets you create and manage backlinks between objects. For example, the Group object includes a Members attribute that contains pointers to all User objects that belong to that group. Similarly, each User object includes a Group Membership attribute that points to the Group objects of which that user is a member. These two-way links between objects are known as *reciprocal mappings*.

Figure 4-15 *Custom Reciprocal Attribute Mapping Property Page for Driver Objects*



You can manage all reciprocal mapping configuration from the toolbar in the property page, which contains the following toolbar icons:

Icon	Description
	Use the *New Attribute* icon to add a new attribute to the reciprocal mapping list.
	Use the *Delete* icon to delete the currently selected reciprocal mapping entry from the list.
	Use the *Clear All Attribute Mappings* icon to delete all reciprocal mappings.
	Use the *Move Up* icon to move the currently selected attribute up in the mapping list. To do so, select the attribute entry you want to move up, then click *Move up*.
	Use the *Move Down* icon to move the currently selected attribute down in the mapping list. To do so, select the attribute entry you want to move down, then click *Move Down*.
	Use the *Expand All* icon to expand all reciprocal attribute mapping entries.
	Use the *Collapse All* icon to expand all reciprocal attribute mapping entries.

The Custom Reciprocal Mapping page lets you do the following:

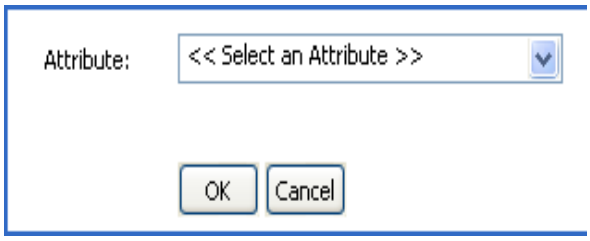
- "Adding a Reciprocal Attribute Mapping" on page 115
- "Removing a Reciprocal Attribute Mapping" on page 115

- "Removing an Attribute from the Reciprocal Mapping List" on page 116
- "Editing Reciprocal Attribute XML" on page 116

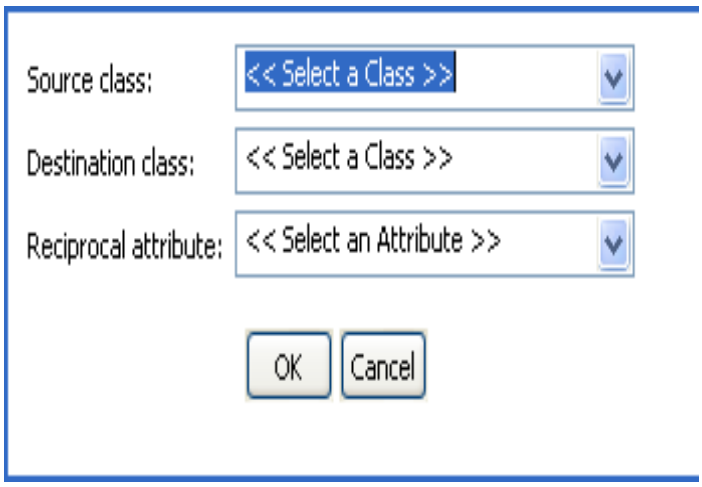
Adding a Reciprocal Attribute Mapping

When creating a reciprocal attribute mapping, you must first add one of the attributes to the reciprocal mapping list:

1. In the Reciprocal Attributes page, click *New Attribute* .
2. In the new attribute entry, select the desired attribute from the dropdown list, then click *OK*.



3. Specify the specifics of the reciprocal mapping, then click *OK*.



Source Class	Specifies the class name to which the attribute in the mapping list is associated. For example, if you placed the Group Membership attribute in the reciprocal mapping list, the associated Source Class is User.
Destination Class	Specifies the class name associated with the attribute to which you want to create a reciprocal mapping.
Reciprocal Attribute	Specifies the attribute name to which you want to create a reciprocal mapping.

Removing a Reciprocal Attribute Mapping

To remove a reciprocal mapping between attributes:

1. Select the reciprocal mapping you want to remove by selecting it in the reciprocal mapping list.

When selected, the attribute name in the *Attribute* tab is highlighted.

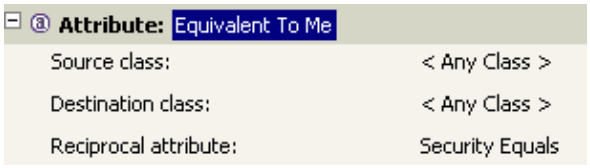
2 Click *Delete* .

Removing an Attribute from the Reciprocal Mapping List

To remove an attribute from the reciprocal mapping list:

1 Select the attribute you want to remove by selecting it in the reciprocal mapping list.

When selected, the attribute name in the *Attribute* tab is highlighted.



2 Click *Delete* .

To remove all attributes from the reciprocal attribute mapping list, click *Clear All Attribute Mappings* .

Editing Reciprocal Attribute XML

If desired, you can edit the XML for a reciprocal attribute directly. To do so, click *Edit XML* in the Custom Reciprocal Attribute Mapping page. This opens a basic XML editor that lets you modify the XML directly. When finished, click *OK* or *Cancel* to close the XML editor.

4.8 Configuring Jobs

Designer has a job scheduling utility to schedule events. Through this utility, the system can be set to disable an account on a specific day, or to initiate a workflow to request an extension for a person's access to a corporate resource. Designer's job scheduler contains the same functionality as the job scheduler found in iManager. For information on creating jobs, see Section 14.1, "Creating a Job," on page 328.

In the Outline view, right-click the Job icon, then select Properties.

- Section 4.8.1, "General," on page 116
- Section 4.8.2, "Trace," on page 117

4.8.1 General

You have one selection under the General heading: *Policy Name*. You can change the job's name by modifying the name that appears in the *Policy Name* entry. Then click *OK*.

4.8.2 Trace

Through the Modeler, you can add a trace level to your jobs. With the trace level set, DS Trace displays the Identity Manager events as the engine processes the events. The trace level only affects the driver where it is set.

WARNING: You should use the trace level only for testing or for troubleshooting driver issues. Setting a driver trace level on a production driver can cause Identity Manager server to process events slowly.

Table 4-18 describes the Job trace settings:

Table 4-18 *Job Trace Settings*

Field	Description
Trace level	As the job trace level increases, the amount of information displayed in DS Trace increases. Trace level 1 shows errors, but not the cause of the errors. To see password synchronization information, set the trace level to 5.
Trace file	Specify a filename and location where the Identity Manager information is written for the selected driver. When a value is set in this field, all Java information for the job is written to file. As long as the file is specified, Java information is written to this file. If you do not need to debug Java, leave this field blank.
Trace File Encoding	The trace file uses the system's default encoding. You can specify another encoding if desired. This option is only available if the job is on a driver or driver set associated with a server that is version 3.6 or newer.
Trace file size limit	Allows you to set a limit for the Java trace file. If you set the file size to Unlimited, the file grows in size until no disk space is available. **NOTE:** The trace file is created in multiple files. Identity Manager automatically divides the maximum file size by ten and creates ten separate files. The combined size of these files equals the maximum trace file size.
Trace name	Helps you track job trace messages. The name that you specify here appears with the job trace messages.

For more information about viewing as saving trace information with DS Trace, see Section 4.7.8, "Driver Trace Levels," on page 100.

4.9 Configuring ID Policy Containers

An ID Policy container is a repository for ID policies and is used in conjunction with the ID Provider driver. When the ID Provider driver receives an ID request from a client, it generates an identification that is based on the ID policy specified in the request and passes the identification to the client.

To configure an ID Policy container, you must first add the ID Provider driver to a driver set that accesses an Identity Manager 3.6 Identity Vault. Then, under the ID Provider driver, create an ID Policy container by right-clicking the ID Provider driver and selecting *New > ID Policy Container*. After the container is created, double-click the ID Policy container in the Outline view, or right-click the ID Policy container and select *Properties*.

Table 4-19 *ID Policy Container General Settings*

Field	Description
Name	The name of the ID Policy container. You can change the name as necessary.
Notes	You can add notes to better define how you are using the ID Policy container.

In order for ID policies to work, you must also add and configure an ID policy in the ID Policy container. See Section 4.10, “Configuring ID Policies,” on page 118.

4.10 Configuring ID Policies

An ID policy allows the ID Provider driver to generate unique IDs. When the ID Provider driver receives an ID request from a client, it generates an identification that is based on the ID policy specified in the request and passes it to the client. ID Provider driver and ID policies are a feature of Identity Manager 3.6.

The ID Provider driver can act as a client itself and can assign IDs to objects in the Identity Vault. For more information about the ID Provider driver and its components, see the ID Provider Driver Guide (http://www.novell.com/documentation/idm36drivers/index.html).

To configure an ID policy, you must first add the ID Provider driver to a driver set that accesses an Identity Manager 3.6 Identity Vault. Then under the ID Provider driver, create an ID Policy container, then add an ID policy. Once created, double-click the ID policy in the Outline view, or right-click the ID policy and select *Properties*.

Figure 4-16 *The ID Policy's General Properties Page*

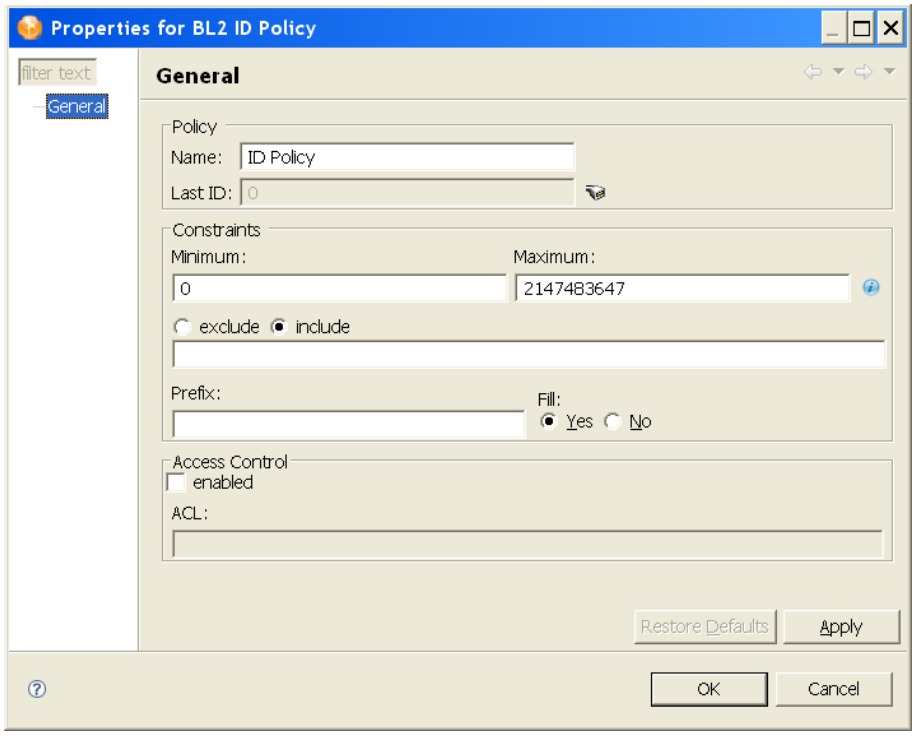


Table 4-20 *The ID Policy's General Settings*

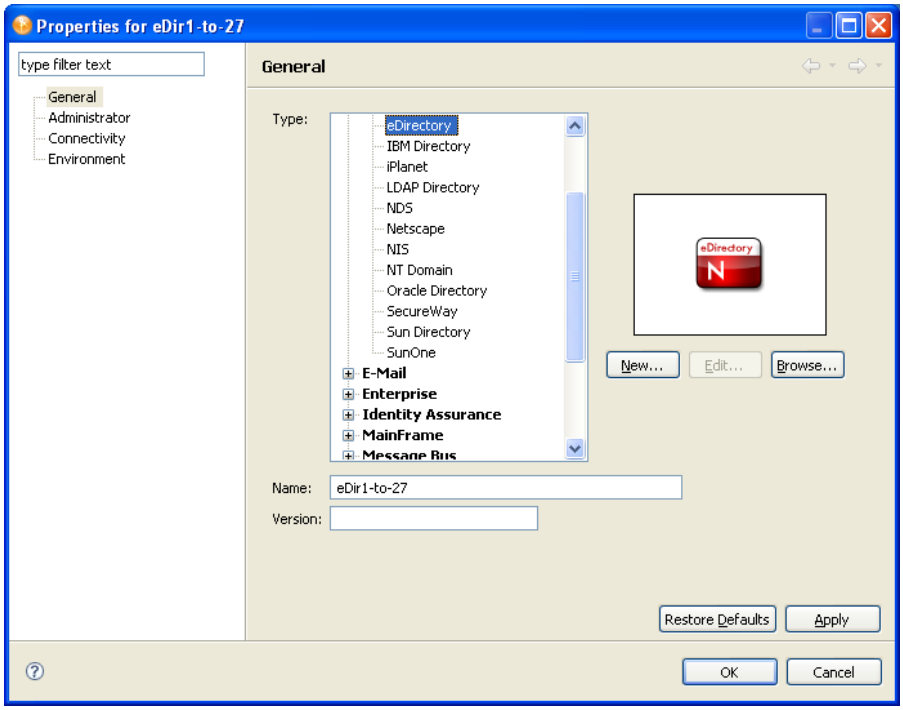
Field	Description
Policy Name	The name of the ID policy.
Policy's Last ID	The last ID number that was used by this ID policy. If you have deployed this ID policy, use the *Connect* icon to update this field to the last ID number that was stored in the Identity Vault for this ID policy. **NOTE:** Only the ID Provider driver can update the last value stored in the Identity Vault.
Constraints:	
Minimum/Maximum	Numbers must be between 0 and 2147483647. If you have a fixed system that can only handle eight digits, set the *Maximum* to 99999999.
Exclude/Include	Allows you to include or exclude a set of numbers that you type in. Numbers can be typed in a coma-delimited list and you can use ranges, such as 10,100,1000,5000-10000,1099, etc.
Prefix:	Allows you to give a prefix to the IDs that are generated using this ID policy. If you create multiple ID policies, a prefix is useful to see which ID policies are being used. An example is WFID, for workforce IDs.

Field	Description
Fill: Yes/No	If you choose *Yes*, the ID is filled with leading zeros (0) up to the maximum length. This helps keep generated IDs at the same length. If you select *No*, it does nothing and the ID lengths increment over time.
Access Control:	
Enabled	Check this box if you want to enable access control lists.
ACL:	Type in the access control lists you want to use. Access control must be enabled before you can type in ACLs.

4.11 Configuring Application Properties

To view or change an application's settings, double-click the application (for example, LDAP Directory) in the Modeler.

Figure 4-17 *The General Properties Page for an Application*



- Section 4.11.1, "General," on page 121
- Section 4.11.2, "Environment," on page 121
- Section 4.11.3, "Administrator," on page 121
- Section 4.11.4, "Connectivity," on page 121
- Section 4.11.5, "AD Domain," on page 125

4.11.1 General

Table 4-21 *An Application's General Settings*

Field	Description
Type	Changes the type of application your driver connects to. For example, if you configure a JDBC driver to connect to a MySQL* database, but then need to change to an Oracle database, you can scroll to *Database*, select *Oracle*, then click *Apply*.
New	Enables you to edit a driver's icon. See Section 4.7.7, "Driver iManager Icon," on page 100.
Edit	Enables you to use the Icon Editor to customize the application's icon. Is available after you click *New*, edit an icon, and click *Update*.
Browse	Enables you to navigate to and select an image file.
Name	Enables you to customize the application's name or label.
Version	Enables you to document the application's version.

4.11.2 Environment

You can enter notes about the application's environment.

4.11.3 Administrator

The *Administrator* option is divided into three sections. Entering information in these sections is optional.

- **Personal Information:** Enables you to enter information specific to the Identity Vault, such as Name, Title, Department, and Location,
- **Contact Information:** Enables you to enter information such as Email, Phone, Cell phone, Pager, and Fax.
- **Notes:** Allows you to type any reminders you might need for future reference.

4.11.4 Connectivity

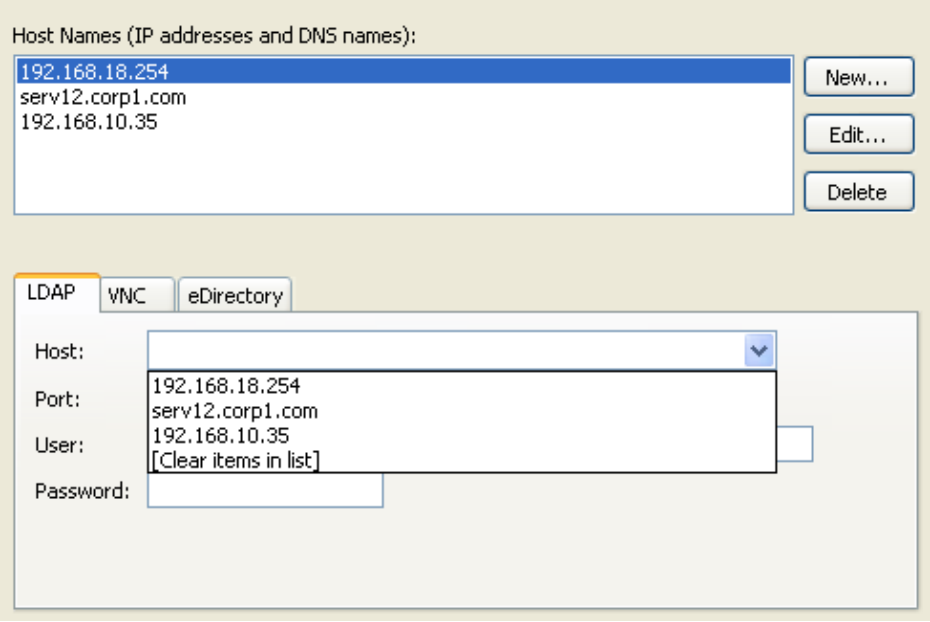
- "Host Names" on page 122
- "LDAP" on page 122
- "VNC" on page 123
- "eDirectory / NDS" on page 123
- "Configuring a Remote Connection" on page 124
- "Customizing the Viewer" on page 124

Host Names

NOTE: This control is available only for eDirectory and NDS applications.

The Host Names field lets you create a list of server IP addresses and DNS names for your eDirectory or NDS application. Since servers can have multiple IP addresses and DNS names, it is useful to be able to create a list of those host names that you can easily access when configuring connectivity for your eDirectory or NDS application.

Figure 4-18 *Host Name List for eDirectory and NDS Applications*



You can add, modify, and delete host names from the Host Names list.

When you specify a host on the *LDAP*, *VNC*, or *eDirectory*/*NDS* tabs, the host entry is automatically added to the Host Names list.

Double-click an entry in the Host Names list to automatically populate the *Host* field in the the *LDAP*, *VNC*, or *eDirectory*/*NDS* tabs.

Host entries in the Host Names list are also available from the *Host* field dropdown list in the *LDAP*, *VNC*, and *eDirectory*/*NDS* tabs.

LDAP

You can configure some applications (for example, Active Directory, eDirectory, and LDAP) for an LDAP connection. If the application doesn't support an LDAP connection, the *LDAP* tab doesn't display.

Host: The server's IP address or DN (in LDAP format).

Port: The server port to communicate with the directory.

User: The user's name.

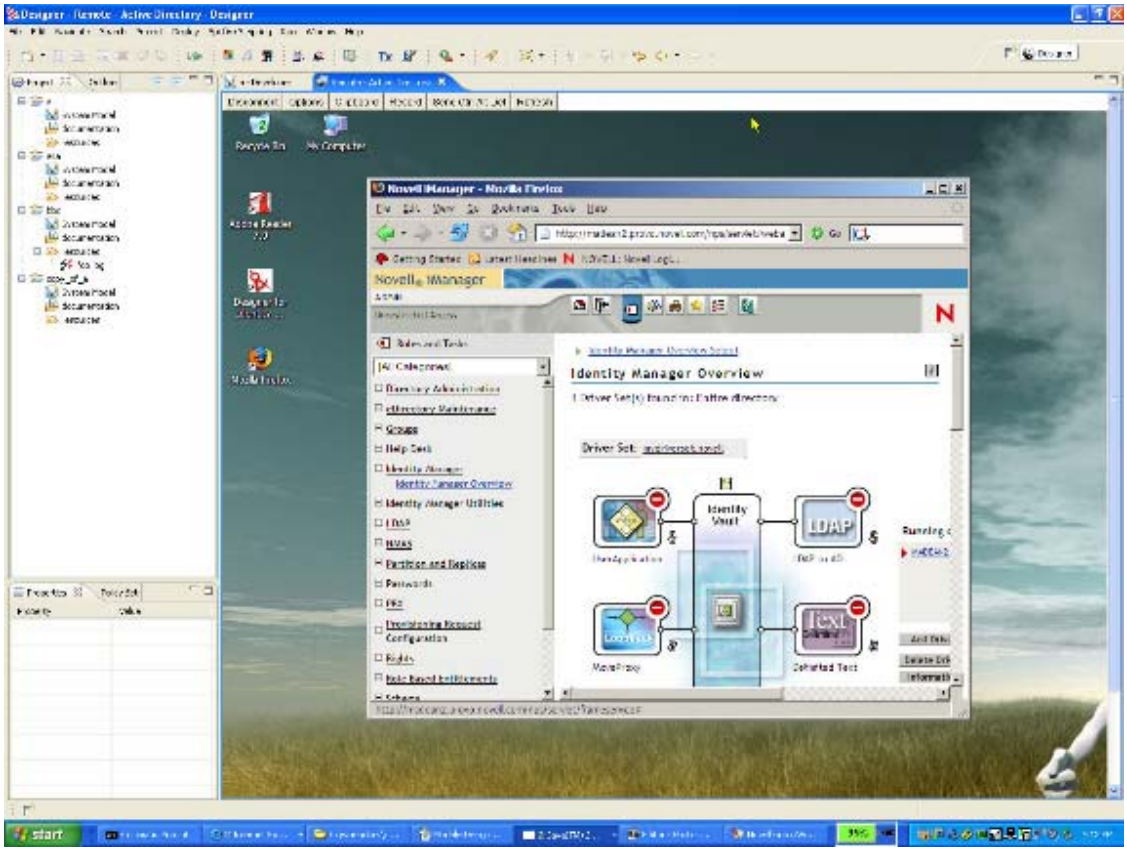
Password: The user's password.

VNC

From within Designer, you can view the desktop of the machine that is running your applications, and remotely control that desktop by interacting with it. This feature enables you to administer users or your applications with the native tools of that system, from one location.

This functionality is hosted in an embedded editor inside Designer. You can have multiple remote control sessions with different systems, all open at the same time.

Figure 4-19 *A Remote Desktop*



eDirectory / NDS

You can configure connectivity to eDirectory and legacy NDS applications. This is similar to configuring an LDAP connection, but uses native eDirectory and NDS protocols instead of LDAP.

Legacy NDS applications display this tab as *NDS*, while eDirectory application display it as *eDirectory*.

Host: The server's IP address or DN (in eDirectory/NDS format).

Port: The server port to communicate with the directory.

User: The user's name.

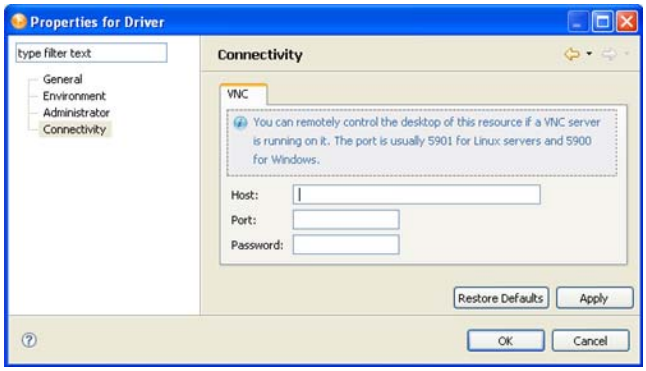
Password: The user's password.

Configuring a Remote Connection

To remotely control a desktop, the machine that is running your application needs to have a VNC (virtual network computing) server installed and running. You can usually download a VNC server for free from the Internet.

You can easily configure any system or design element in Designer for this feature by editing any application or design element:

1. Right-click an application or design element.
2. Select *Properties* > *Connectivity*.



3. On the *VNC* tab, type the authentication information.

   Host: The DN (for example, server33.houston.company.com) of the server where the VNC is running.

   Port: Typically 5901 for Linux servers or 5900 for Windows.

   Password: The password to the VNC server.
4. Click *OK*.

Customizing the Viewer

A toolbar at the top of the desktop viewer enables you to configure the following:

- Encoding type (RAW, RRE, CoRRE, Hextile, Zlib, Tight).

  The default is Tight.
- Compression level.
- JPEG Image Quality (0 - 9).

  The default is 6.
- Cursor shape updates.

  The default is Enable.
- Use CopyRect.

  The fault is Yes.
- Mouse buttons 2 and 3.

  The default is Normal.
- View only.

The default is No, so that you can interact with the desktop.

- Clipboard.
- Record session and save to file.
- Send Ctrl+Alt+Delete.
- Refresh.

For more information, see TightVNC documentation.

4.11.5 AD Domain

You can capture information about an Active Directory application. This information is useful if you want Document Generator to include this information when you document the project.

If you provided information in the LDAP settings, Designer populates the AD Domain fields.

4.12 Configuring Policies

- Section 4.12.1, “Editing a Policy Name,” on page 125
- Section 4.12.2, “Viewing References,” on page 125
- Section 4.12.3, “Editing a Credential Provisioning Policy,” on page 125

4.12.1 Editing a Policy Name

1. In the Outline view, right-click a policy or rule.
2. Select *Properties*.

   The *General* setting displays by default.
3. Edit the name in the *Policy Name* field, then click *OK*.

4.12.2 Viewing References

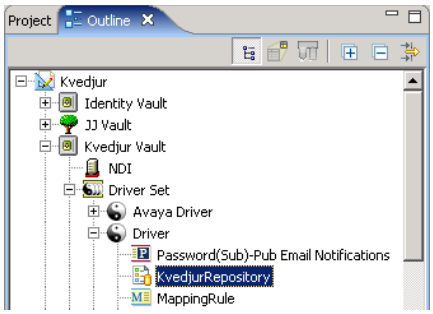
The References page lists policy sets and policies that reference the policy listed in the General page. To view the references to this policy:

1. In the Outline view, right-click a policy or rule.
2. Select *Properties* > *References*.

Linkage is how the policies reference each other. In Identity Manager versions earlier than 3.5, linkage determined the order that policies were executed. To change the linkage, use the Policy Builder.

4.12.3 Editing a Credential Provisioning Policy

1. In the Outline view, right-click a credential provisioning policy (for example, a repository).



2 Select *Properties*.

3 Edit the policy name, then click *OK*.

4.13 Adding Prompts to a Driver Configuration File

Several node types are defined for driver configuration files. These extensions were made to support the following:

- Prompting once for a value that is used repeatedly throughout a single driver configuration file.
- Prompting once for a value that is used across multiple driver configuration files, as part of the Import Drivers Wizard.
- Allowing the user to select a value from a drop-down list of values.
- Global modification of the driver configuration file as per a contained XSL stylesheet.
- Built-in variables that can be referenced without declaring them to access information about the driver and its environment (a tree name, driver set name, driver set DN, server name, server DN, driver name and driver DN).
- The ability to “layer” prompts. It is possible to ask the user multiple sets of questions, with the second and later sets being controlled by the user's responses to prior sets. For more information, refer to the FlexiblePrompts document.

The primary new node types are variable-decl, variable-ref, and xsl-modify.

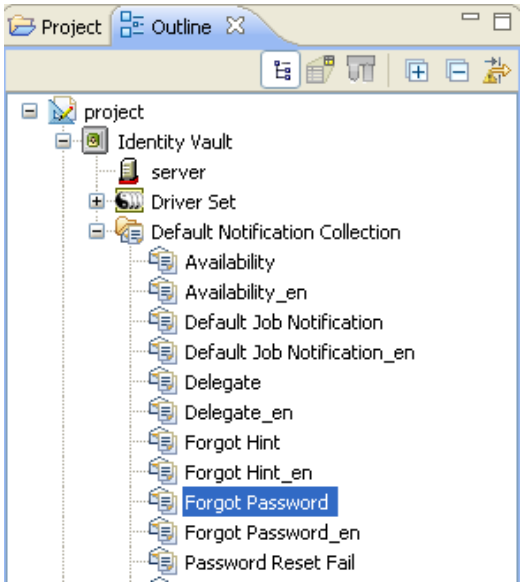
Table 4-22 *New Node Types*

New Node Type	Description
variable-decl	Allows you to define driver configuration variables that are prompted for (optionally) and replaced into a driver configuration file during its import. Multiple variable-decl blocks can be used to define a “layered” set of prompts. Refer to the FlexiblePrompts document for more information.
variable-ref	Used to reference a variable defined in a variable-decl within your driver configuration files.
xsl-modify	Used to globally modify the driver configuration file after all variables (and prompting) have been resolved. The contents of this node are extracted and used as an XSL style sheet that is applied to the patched driver configuration file.

For information on adding prompts to a sample configuration file, see “Editing Driver Configuration Files,” which is an appendix in the Novell Identity Manager Administration Guide (http://www.novell.com/documentation/idm35/index.html).

4.14 Editing a Notification Template Name

1 In the Outline view, expand *Default Notification Collection*.



2 Right-click a notification template (for example, Forgot Password), then select *Properties*.

3 Edit the name, then click *OK*.

For additional information, see Chapter 10, “Setting Up E-Mail Notification Templates,” on page 215.

4.15 Synchronizing Passwords

To view or edit password synchronization, use the Dataflow editor. See Section 7.2.1, “Filtering Views,” on page 174 and “Synchronizing Passwords” on page 174.

Managing Identity Manager Versions

5

Your environment might have versions of Identity Manager earlier than version 3.6, or you might have a mixture of different versions of Identity Manager.

Before Designer 2.0, if you configured and wrote policies for an earlier version of Identity Manager in your environment, you might have encountered the following issues:

- You could easily build a solution that would not deploy.
- You did not know which features worked in one environment versus another environment.

To solve these issues, Designer 3.5 tracks versions of the following objects:

- Identity Manager engines
- Identity Vaults (trees)
- Drivers

As you use Designer 3.5, you see only the UI of features that apply to the version that you are working on. Project Checker and Deploy ensure that what you have configured is supported in the target environment.

- Section 5.1, “Key Differences in Identity Manager Versions,” on page 130
- Section 5.2, “Changing the Identity Manager Version,” on page 130
- Section 5.3, “Tracking Versions of Identity Manager,” on page 131
- Section 5.4, “Information in the Driver Configuration Wizard,” on page 133
- Section 5.5, “Checking Projects for Version Issues,” on page 134
- Section 5.6, “Adjusting the UI Based on the Version Number,” on page 135

5.1 Key Differences in Identity Manager Versions

Table 5-1 *Differences in Identity Manager Versions*

Identity Manager 3.0.1	Identity Manager 3.5	Identity Manager 3.6

◆ New DirXML Script actions
◆ New Log Events:
  ◆ Add Value (on add)
  ◆ Get Named Password
  ◆ Reset Attributes
  ◆ Add Value (on modify)
◆ Resource Objects were added:
  ◆ Credential Provisioning Applications
  ◆ Credential Provisioning Repository Objects
  ◆ Generic Resource Objects
	◆ New object types were added:
	  ◆ ECMAScript Objects
	  ◆ Jobs
	  ◆ Mapping Table Resource Objects
	  ◆ Resource Libraries
	◆ New Policy Linking capabilities where a policy can be in multiple lists
	◆ Many new DirXML Script actions, conditions, tokens, and verbs
	◆ Ability for DirXML Script to nest conditions
	◆ Driver-scoped local variables in DirXML Script that let you refer to variable outside of the policy
		◆ Support for 64-Bit Operating Systems
		◆ New installation Program
		◆ New Driver Configuration Files
		◆ Driver Health Monitoring
		◆ New ID Provider Driver
		◆ Reciprocal Attribure Mapping
		◆ Addtional DirXML Script Elements
		◆ Nested Group Support
		◆ User Application
		◆ Designer 3.5

5.2 Changing the Identity Manager Version

You can both import and deploy to all versions of Identity Manager that shipped since Identity Manager 2.0, up to and including Identity Manager 3.6. You can also import from DirXML 1.x environments.

Because versions earlier than Designer 2.0 M5 did not track Identity Manager versions, those earlier projects do not have version information.. When you convert an earlier project, Designer 3.5 defaults the Identity Manager version numbers to 3.5. During conversion, Designer informs you that this default is being applied.

You can change this version number by doing either of the following:

- In the Outline view, right-click the Server object, select *Properties*, then select from the *Identity Manager Version* drop-down list.
- In the Modeler, select an Identity Vault, click *Window* > *Preferences*, expand *Novell* and select *Identity Manager*, then select a version from the drop-down list.

You can also find information on upgrades, information on downgrades, and a link to a help topic. This information explains the key differences between versions of Identity Manager.

When you import into a new server (or create a server based on a server that you have browsed to in the directory), the new server inherits the imported version of Identity Manager.

If you do a live update in the server properties page, Designer updates the server to the current version of Identity Manager that is in the target environment.

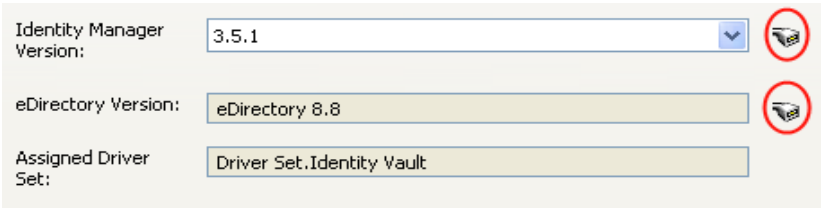
5.3 Tracking Versions of Identity Manager

Designer 4.0 tracks the Identity Manager version. Filtering functionality is based on this version information. When multiple servers are associated to a driver set, Designer calculates an “effective engine version.” This version is the earliest Identity Manager version in the driver set.

If you want to use the latest Identity Manager 3.5.1 features, it is important that all servers belonging to the driver set are upgraded to 3.5.1. This version can be manually upgraded or downgraded from the server properties page.

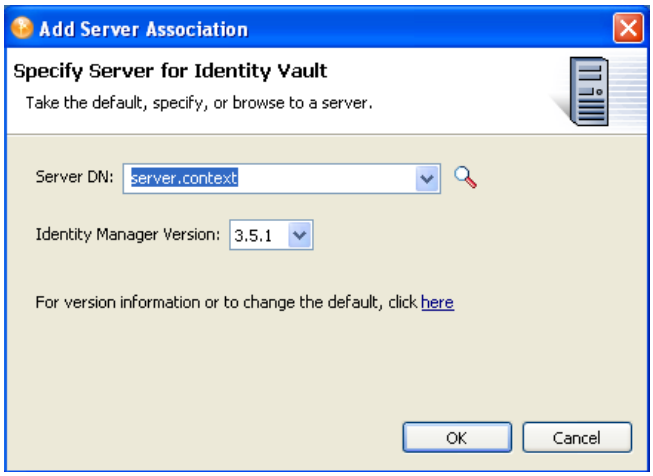
Additionally, live update buttons retrieve current Identity Manager and eDirectory version information on the server properties page.

Figure 5-1 *Live Update Buttons*



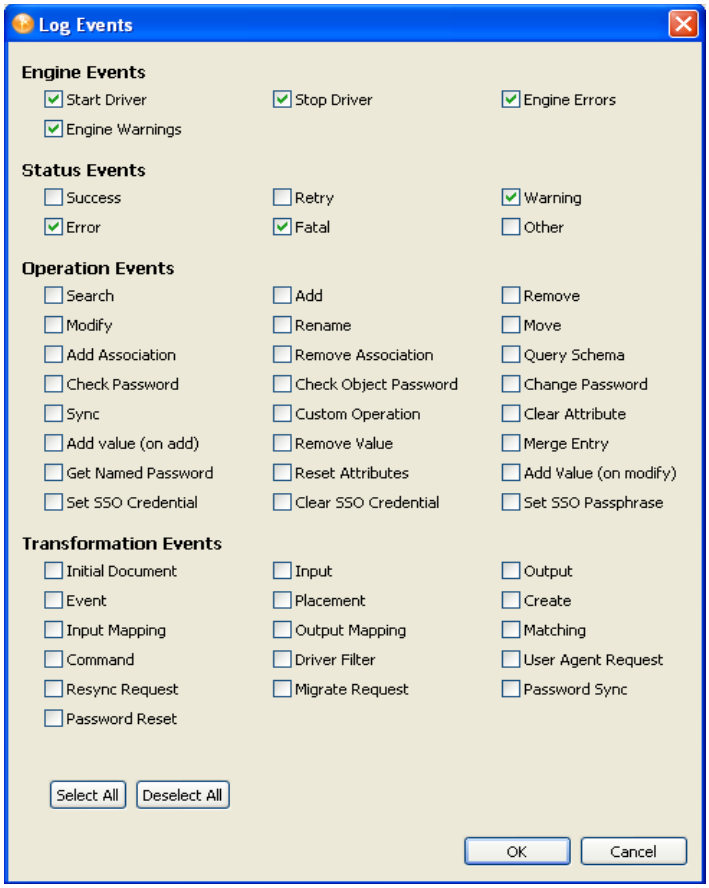
The *Add Server* dialog box allows you to specify version information when an Identity Vault is created.

Figure 5-2 *The Add Server Dialog Box*



The Driver Set Log Level and Driver Log Level propety pages have dynamic version widgets next to any log event that is not supported by your effective Identity Manager version. Designer 2.0 M5 added support for several Identity Manager log events: Set SSO Credential, Clear SSO Credential and Set SSO Passphrase.

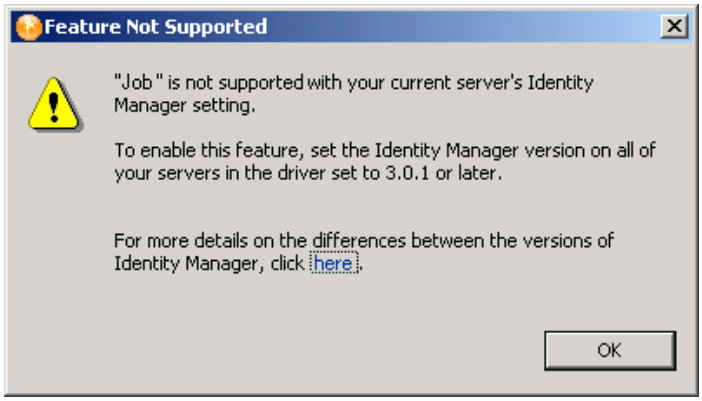
Figure 5-3 *Log Events*



For more information about these events, see the Credential Provisioning Policies (http://www.novell.com/documentation/idm35/index.html?page=/documentation/idm35/policy_credprov/data/bookinfo.html) documentation.

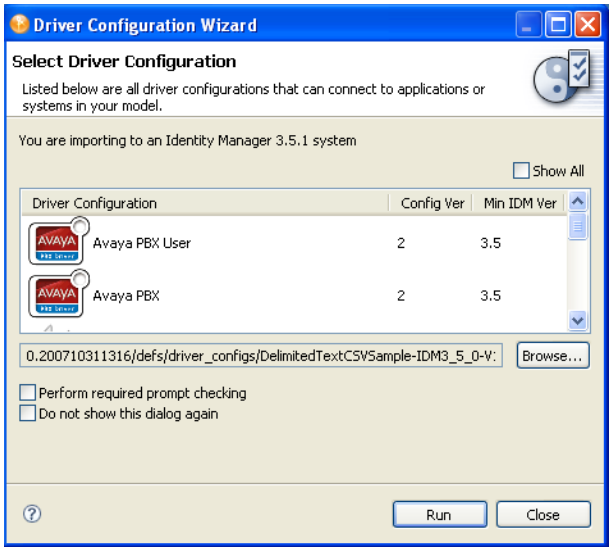
The following figure illustrates an unsupported log event:

Figure 5-4 *Message Concerning an Identity Manager Version*



5.4 Information in the Driver Configuration Wizard

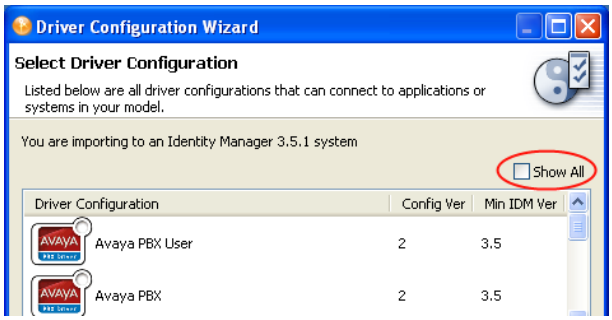
Figure 5-5 *The Driver Configuration Wizard*



In Designer 3.5, the Driver Configuration Wizard provides more information than previous milestone releases, including the following:

- The engine version that you are importing into. This information is taken from the current project. You control the version number.
- A descriptive name of the driver configuration (no longer just the filename).
- The version of the configuration as a single (undelimited) version number.
- The minimum required engine version for this configuration to run.
- The full filename of the selected list item. This name is below the list. It is displayed there for transparency.
- A check box that indicates possible unrecommended or incompatible configuration files.

Figure 5-6 *A Deselected Show All Check Box*



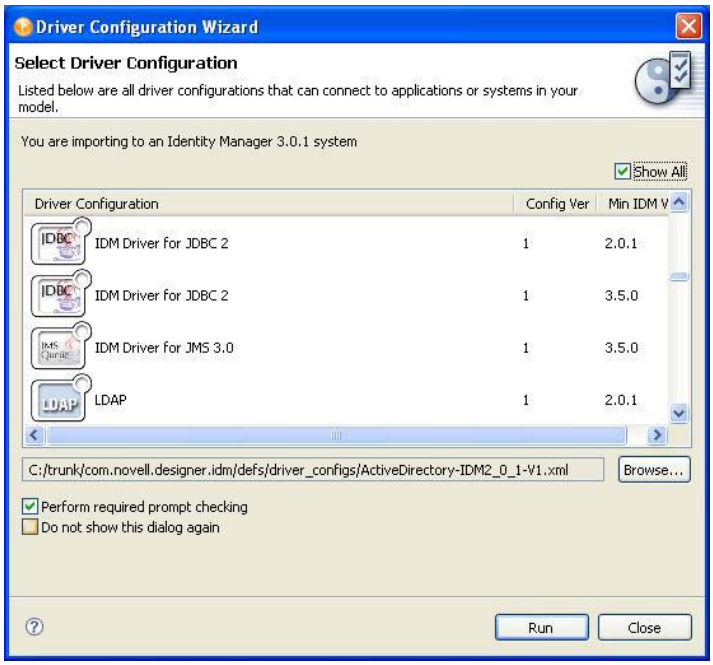
By default, the *Show All* check box is deselected if unrecommended or possibly incompatible configuration files are available. If all available driver configuration files are recommended and guaranteed compatible, the check box is dimmed and selected, indicating that all available options are displayed.

A deselected *Show All* check box implies the following:

- Additional driver configuration files are available but they are not recommended.
- The additional driver configuration files are probably incompatible with the engine version that you are importing to.

In the following figure, the *Show All* check box is selected.

Figure 5-7 *A Selected Check Box in the Wizard*



The list now contains many more items than were displayed when the check box was deselected. These new items were previously hidden because the minimum required engine version for them is 3.5. Because the user is importing to 3.5, the configuration might be incompatible.

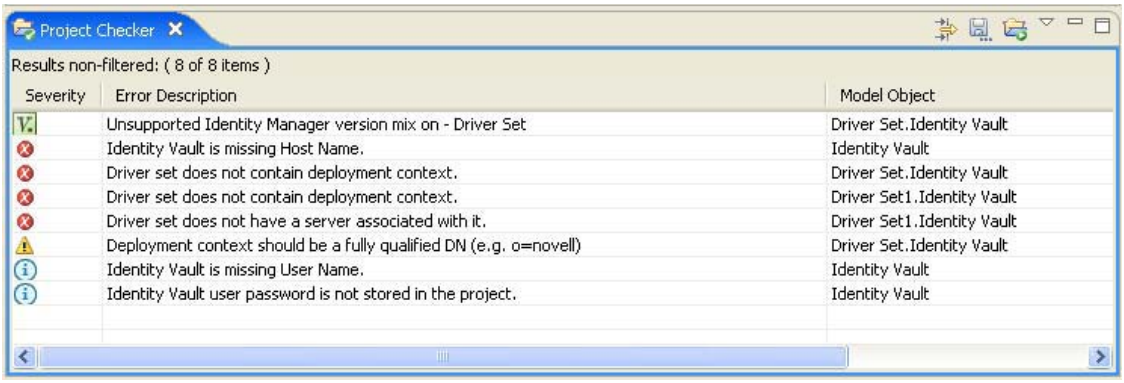
5.5 Checking Projects for Version Issues

A full suite of project checks makes sure that what you have configured makes sense for your target environment and can be successfully deployed. Designer's UI blocks the creation of unsupported objects and hides features based on the version number. Nevertheless, unsupported actions might still occur through a few “back-door” methods, such as copying and pasting, importing, and downgrading your server after you have configured for a newer environment.

In all of these methods, Project Checker catches the problems.

For example, for policy libraries to work, all of the servers on a given driver set need to be at the same Identity Manager 3.5 version. Project Checker catches problems like this where you might have an unsupported mix of servers. In this case, the project check results would look like the following figure:

Figure 5-8 *Project Checker*



Version problems are sorted to the top and have a version icon. If you double-click the item, you get more details about the problem and how to resolve it.

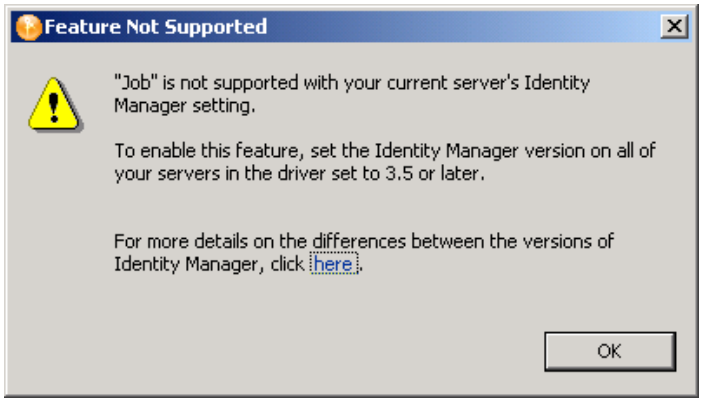
5.6 Adjusting the UI Based on the Version Number

Designer displays and enables or disables capabilities based on the version of Identity Manager that is associated with the Identity Manager engine. For example, if you edit a policy that is associated with a server that uses Identity Manager 3.5, Policy Builder shows you all of the new actions, conditions, verbs, and tokens that ship with that release. This feature lets you try out the next version of Identity Manager before it is even released.

Also, if you set the server to Identity Manager 3.5 (or earlier), you get the previous version of Policy Builder that Designer has shipped with in the past.

If you try to create an object that is not supported by your server version, a prompt tells you that this action isn't supported. For example, Identity Manager 3.5 introduces the concept of Jobs, Mapping Tables, and Policy Libraries. If you try to create one of these objects on a 3.5 server, you see the following message:

Figure 5-9 *Prompt: Feature Not Supported*



Future milestones of Designer will continue to evolve the UI to better handle version differences.

Managing the Schema

6

Designer includes a copy of the base Identity Vault schema, which is stored in the `BaseIVSchema.xml` file. This file is located in `\Designer\eclipse\plugins\com.novell.core.datatools_x.x.x.x\defs\schema`, where *x.x.x.x* represents the specific Designer build.

Do not directly modify `BaseIVSchema.xml`. Instead, use Designer to add the schema information from this file into your project. The Manage Schema tool allows you to change the schema as part of the project without modifying the original `BaseIVSchema.xml` file.

You can add, delete, rename, and modify classes and attributes in the Identity Vault schema. You can import the Identity Vault schema from the production environment, or use the default schema. After modifying the schema, you can deploy it into the production Identity Vault.

WARNING: If you do not have a good understanding of how the Identity Vault schema works, changing the default schema can cause data corruption. If you modify classes or attributes and then deploy the modified schema into a tree where these classes are in use, one of the following problems can occur:

- Those objects can become unknown.
- Synchronization errors can occur.

To understand the basics of the schema, see "Managing the Schema" (http://www.novell.com/documentation/edir88/index.html?page=/documentation/edir88/edir88/data/a4a9bz0.html) in the online documentation for Novell eDirectory 8.8.

If you subscribe to LogicSource, see Novell LogicSource for eDirectory (http://support.novell.com/subscriptions/articles/novell_logicsource.html) for additional information. LogicSource is a subscription-based service that Novell provides to its customers.

- Section 6.1, "Using the Manage Schema Tool," on page 137
- Section 6.2, "Creating Classes and Attributes," on page 147
- Section 6.3, "Modifying the Schema," on page 149
- Section 6.4, "Deploying the Schema into the Identity Vault," on page 150
- Section 6.5, "Exporting the Schema to a File," on page 152
- Section 6.6, "Importing the Schema," on page 155
- Section 6.7, "Managing a Copy of an Application Schema," on page 160
- Section 6.8, "Mapping Identity Vault to an LDAP Schema," on page 162
- Section 6.9, "Comparing the Schema," on page 163

6.1 Using the Manage Schema Tool

To open the Manage Schema tool, right-click an Identity Vault object (in the Modeler or Outline View), then select *Manage Vault Schema*.

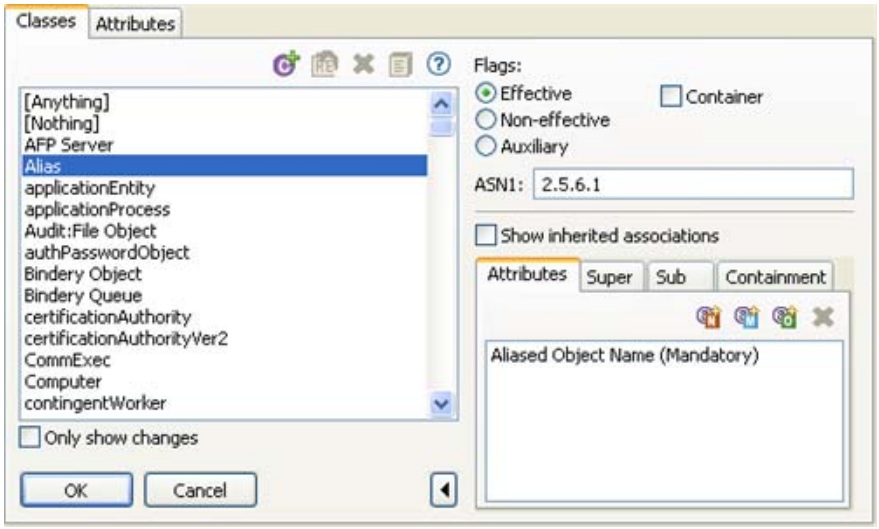
If a custom schema in the production environment needs to be tested, you can import the schema into Designer. After you have tested and modified the schema, you can deploy it into the production environment. For information about importing schema, see Section 6.6, “Importing the Schema,” on page 155.

The Manage Schema tool lets you add, delete, rename, and modify classes and attributes in the Identity Vault schema. The class information and the attribute information is organized into separate tabs in the Manage Schema tool.

- Section 6.1.1, “The Classes Tab,” on page 138
- Section 6.1.2, “The Attributes Tab,” on page 141

6.1.1 The Classes Tab

From the *Classes* tab, the Manage Schema tool lets you add, delete, rename, and modify schema classes.



The *Classes* tab includes the following components:

- “Classes List Toolbar” on page 138
- “Only Show Changes” on page 139
- “ASN1” on page 139
- “Flags” on page 139
- “Show Inherited Associations” on page 140
- “Associations List” on page 140

Classes List Toolbar

The Classes list includes the following tools:

Table 6-1 *Classes List Toolbar*

Icon	Description
Add Class	Launches the New Class Wizard to create a new Identity Vault class.
Rename Class	Renames any non-base class. You cannot rename base classes.
Delete Class	Deletes any non-base class. You cannot delete base classes.
Schema Notes	Adds descriptive notes to any non-base class.You cannot add notes to base classes.

Only Show Changes

The *Only show changes* checkbox is below the Classes list. When selected, the Classes list displays only those classes that are not part of the base schema, as defined in `BaseIVSchema.xml`. If no non-base classes exist, the Classes list is empty.

Deselect Only Show Changes to see a complete list of base and non-base classes in the Identity Vault schema.

ASN1

Specifies the class's "Abstract Syntax Notation number One" ID. The ASN1 ID is important is you plan to make the schema definition publicly available.

If you register your schema definition with Novell, Novell assigns your class an ASN1 ID. This unique identifier eliminates the possibility of schema collisions caused by duplicate schema names with different definition structures.

For more information about ASN1, visit the ASN.1 Consortium (http://www.asn1.org/).

Flags

The Flags options let you modify the class type:

Table 6-2 *Supported Class Types*

Flag	Description
Effective	You can create an instance of the defined object in the Identity Vault.
Noneffective	Only used to define other classes. You cannot create an object of a noneffective class.
Auxiliary	Combines attributes to be added to other classes by extending the object class attribute.
Container	Sets the object as a container object instead of a leaf object. If it is set to container, this object can contain other objects.

Show Inherited Associations

The Show Inherited Associations checkbox determines whether the Associations list displays all attributes associated with a class. When selected (the default), the Associations list displays both assigned and inherited attributes. When deselected, the Associations list displays only assigned attributes.

NOTE: When you select *Show Inherited Associations*, you cannot delete entries from the Associations list.

Associations List

The Associations List displays the classes and attributes associated with the selected class. The Associations List includes four tabs, each with a toolbar.

Attributes: The *Attributes* tab displays the attributes associated with the selected class. It also identifies if attributes are mandatory or naming. All unmarked attributes are optional.

The *Attributes* tab includes the following tools:

Class Field	Description
Add Naming	Adds a naming attribute association to the selected class.
Add Mandatory	Adds a mandatory attribute association to the selected class.
Add Optional	Adds an optional attribute association to the selected class.
Delete	Deletes an attribute association from the select class.

Super: The *Super* tab displays the classes from which the selected class inherits attributes. A class that another class inherits from is called a superclass.

A class can inherit attributes from more than one super class. The super class that every class inherits from is Top. No class exists above Top. For example, Group inherits directly from Top, but User inherits from Organizational Person. Organizational Person inherits from Person. Person inherits from ndsLoginProperties, and ndsLoginProperties inherits from Top.

The *Super* tab includes the following tools:

Class Field	Description
Add Superclass Association	Adds a superclass association to the selected class.
Delete	Deletes a superclass association from the selected class.

Sub: The *Sub* tab displays all classes that inherit from the selected class. If the Sub tab is empty, no classes inherit from the selected class.

The *Sub* tab includes the following tools:

Class Field	Description
Add Subclass Association	Adds a subclass association to the selected class.
Delete	Deletes a subclass association from the selected class.

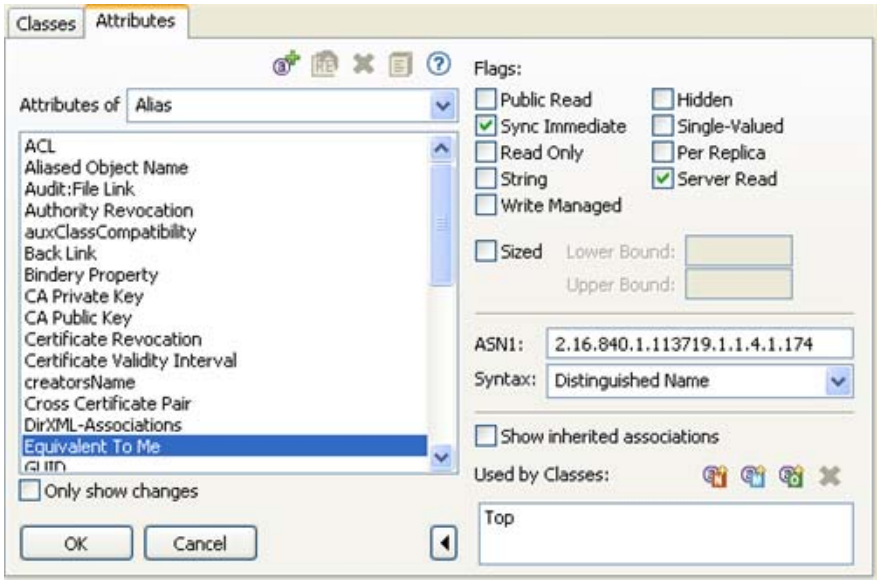
Containment: The Containment tab displays the container classes that can contain the selected class. For example, if you select the class Group, the Manage Schema tool lists the domain, Organization, and Organizational Unit classes, which can contain the Group class.

The *Containment* tab includes the following tools:

Class Field	Description
Add Containment Class Association	Adds a containment class association to the selected class.
Delete	Deletes a containment class association from the selected class.

6.1.2 The Attributes Tab

From the *Attributes* tab, the Manage Schema tool lets you add, delete, rename, and modify attributes associated with schema classes.



The *Attributes* tab includes the following components:

- “Attribute List Toolbar” on page 142
- “Only Show Changes” on page 142
- “Flags” on page 142

- “ASN1” on page 143
- “Syntax” on page 143
- “Show Inherited Associations” on page 146
- “Associations List” on page 146

Attribute List Toolbar

The Attributes list includes the following tools:

Table 6-3 *Classes List Toolbar*

Icon	Description
Add Attribute	Launches the New Class Wizard to create a new attribute.
Rename Attribute	Renames the selected non-base attribute. You cannot rename base attributes.
Delete Class	Deletes the selected non-base attribute. You cannot delete base attributes.
Schema Notes	Adds descriptive notes to any non-base attribute.You cannot add notes to base classes.

Only Show Changes

The *Only show changes* checkbox is below the Attributes list. When selected, the Attributes list displays only those attributes that are not part of the base schema, as defined in `BaseIVSchema.xml`. If no non-base attributes exist, the Attributes list is empty.

Deselect Only Show Changes to see a complete list of base and non-base attributes in the Identity Vault schema.

Flags

Attribute flags specify the information that is stored in the attribute and limit the list of acceptable operations that the Identity Vault, and eDirectory clients, can perform on the attribute.

Table 6-4 *Supported Attribute Flags*

Constraint	Description
Public Read	Allows anyone to read this attribute without the read privilege specifically assigned. You can’t use inheritance masks to prevent an object from reading attributes with this constraint.
Sync Immediate	When the attribute is modified, it is synchronized immediately to all of the servers in the replica ring.
Read Only	Only the eDirectory server process can read this attribute.
String	Allows only string information to be stored in the attribute.

Constraint	Description
Write Managed	Explicit rights are granted before this attribute can be changed. In order to modify this attribute, users must have managed rights on the object to change the attribute.
Hidden	Only the eDirectory server process can read this attribute.
Single Valued	Allows one value to be stored in the attribute.
Per Replica	Allows one value to be stored in the attribute.
Server Read	The attribute can be read by an NCP™ server object even though the right to read is not inherited or explicitly granted. The NCP server object is always able to read this attribute, regardless of the rights granted in the ACL.
Sized	Limits the range of values supported by the attribute to some subset of those supported by the attributes data type. For example, you might restrict an Integer attribute to only accept values between 1 and 100.

ASN1

Specifies the attribute's "Abstract Syntax Notation number One" ID. The ASN1 ID is important is you plan to make the schema definition publicly available.

If you register your schema definition with Novell, Novell assigns your attribute an ASN1 ID. This unique identifier eliminates the possibility of schema collisions caused by duplicate schema names with different definition structures.

For more information about ASN1, visit the ASN.1 Consortium (http://www.asn1.org/).

Syntax

An attribute syntax defines a standard data type that an attribute uses to store its values in the Identity Vault. Each attribute must have a syntax. Table 6-5 describes the available syntaxes for Identity Vault attributes.

Table 6-5 *Syntax for Attributes*

Syntax	Description
Back Link	The *remoteID* field identifies the backlinked object on the server, and the *objectName* field identifies the server holding an external reference.
Boolean	Two Boolean attributes match for equality if they are both True or both False. True is represented as one (1), and False is represented as zero (0). Any attribute defined by using this syntax is single valued.
Case Exact String	Attributes using this syntax can set size limits. Two Case Exact Strings match for equality when they are of the same length and their corresponding characters are identical.

Syntax	Description
Case Ignore List	Two Case Ignore Lists match for equality if the number of strings in each is the same, and all corresponding strings match. For two corresponding strings in the list to match, they must be the same length and their corresponding characters must be identical (according to the rules for case ignore strings).
Case Ignore String	Used in attributes whose values are strings and where the case (upper or lower) is ignored.
Class Name	Used to match two class names where the case (upper or lower) is ignored.
Counter	The attribute is single valued. The syntax is similar to Integer, except that any value added to an attribute is arithmetically added to the total, and any value deleted is arithmetically subtracted from the total.
Distinguished Name	The attribute is the distinguished name of the object up to 256 Unicode* characters. This is not case sensitive.
EMail Address	Used to match attributes whose values are e-mail addresses and whose lengths and corresponding characters are identical; however, it ignores case (upper and lower). Only the EMail Address attribute uses this syntax.
Facsimile Telephone Number	Facsimile Telephone Number values are matched based on the *telephone number* field. The rules for matching fax telephone numbers are identical to those for the Case Exact syntax except that all space and hyphen (-) characters are skipped during the comparison. Only the Facsimile Telephone Number attribute uses this syntax.
Hold	This syntax is an accounting quantity, which is an amount tentatively held against a subject's credit limit, pending completion of a transaction. In the wire format, the *Subject* field is the distinguished name of the object. The Identity Vault treats the Hold amount similarly to the Counter syntax, with new values added to or subtracted from the base total. If the evaluated Hold amount goes to 0 (zero), the Hold record is deleted.
Integer	The attribute is an integer. Attributes using this syntax can set size limits.
Interval	The Interval value is the number of seconds in a time interval.
Net Address	Stores the network address as a binary string. The string is the literal value of the address. It lists the type of communication protocol used.
Numeric String	Two numeric strings match for equality when they are of the same length and their corresponding characters are identical. It matches the digits 0-9 and spaces if they are contained in the numeric string.
Object ACL	An Object ACL value can protect either an object or an attribute. The protected object is always the one that contains the ACL attribute. If an ACL entry is to apply to the object as a whole, the protected attribute name should be left empty (NULL). If a specific attribute is to be protected, it should be named in the ACL entry.
Octet List	A presented octet list matches a stored list if the presented list is a subset of the stored list. Octet strings are so designated because they are not interpreted by the Directory. They are simply a series of bits with no Unicode implications. The length is the number of bits divided by 8 and rounded to the nearest integer. Thus, each octet represents eight bits of data. The number of data bits is always evenly divisible by 8.

Syntax	Description
Octet String	For two octet strings to match, they must be the same length and the corresponding bit sequence (octets) must be identical. When comparing two strings, the first pair of octets that do not match are used to determine the order of the strings. Octet strings are not Unicode strings.
Path	The string represented by the path field is compared for equality using the same rules that Case Exact String uses. That is, two paths match for equality when their lengths and corresponding characters, including case, are identical.
Postal Address	An attribute value for Postal Address is typically composed of selected attributes from the MHS Unformatted Postal O/R Address version 1 according to Recommendation F.401. The value is limited to 6 lines of 30 characters each, including a Postal Country Name. Normally the information contained in such an address could include a name, street address, city, state or province, postal code, and possibly a postal office box number depending on the specific requirements of the named object.
Printable String	The following characters are in the printable string character set. A...Z a...z 0...9 Space Character ‘ Apostrophe (Left Parenthesis) Right Parenthesis + Plus Sign , Comma - Hyphen . Period / Slash : Colon = Equal Sign ? Question Mark Two printable strings match for equality when they are the same length and their corresponding characters are identical. Case (upper or lower) is significant when comparing printable strings. For example, as printable strings, “Jones” and “JONES” do not match.
Replica Pointer	Each value of the replica pointer syntax is composed of five parts: ◆ The complete name of the server that stores the replica. ◆ A value describing a the capabilities of this copy of the partition: master, secondary, read-only, or subordinate reference. ◆ A value indicating the current state of the replica (new, dying, locked, changing state, splitting, joining, or moving). ◆ A number representing the replica. All replicas for a partition have a different number assigned when the replica is created. ◆ A referral that contains a count of the addresses and one or more network addresses that hints at the node where the server probably resides. Because servers are accessible over different protocols, the server might have an address for each supported protocol.
Stream	Streams are files of information. The data stored in a stream file has no syntax enforcement of any kind. It is purely arbitrary data, defined by the application that created and uses it. The attribute is single valued.

Syntax	Description
Telephone Number	The length of telephone number strings must be between 1 and 32. Two telephone numbers string match for equality when they are of the same length and their corresponding characters are identical. All spaces and hyphen (-) characters are skipped during the comparison.
Time	A time value consists of a whole number of seconds, where zero equals 12:00 midnight, January 1, 1970, UTC.
Timestamp	A Timestamp value contains three components: ◆ The *wholeSeconds* field consists of the whole number of seconds, where zero equals 12:00 midnight, January 1, 1970, UTC. ◆ The *replicaNum* field identifies the server that created the Timestamp. A replica number is assigned whenever a replica is created on a server. ◆ The *eventID* field is an integer that orders events occurring within the same whole-second interval. The event number restarts at one for each new second.
Typed Name	The syntax names an Identity Vault object and attaches two numeric values to it: ◆ The level of the attribute indicates the priority. ◆ The interval indicates the frequency of references. The the objectName or Distinguished Name identifies the Identity Vault object referred to by the Typed Name.
Unknown	Unknown syntax is used to stop the loss of data, if the Identity Vault database becomes corrupted. When an object becomes Unknown, there is information stored in this attribute that can allow the object to be recovered. This syntax is used by the Identity Vault.

NOTE: The information in Table 6-5 comes from the Novell LogicSource for eDirectory. LogicSource is a subscription-based service Novell provides to its customers. For more information about LogicSource, see Novell Technical Subscriptions (http://support.novell.com/subscriptions/articles/novell_logicsource.html).

Show Inherited Associations

The *Show Inherited Associations* checkbox determines whether the Associations list displays all classes associated with an attribute. When selected (the default), the Associations list displays both assigned and inherited classes. When deselected, the Associations list displays only assigned classes.

The schema allows for inheritance of other attributes from super classes. If you select this item, all attributes that are associated with a class, whether assigned or inherited, are listed. If you don’t select this item, only the assigned attributes are listed.

Used by Classes lists all classes that use the selected attribute. If you select *Show inherited associations*, the list includes classes that inherit the attribute.

Associations List

The Associations List displays the classes associated with the selected attribute. The Associations List toolbar lets you make changes to the classes associated with the attribute.

Class Field	Description
Add as Naming	Associates the selected attribute as a naming attribute to a class.
Add as Mandatory	Associates the selected attribute as a mandatory attribute to a class.
Add Optional	Associates the selected attribute as an optional attribute to a class.
Delete	Deletes the selected classes from the association list.

6.2 Creating Classes and Attributes

Designer allows you to create Identity Vault classes and attributes to fit the needs of your environment. You can test and use the new schema with the Identity Manager drivers in Designer before implementing it in the production environment.

- Section 6.2.1, “Creating Identity Vault Classes,” on page 147
- Section 6.2.2, “Creating Identity Vault Attributes,” on page 148

6.2.1 Creating Identity Vault Classes

- “Adding a Class” on page 147
- “Adding a Note” on page 148

Adding a Class

1. In the Modeler, right-click the Identity Vault, then select *Manage Vault Schema*.

   The *Classes* tab lists all classes that are defined in the schema and stored in Designer. For more information about the Classes tab, see Section 6.1.1, “The Classes Tab,” on page 138.
2. Select the *Add a Class* icon .
3. In the Create Class Name dialog box, specify the class name (for example, EmpInfo) and ASN1 ID (if applicable), then click *Next*.

   For more information about ASN1 IDs, see “ASN1” on page 139.
4. In the Class Flags dialog box, select the class type, then click *Next*.

   For information about the class type options, see Table 6-2 on page 139.
5. In the Class Inheritance dialog box, select the classes from which the new class inherits, then click *Next*.

   Select one or more classes in the *Available classes* list and move them to the *Inherited classes* list with the right-arrow button. Remove classes from the *Inherited classes* list using the left-arrow buttons.
6. In the Mandatory Attributes dialog box, select the mandatory attributes, then click *Next*.

   The inherited attributes displayed in the *Inherited mandatory attributes* pane depend upon the classes from which the new class inherits.

7 In the Optional Attributes dialog box, select optional attributes, then click *Next*.

The *Inherited optional attributes* pane lists default optional inheritances.

8 In the Naming Attributes dialog box, select the naming attributes, then click *Next*.

The Identity Vault schema allows for inheritance from other classes. A class that another class inherits from is called a superclass. A class can inherit attributes from one or more superclasses.

Every class inherites from the superclass Top. No class exists above Top. For example, Group inherits directly from Top, but User inherits from Organizational Person. Organizational Person inherits from Person. Person inherits from ndsLoginProperties, and ndsLoginProperties inherits from Top.

9 In the Containment Classes dialog box, select the containment classes for the new class, then click *Next*.

This specifies the types of container classes that can contain the new class. For example, if you select the class Group, the Manage Schema tool lists Domain, Organization, and Organizational Unit classes as containment classes for the Group class

10 In the New Class Summary, review the new class information, then click *Finish*.

The new class appears in the Classes pane.

11 Click *OK* to save changes and close the Manage Schema tool.

Adding a Note

Designer allows you to add notes about any class you create. The information is stored as `desc` in the `.ldif` file and as a note in the `.sch` file.

1 Select the class you want to add a note to, then click the *Schema Notes* icon .

2 Type the note in the window, then click *OK*.

6.2.2 Creating Identity Vault Attributes

To create a new Identity Vault attribute:

1 In the Modeler, right-click the Identity Vault, then select *Manage Vault Schema*.

2 Select the *Attributes* tab.

The Attributes list displays all attributes that are defined in the schema and stored in Designer. You can view all attributes at once, or view the attributes associated with a specific class by selecting a class from the drop-down list.

For more information about the components of the *Attributes* tab, see Section 6.1.2, “The Attributes Tab,” on page 141.

3 Select the *Add an Attribute* icon .

4 In the Create Attribute Name dialog box, specify the attribute name (for example, EmpID) and an ASN1 ID, if applicable, then click *Next*.

For more information about the ASN1 ID, see “ASN1” on page 143.

5 In the Attribute Syntax dialog box, select the proper attribute syntax, then click *Next*.

An attribute syntax defines a standard data type that an attribute uses to store its values in the Identity Vault. Each attribute must have a syntax. See Table 6-5 on page 143.

6 In the Attribute Flags dialog box, select the flags for the attribute, then click *Next*.

Attribute flags constrain the information that is stored in the attribute, and the list of acceptable operations that the Identity Vault, and Identity Vault clients, can perform on the attribute. For more information about attribute flags, see "Flags" on page 142.

7 In the New Attribute Summary dialog box, review the new attribute information, then click *Finish*.

The new attribute appears in the Attributes list.

8 Click *OK* to save changes and close the Manage Schema tool.

6.3 Modifying the Schema

Designer allows you to modify the Identity Vault schema. The following sections describe fields and definitions used in the Manage Schema tool for classes and attributes.

- Section 6.3.1, "Deleting Schema Definitions," on page 149
- Section 6.3.2, "Modifying Classes or Attributes," on page 149
- Section 6.3.3, "Renaming Schema Definitions," on page 150

6.3.1 Deleting Schema Definitions

You can delete an extended schema definition. You cannot delete base schema elements. If you select a base schema class or attribute, the *Delete* icon is disabled.

1 In the Modeler, right-click an Identity Vault, then select *Manage Schema*.

2 Select the class or attribute that you want to delete, then click the *Delete* icon ✖.

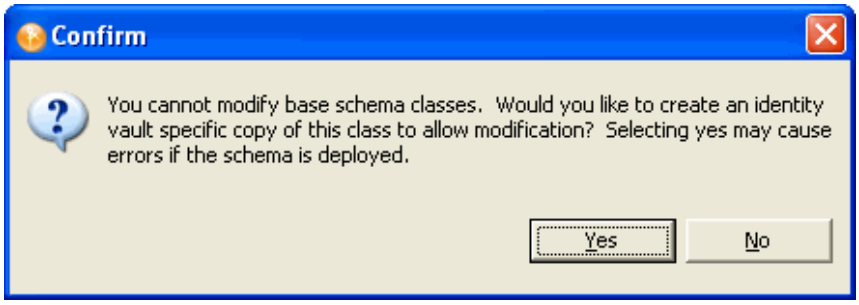
6.3.2 Modifying Classes or Attributes

1 In the Modeler, right-click an Identity Vault, then select *Manage Vault Schema*.

2 Select the class or attribute that you want to modify.

3 Modify the class or attribute as desired.

If you select a base schema class or attribute, the following pop-up message appears:



It is best to modify only the extended schema and not the base schema. Modifying the base schema can cause data corruption and synchronization errors.

6.3.3 Renaming Schema Definitions

You can rename extended schema definitions. You cannot rename any base schema classes or attributes. If you select a base schema item, the *Rename* icon is dimmed, indicating it is unavailable.

- "Renaming a Class" on page 150
- "Renaming an Attribute" on page 150

Renaming a Class

1. In the Modeler, right-click an Identity Vault, then select *Manage Vault Schema*.
2. In the *Class* page, select a class that you want to rename, then click the *Rename Class* icon .
3. In the Rename Class dialog box, specify the new class name, then click *OK*.

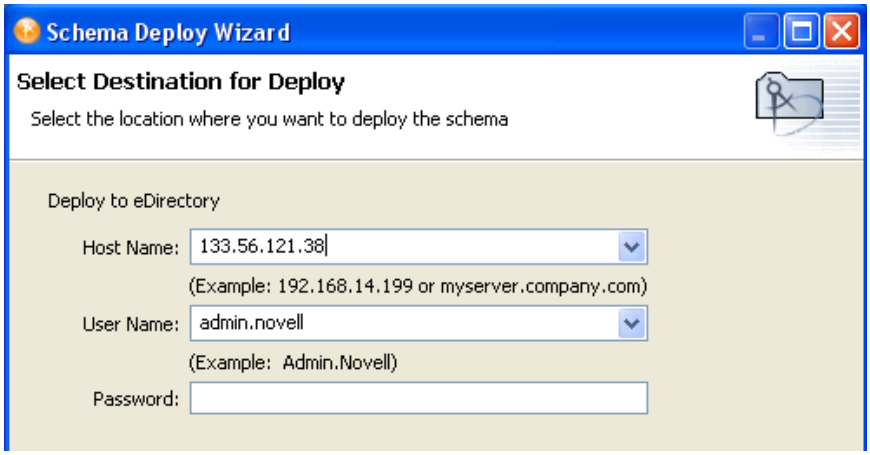
Renaming an Attribute

1. In the Modeler, right-click the *Identity Vault* icon, then select *Manage Vault Schema*.
2. Select an attribute you want to rename in the *Attribure* tab, then click the *Rename an Attribute* icon .
3. In the Rename Attribute dialog box, specify the new attribute name, then click *OK*.

6.4 Deploying the Schema into the Identity Vault

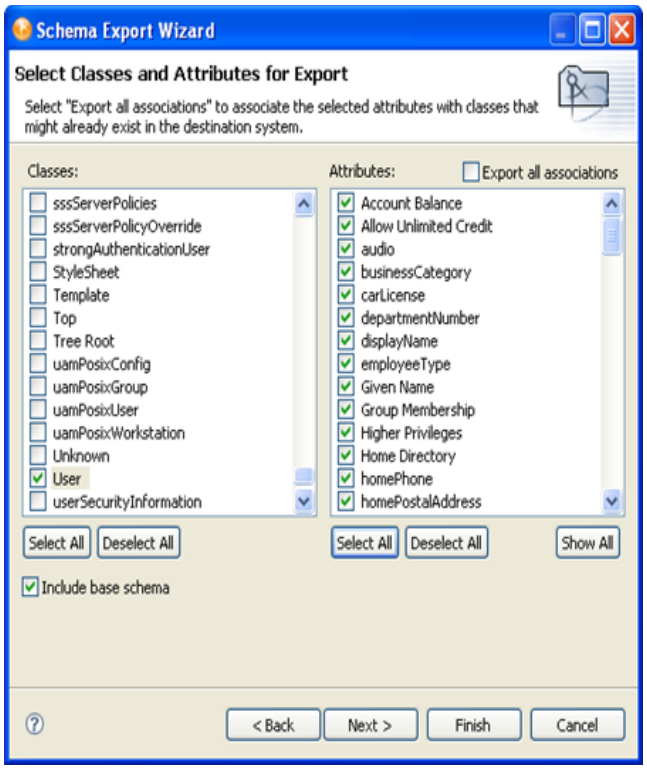
After the Identity Manager driver is tested with the new schema, you can deploy the modified schema into the Identity Vault.

1. In the Modeler, select the Identity Vault.
2. Select *Live* > *Schema* > *Deploy*.
3. Specify the *Host Name*.



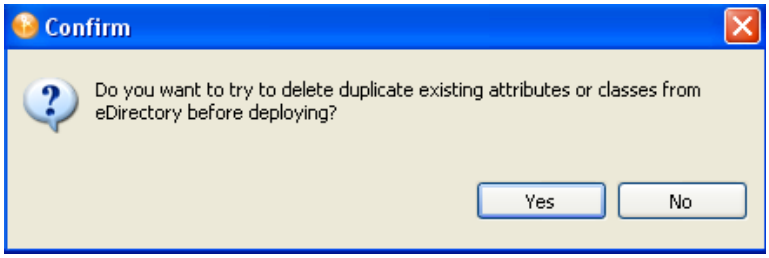
The host name can be the server's IP address or the DNS name of the server.

4. Specify the *User Name*, which must be a user with administrative rights to the schema.
5. Specify the user's password, then click *Next*.
6. Select the classes and attributes to deploy into the Identity Vault schema, then click *Next*.



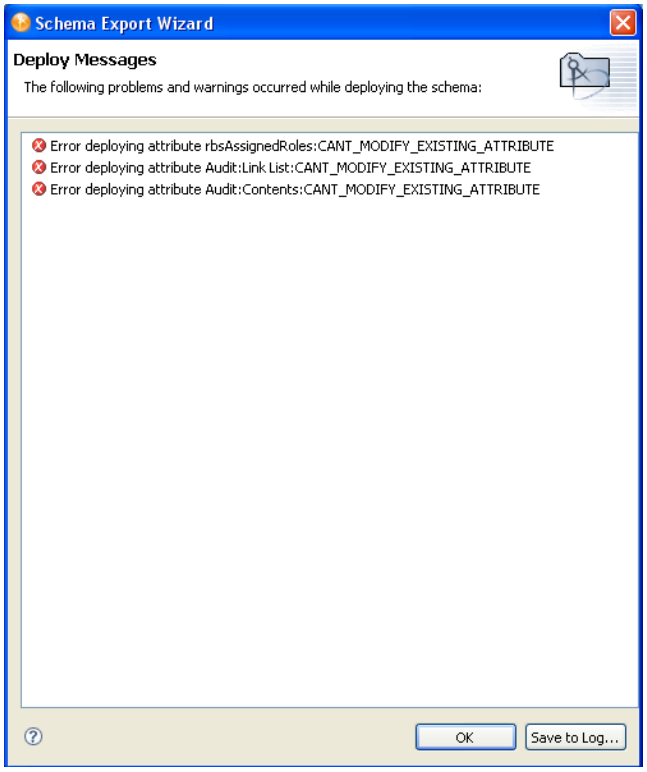
7 Review the summary of classes and attributes to be deployed, then click *Finish*.

If you have selected duplicate attributes or classes, a warning box appears



8 Select *Yes* or *No*, depending upon whether you want to resolve the duplicate classes or attributes.

9 Review errors or warnings, then click *OK*.



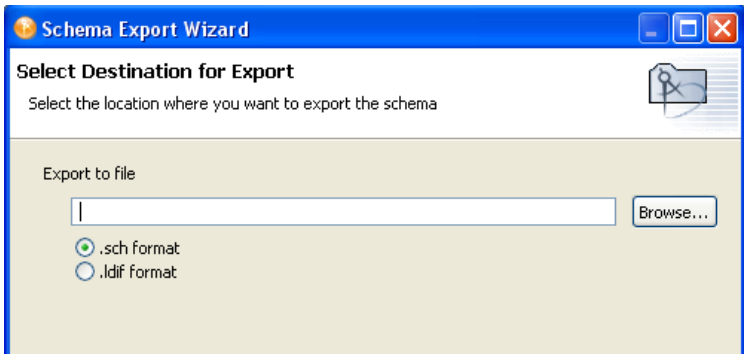
6.5 Exporting the Schema to a File

- Section 6.5.1, “Exporting the Schema to a .sch File,” on page 152
- Section 6.5.2, “Exporting the Schema to an LDIF File,” on page 153

6.5.1 Exporting the Schema to a .sch File

1 In the Modeler, right-click an Identity Vault, then select *Export to File* > *Schema*.

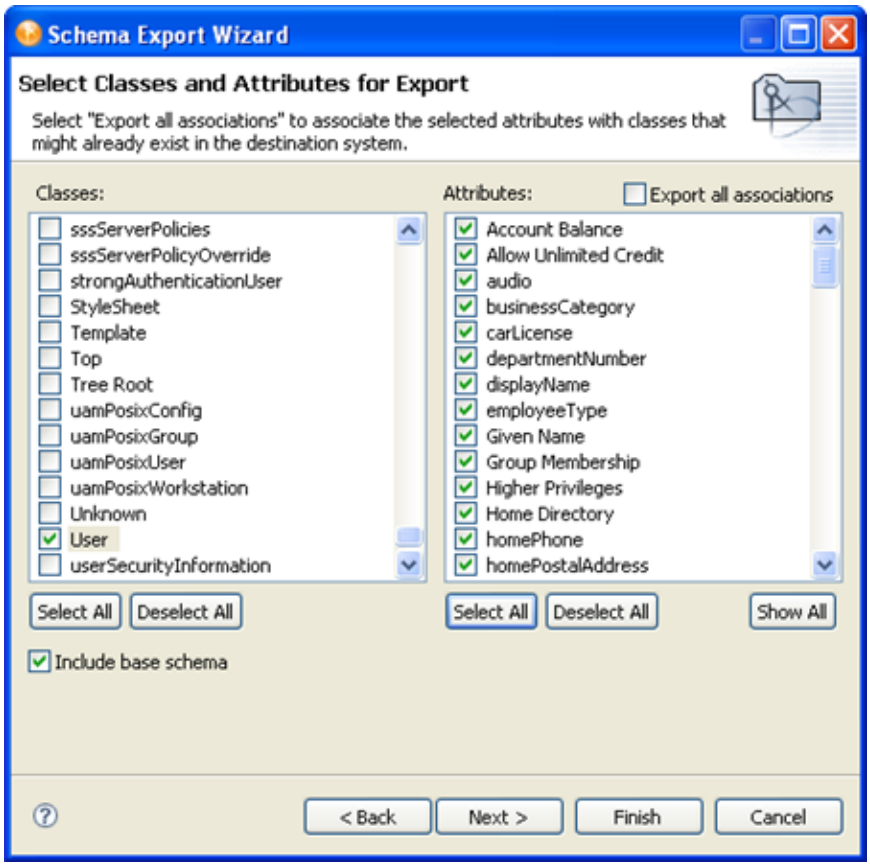
2 In the Schema Export Wizard, select *.sch format*.



3 Specify a filename and location where you want to save the schema file, then click *Next*.

Designer appends the .sch extension when you export the file.

4 In the Select Classes and Attributes for Export page, select the classes and attributes to export to the .sch file, then click *Finish*.



Export all associations (above the *Attributes* pane) enables you to associate the selected attributes with the classes that might already exist in the Identity Vault. If you do not select this box, the new attributes that should be associated with the class are not associated.

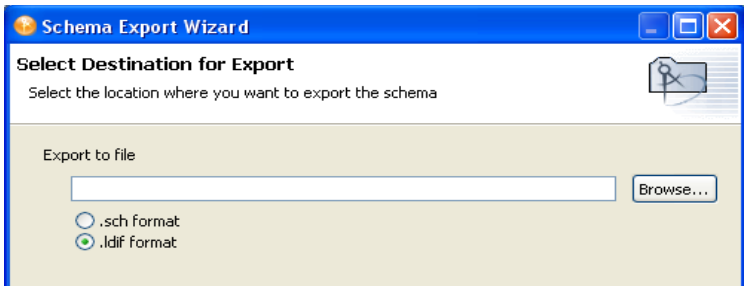
For example, if the Employee Photo attribute is associated with the User class, and *Export all associations* is not selected, Employee Photo is not associated with the User class.

The classes and attributes that are in Designer are listed in the two columns. All classes and attributes are selected by default. To prevent a class or attribute from being deployed, deselect it. To add all classes and attributes, click *Select All*. To remove all classes and attributes, click *Deselect All*.

6.5.2 Exporting the Schema to an LDIF File

1 In the Modeler, right-click the Identity Vault, then select *Export to File > Schema*.

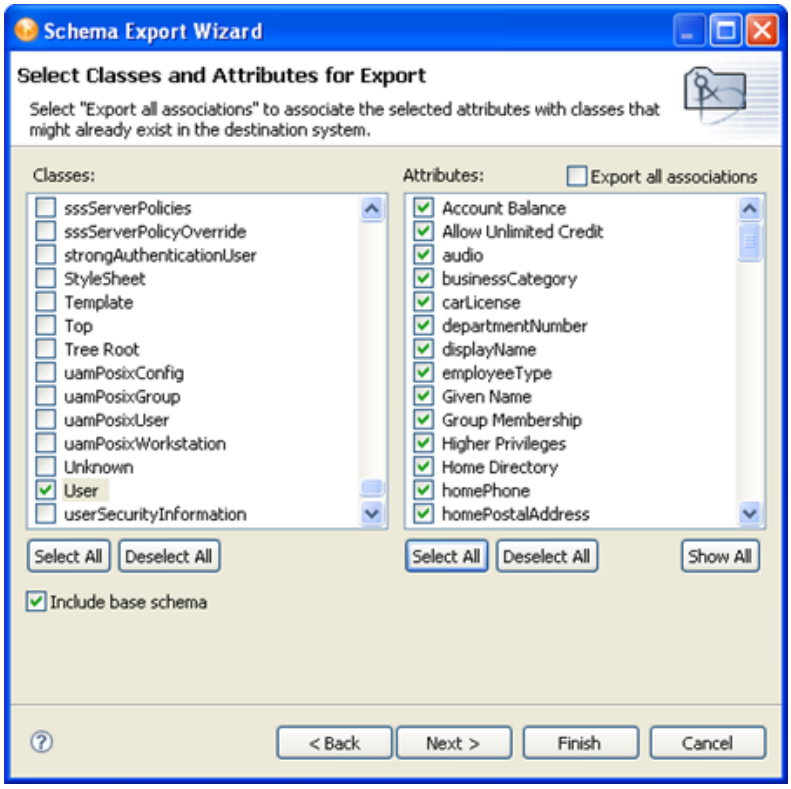
2 In the Schema Export Wizard, select *.ldif format*.



3 Specify a name and location where you want to save the schema file, then click *Next*.

Designer appends the `.ldif` extension when you export the file.

4 In the Select Classes and Attributes for Export page, select the classes and attributes to export to the `.ldif` file, then click *Finish*.



Export all associations (above the *Attributes* pane) enables you to associate the selected attributes with the classes that might already exist in the Identity Vault. If you do not select this box, the new attributes that should be associated with the class are not associated.

For example, if the Employee Photo attribute is associated with the User class, and *Export all associations* is not selected, Employee Photo is not associated with the User class.

5 Click *OK* in the warning.

The class and attribute names in Designer are the Identity Vault (eDirectory) names. The names for the classes and attributes in the LDIF file are the LDAP names. The Identity Vault names differ from the LDAP names. Verify that the names listed in the LDAP file are correct for your

environment before importing the file. For a list of Identity Vault class and attribute names mapped to LDAP class and attribute names, see Section 6.8, “Mapping Identity Vault to an LDAP Schema,” on page 162.

6.6 Importing the Schema

Designer allows you to import the schema from your production environment to do in-depth testing with the Identity Manager drivers.

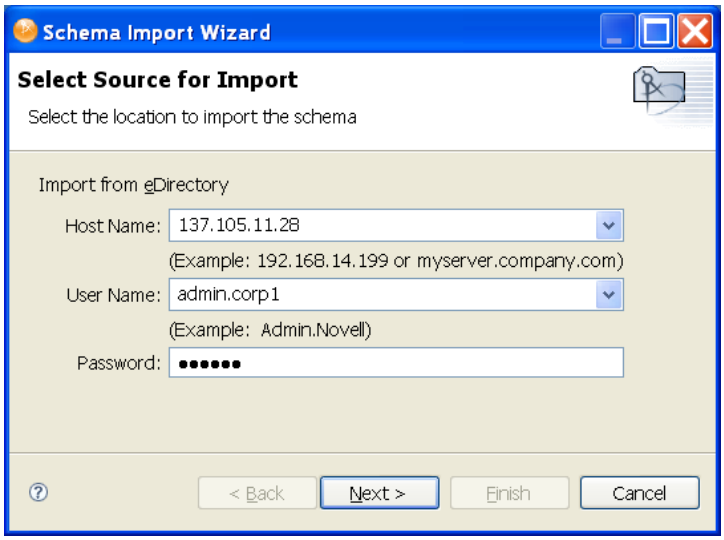
- Section 6.6.1, “Importing the Schema from the Identity Vault,” on page 155
- Section 6.6.2, “Importing the Schema from a File,” on page 157

6.6.1 Importing the Schema from the Identity Vault

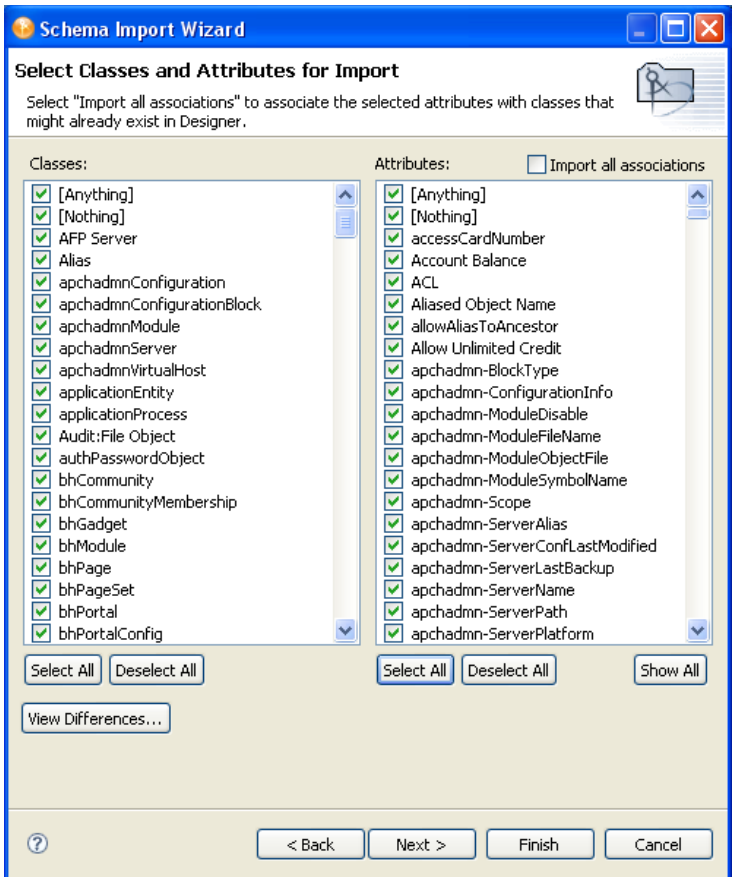
1. In Designer, select an Identity Vault, then select *Live* > *Schema* > *Import*.
2. In the Select Source for Import dialog box, specify the access information to access the server that has the schema to import, then click *Next*.

   Specify the appropriate host name (or IP address), username, and password to access the server.

   NOTE: The specified user must have administrative rights to the schema.



3. In the Select Classes and Attributes for Import page, select the classes and attributes to import into the project, then click *Next*.



Import All Associations: Enables you to associate the selected attributes with the classes that might already exist in Designer. If you do not select this box, the new attributes that should be associated with the class are not associated.

For example, if the attribute of Employee Photo is associated with the User class, and you do not select *Import all associations*, Employee Photo is not associated with the User class.

View Differences: Enables you to view the differences in the schema between the Identity Vault and Designer.

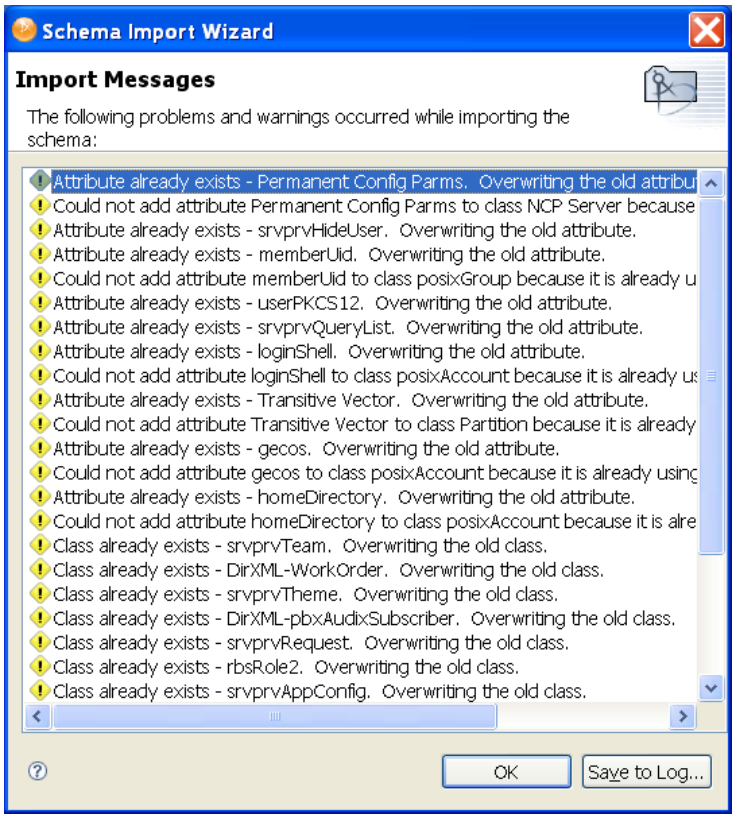
When you click *View Differences*, Designer opens the Schema Differences page, where you can select those differences between the live Identity Vault and the Identity Vault in your project.

You can select schema differences individually, or click *Select All* to import all the schema differences.

Click *OK* to move the selected class and attribute import selections into the Select Classes and Attributes for Import page.

4 Click Next to bring up the Import Summary page, where you can review classes and attributes to import into the project. Then click *Finish*.

If errors occur during the import process, the Import Messages page lists them.



5 On the Import Messages page of the Schema Import Wizard, click *OK*.

If you want to save the differences to a log file, click *Save to Log*. This brings up the Save As dialog box, where you can choose a filename and directory to store the file in.

6 Click *Save*, then click *OK*.

6.6.2 Importing the Schema from a File

When you created an Identity Vault in the Modeler, Designer created a base schema in your project. If a `.sch` file or `.ldif` file has been saved, you can quickly add classes and attributes for your drivers by importing classes and attributes from the saved file.

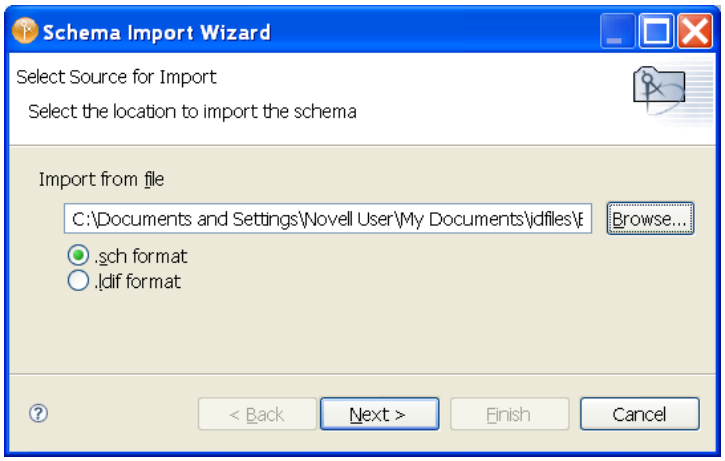
- "Importing the Schema from a .sch File" on page 157
- "Importing the Schema from an LDIF File" on page 159

Importing the Schema from a .sch File

1 In the Modeler, right-click the Identity Vault that will use the imported `.sch` file.

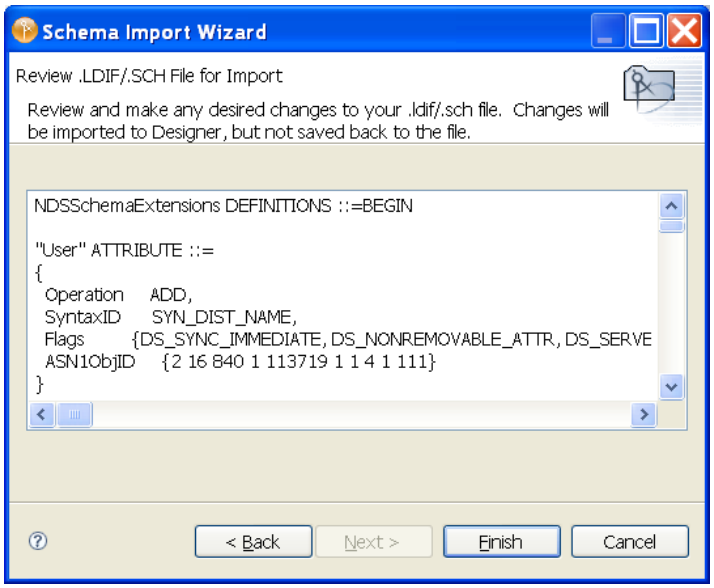
2 Select *Import Schema from File*.

3 Select *.sch format*.



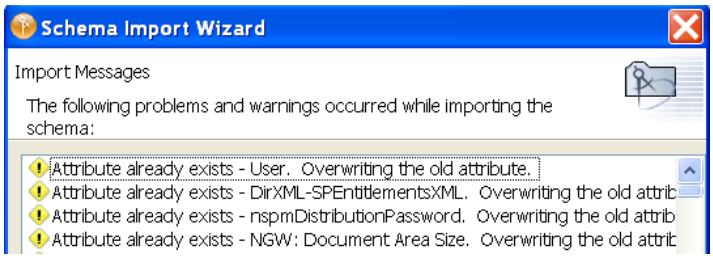
4 Browse to and select the *.sch* file that you want to use, then click *Open*.

5 Click *Next*, then review the .sch file.



6 Make changes if necessary, then click *Finish*.

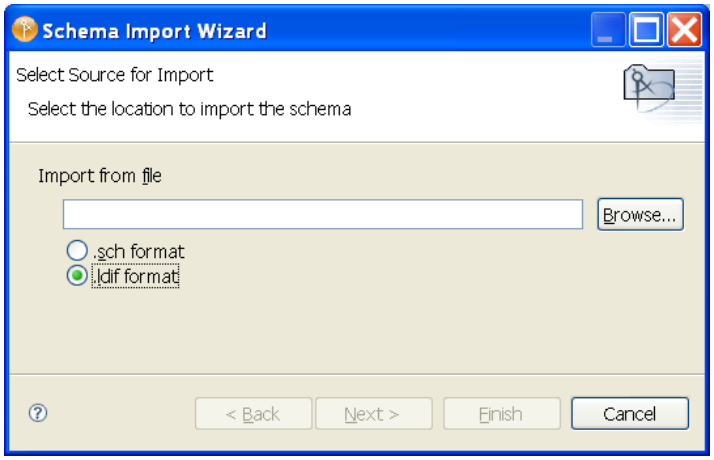
7 Click *OK*.



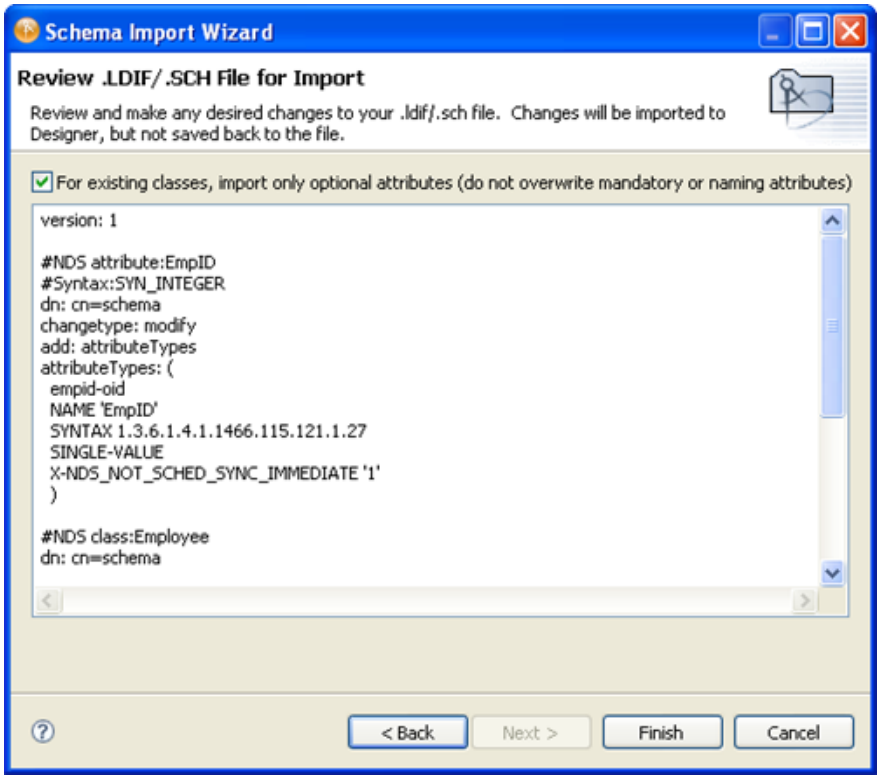
If errors occur, a deployment summary screen lists them.

Importing the Schema from an LDIF File

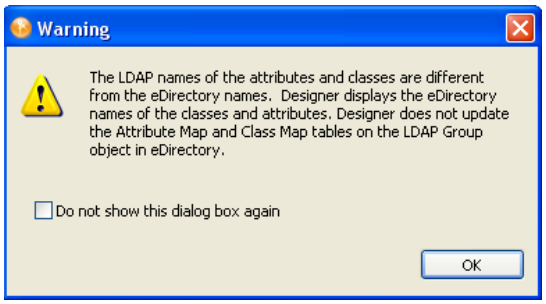
1. In the Modeler, right-click the Identity Vault that will use the imported `.ldif` file.
2. Select *Import Schema from File*.
3. Select *.ldif format*.



4. Specify, or browse to, the `.ldif` file that you want to use, then click *Open*.
5. Click *Next*, then review the `.ldif` file.

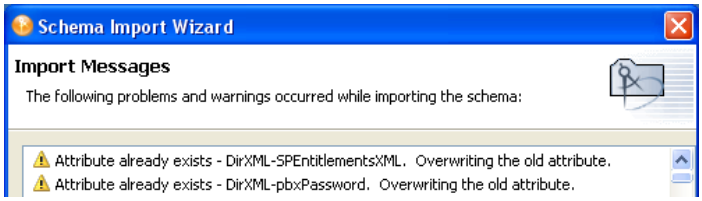


6. Make changes if necessary, then click *Finish*.
7. If you receive a Warning box, read the message and click *OK*.



The class and attribute names in Designer are the Identity Vault (eDirectory) names. The names for the classes and attributes in the LDIF file are the LDAP names. The Identity Vault names differ from the LDAP names. Verify that the names listed in the LDAP file are correct for your environment before importing the file. For a list of Identity Vault class and attribute names mapped to LDAP class and attribute names, see Section 6.8, “Mapping Identity Vault to an LDAP Schema,” on page 162.

8 Click *OK*.



If errors occur, a deployment summary dialog box lists them.

6.7 Managing a Copy of an Application Schema

The Identity Manager engine currently uses the application schema for the following:

- DirXML® Script uses the dn-format/dn-delims to figure out how to parse or convert DNs coming from and going to the application.
- The multi-valued flag on attributes that are used during the attribute merge process that happens as part of a match, resync, or migrate.

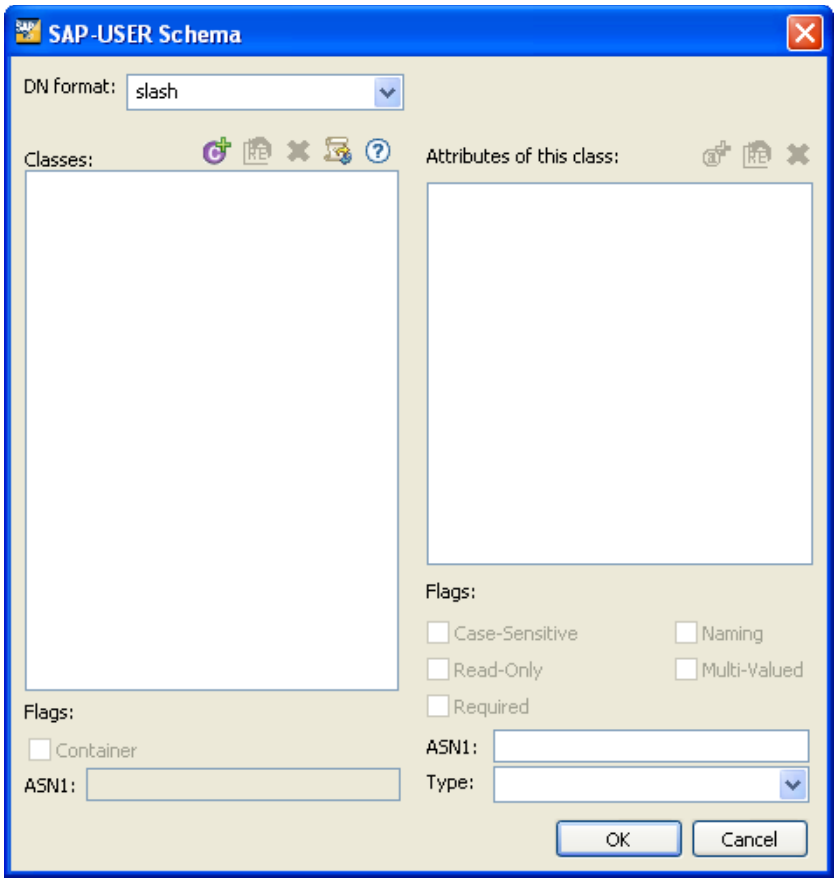
- Section 6.7.1, “Editing an Application’s Schema,” on page 160
- Section 6.7.2, “Refreshing the Application Schema,” on page 162

6.7.1 Editing an Application’s Schema

Designer enables you to manage a copy of the connected system’s schema. You can make changes to a copy of the application schema so that you can test the Identity Manager drivers in Designer. The schema changes cannot be deployed into the live application schema.

1 Right-click the driver connection in Designer, then select *Manage Application Schema*.

2 Add, rename, or delete the application’s classes or attributes, then click *OK*.



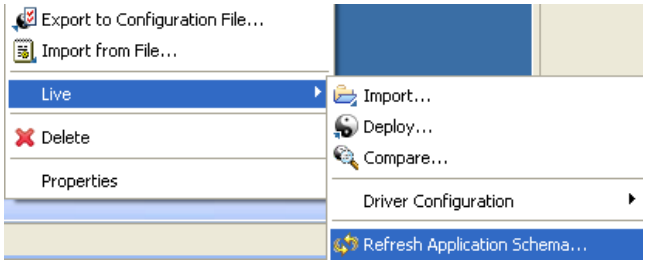
Field	Description
DN format	Specifies the separator character used when specifying distinguished names. For example, admin.utah.novell.com.
Classes	Lists all of the classes stored in Designer from the application's schema.
Add a class	Adds a new class.
Rename class	Renames the selected class.
Delete class	Deletes the selected class.
Refresh application schema	Provides a new copy of the application's schema.This option is useful if the application schema changes.
Help	Launches the Help documentation for the Manage Schema tool.
Flags: Container	Specifies whether the class is a container.
ASN1	The unique ID of the class.
Attributes of This Class	Lists all of the attributes stored in Designer for the selected class from the application's schema.

Field	Description
Add an attribute	Adds a new attribute for the selected class.
Rename attribute	Renames the selected attribute.
Delete attribute	Deletes the selected attribute.
Flags	Specifies the details of the attribute. To edit the flags, select an attribute.
Type	Specifies the syntax of the attribute. To view the syntax, select an attribute. To change the syntax, select an option from the drop-down list.

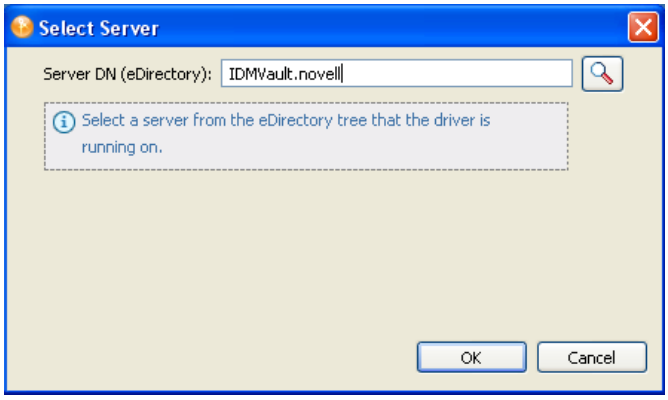
6.7.2 Refreshing the Application Schema

If the application schema changes, you can get a new copy of the application's schema by refreshing the application schema.

1 Right-click the driver connection, then select *Live* > *Refresh Application Schema*.



2 Click the browse icon.



3 Browse to and select the server where the driver is installed, then click *OK* twice.

6.8 Mapping Identity Vault to an LDAP Schema

When accessing the Identity Vault through LDAP, the names of classes and attributes might be different than when accessed through the standard NCP-based APIs.

For more information about how that mapping is performed, see

- “Class and Attribute Mappings” (http://www.novell.com/documentation/edir88/index.html?page=/documentation/edir88/edir88/data/h0000007.html#a5bwxyz)

NDK: *Novell eDirectory Schema Reference* (http://devsup.novell.de/ndk/doc/ndslib/index.html?page=/ndk/doc/ndslib/schm_enu/data/h4q1mn1i.html) at the Novell Developer Support Web page

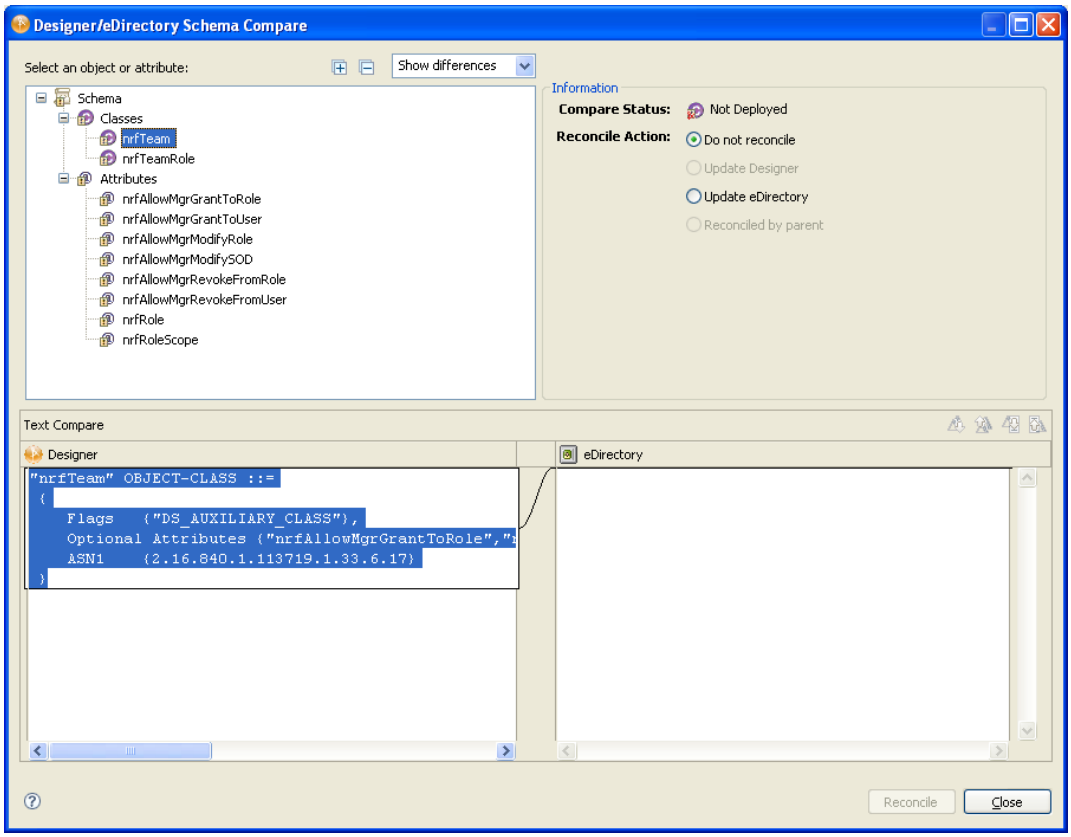
6.9 Comparing the Schema

Designer allows you to compare schema from your production environment to do in-depth testing with the Identity Manager drivers. Designer now provides conflict resolution on individual classes and attributes and allows you to view the differences between existing and new values when importing and deploying the schema. For example, before deploying schema to an Identity Vault you can run Schema Compare.

Schema Compare shows the differences between the schema of the ID Vault and that of the Designer. The viewer shows 'Schema' at the root level with 'Classes' and 'Attributes' under it. Classes and Attributes can be expanded to list various classes and attributes respectively. When an item is selected, the information section shows it as equal (no action is necessary) or unequal. If they are unequal, you can choose not to reconcile them, choose to update them in Designer, or choose to update them in eDirectory. To reconcile the differences between the schema of the Designer and that of the eDirectory while in Compare, you don’t need to run the Import or Deploy of Schema.

Do the following to determine if you have deployed all the changes you made in the Designer schema to the Identity Vault schema.

1. Right-click the ID Vault in either the Modeler view or in the Outline view. Select *Live* > *Schema* > *Compare* to bring up the Designer/eDirectory Schema Compare window.

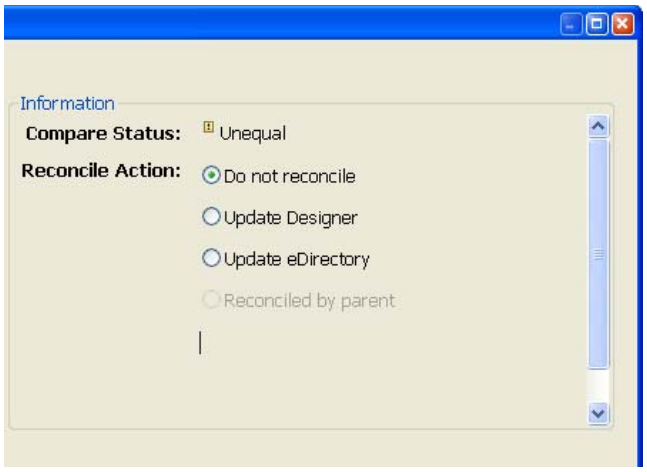


2 Expand the *Select a class or attribute* portion of the window to see the listed classes or attributes respectively. Select an individual class or an attribute to see the actual differences displayed in the *Text Compare* portion of the window.

   The plus icon at the right side of the *Select a class or attribute* allows you to expand all elements in the parent object, and the minus icon collapses all of the elements. The ? icon displays the Summary/Compare dialog box help.

3 By default, the Compare window only displays values that are different between eDirectory and Designer. To view all the classes and attributes, select *Show all* from the pull-down menu. Your choices are *Show differences*, *Show deletes*, and *Show all*.

4 Check to see the status of the values that are shown. Values that are equal are shown as *Equal* on the *Compare Status* line in the Information portion of the Compare window.



The overlay image displayed in the *Compare Status* entry identifies objects or attributes that need reconciliation. The following table describes what you see in the *Compare Status* line and the overlays that you can see:

Compare Status	Description
Equal	The selected classes or attributes are same in eDirectory and Designer.
Unequal	The value of the selected class or an attribute, or one or more classes or attributes, are different in eDirectory and Designer.
Not Deployed	The selected class or an attribute is not deployed to eDirectory.
Not Imported	The selected class or an attribute does not exist in Designer.

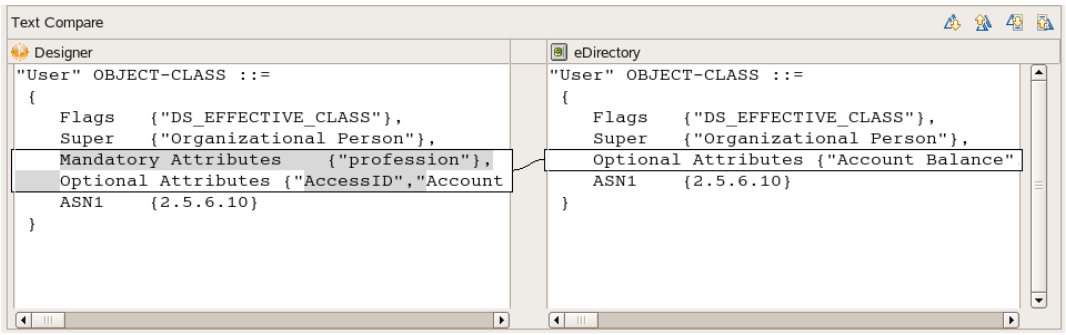
5 Under the Information portion of the Compare window, select how you want to reconcile the differences between the Source and Destination.

If *Compare Status* shows *Unequal*, you have three choices:

- To do nothing, keep the default value of *Do Not Reconcile*.
- To update the schema in Designer so that it contains the same information as the schema in eDirectory, select *Update Designer*.
- To update the schema in eDirectory to reflect the changes you have just made to the schema in Designer, select *Update eDirectory*.

6 Check to see the *Text Compare* values.

The *Text Compare* values displayed in the bottom portion of the Designer/eDirectory Schema Compare window shows the difference at the child object level. The Text Compare dialog box uses the Eclipse* Compare editor to compare classes and attributes. The differences in the code are highlighted in blue.



7 After you view the differences, click *Reconcile* to perform the reconciliation actions for each object in the tree, or click *Close* to close the Designer/eDirectory Object Schema Compare window.

Managing the Flow of Data

7

Designer allows you to manage how the data flows between the Identity Vault and the connected systems. You can see how the data flows between all of the connected systems, make changes as needed, create reports about the data, and view the flow of passwords between the systems.

The Dataflow view and the Dataflow editor manage the data. The Dataflow view displays the flow of data in the modeler per driver. The Dataflow editor displays a more granular view.

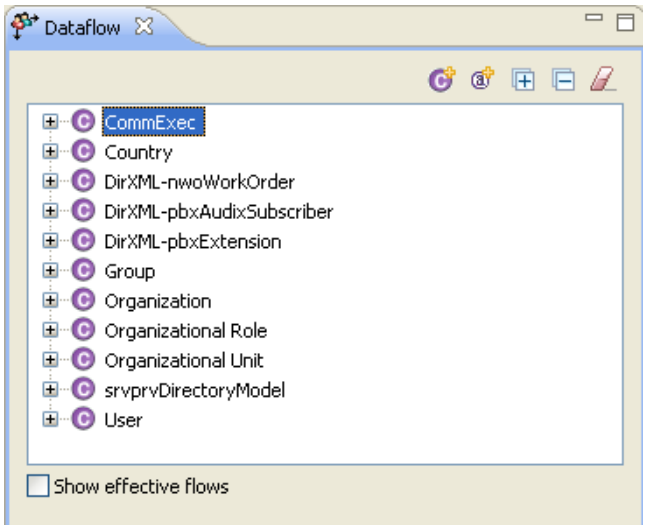
- Section 7.1, “The Dataflow View,” on page 167
- Section 7.2, “The Dataflow Editor,” on page 173
- Section 7.3, “Adding Items in the Dataflow Editor,” on page 181
- Section 7.4, “Editing Items,” on page 185
- Section 7.5, “Generating HTML Reports,” on page 189
- Section 7.6, “Integrating Passwords,” on page 191

7.1 The Dataflow View

The Dataflow view displays a toolbar in the upper right corner of the view. For information on the icons in this toolbar, see The Dataflow View (http://www.novell.com/documentation/designer35/designer_intro/data/projdataflow.html) in the *Understanding Designer for Identity Manager*.

The following figure illustrates the Dataflow view. You can use it to control the flow of data between the Identity Vault and connected systems. The Modeler displays the dataflow.

Figure 7-1 *The Dataflow View*



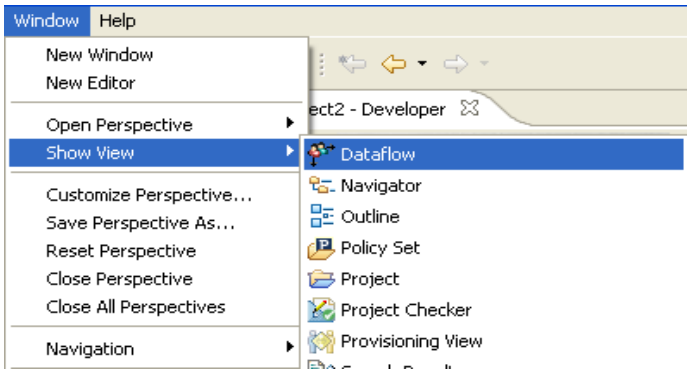
- Section 7.1.1, “Accessing the Dataflow View,” on page 168
- Section 7.1.2, “Flow Arrows in the Modeler,” on page 168

- Section 7.1.3, “Viewing How Attributes Are Synchronized,” on page 170
- Section 7.1.4, “Changing the Data Flow,” on page 170

7.1.1 Accessing the Dataflow View

If you have closed the Dataflow view, you can access it by selecting *Window* > *Show View* > *Dataflow*.

Figure 7-2 *Selecting the Dataflow View*



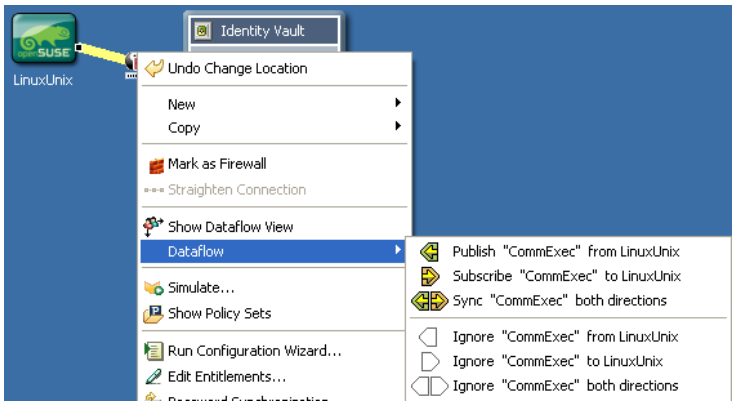
If the Dataflow view is blank and no project is displayed in the Modeler:

1. Expand a project in the Project view.
2. Open the project by double-clicking *System Model*.

   Objects and icons appear in the Dataflow view.

If you want to change how the data flows from the Modeler:

1. Right-click a driver or application in the Modeler.

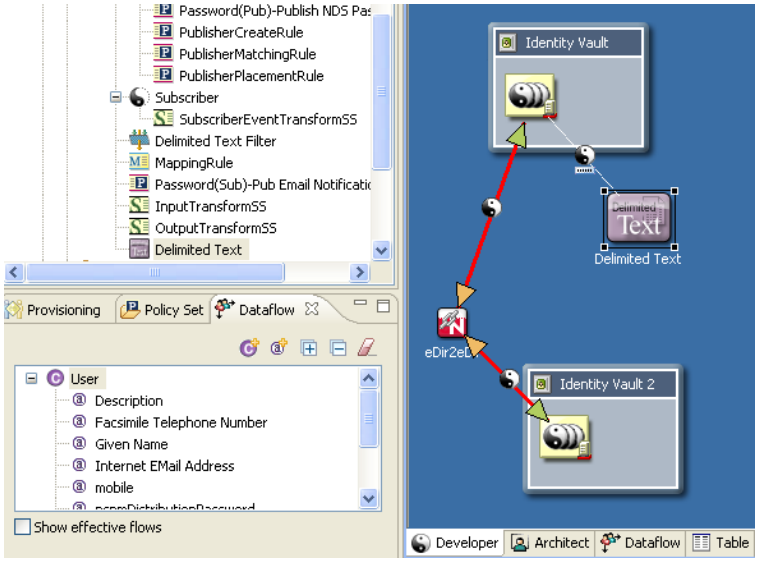


2. Select *Dataflow*, then select how you want the data flow to change.

7.1.2 Flow Arrows in the Modeler

When the Dataflow view opens, it automatically reads the filters and shows the classes and attributes. (If a filter with classes and attributes doesn’t exist, you can create one.)

Figure 7-3 *Flow Arrows in the Modeler*



As you select a class (for example,) or attribute in the Dataflow list, the appropriate driver lines are highlighted in red in the Modeler. Icons enable you to see Sync, Notify, Reset, and Ignore filter settings all at the same time.

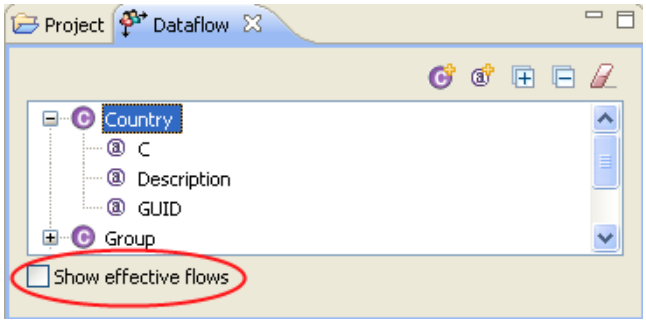
Table 7-1 *Dataflow Icons*

Icon	Description
	Green arrow: the Publisher channel is synchronized.
	Orange arrow: the Subscriber channel is synchronized.
	Bell: the attribute is set to Notify.
	Reset arrow: the attribute is set to Reset.
No decoration	The attribute is set to Ignore.

The color coding matches the Dataflow icons in the Filter editor and the Dataflow editor.

7.1.3 Viewing How Attributes Are Synchronized

Figure 7-4 *Show Effective Flows*

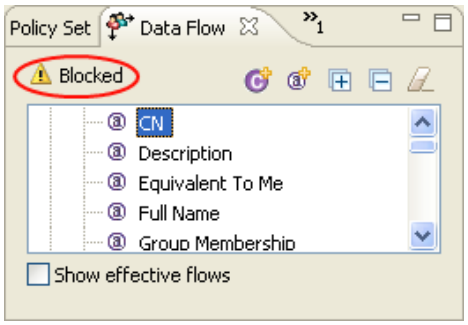


To view whether attributes are synchronized or whether they will be notified, select *Show effective flows*. When you select this check box, the synchronize arrows don't show if the parent class isn't set to synchronize. Therefore, you view an accurate diagram of actual flows.

However, if you want to view how attributes are configured to synchronize, regardless of the parent class, deselect *Show effective flows*. The synchronize arrows indicate which items are synchronized.

If you select an attribute that can't synchronize (whether or not *Show effective flows* is selected), you see a *Blocked* warning in the upper left. This warning indicates that this attribute can't be synchronized or notified because the parent class isn't synchronized.

Figure 7-5 *The Blocked Text and Icon*



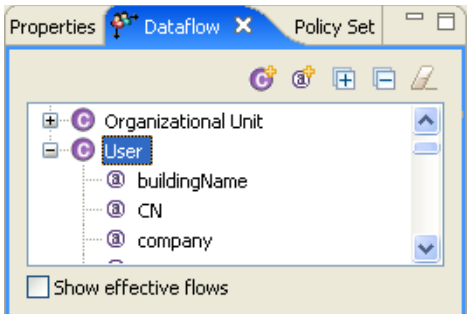
To view an explanation, mouse over the *Warning* icon.

7.1.4 Changing the Data Flow

You can change how the data flows for classes and attributes from the Dataflow view.

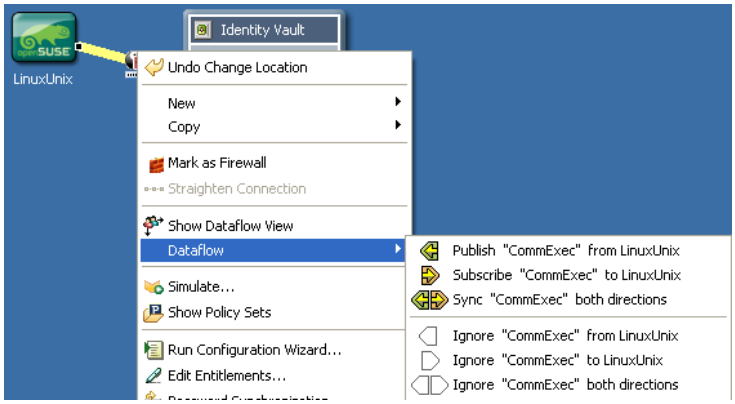
To change the flow for a class:

1. Select a class in the Dataflow view.



2 Right-click a driver line in the Modeler.

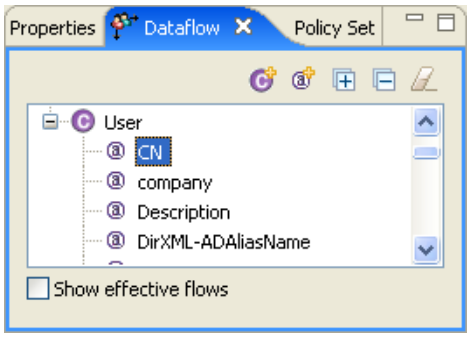
3 Select *Dataflow.*



4 Select the option to change the data flow for the class.

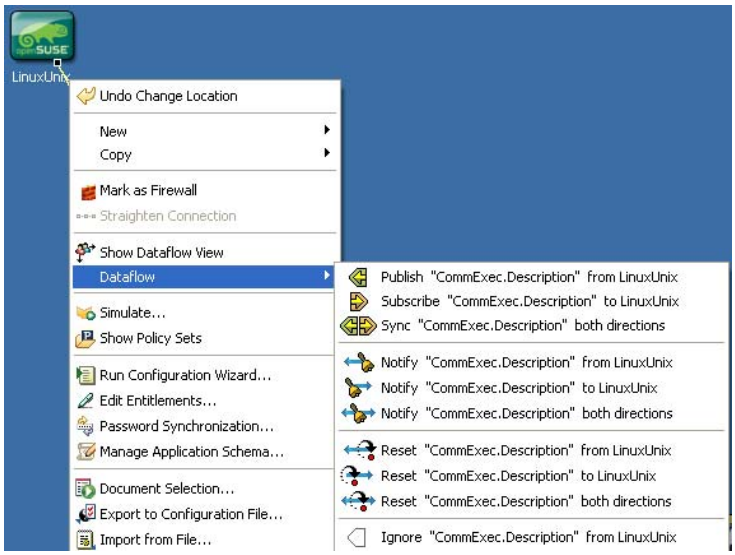
To change the flow for an attribute:

1 Select an attribute in the Dataflow view.



2 Right-click a driver line in the Modeler.

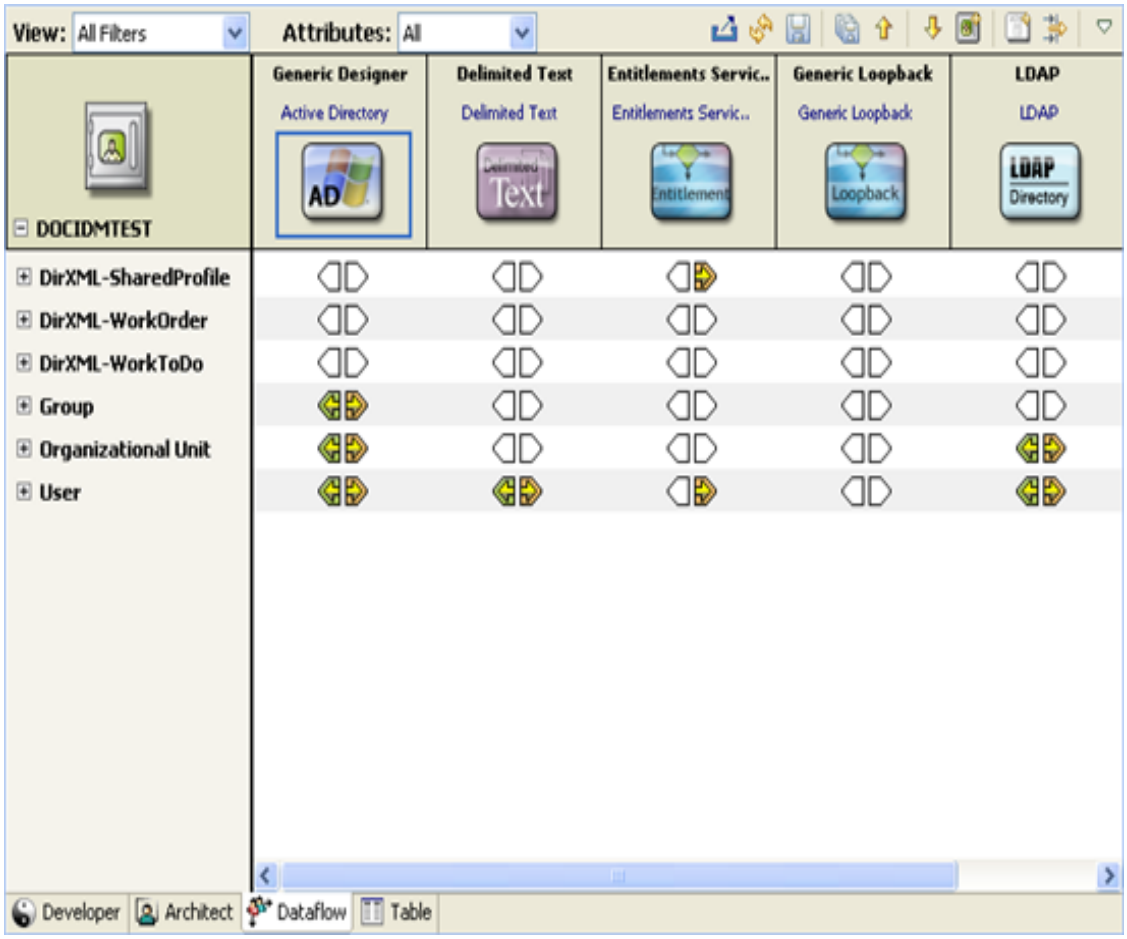
3 Select *Dataflow*.



4 Select the option to change the data flow for the attribute.

7.2 The Dataflow Editor

Figure 7-6 *The Dataflow Editor*



The Dataflow editor enables you to do the following:

- Use filters to display how data flows between all systems and Identity Vaults.
- View how passwords flow from each server.
- Generate reports of the data.

When object additions, deletions, changes, and selections synchronization occur, the Dataflow editor synchronizes with the Modeler and the Outline view.

To access the Dataflow editor, click the Dataflow tab.

To adjust the area for the Identity Vaults, move the slider bar. This setting persists and is restored the next time you run the editor.

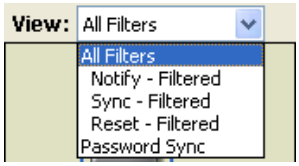
- Section 7.2.1, “Filtering Views,” on page 174
- Section 7.2.2, “Filtering Identity Vaults and Applications,” on page 176
- Section 7.2.3, “Pinning the Identity Vault,” on page 176
- Section 7.2.4, “Expanding and Collapsing the Identity Vault,” on page 177

- Section 7.2.5, “Switching to an eDirectory Tree Icon,” on page 179
- Section 7.2.6, “Viewing an eDir-to-eDir Driver,” on page 180
- Section 7.2.7, “Keyboard Support,” on page 180

7.2.1 Filtering Views

By default, the Dataflow editor shows all dataflows. The View drop-down list (in the upper left corner of the Dataflow editor, not in the Dataflow view), enables you to view notification, synchronization, reset, or Password Sync information. These filtered views do not allow you as much editing capability as the main view, but just what is necessary in that filter. For example, you can’t add attributes, vaults, or applications, because by default they wouldn’t appear in the filter.

Figure 7-7 *Options to Filter Views in the Dataflow View*

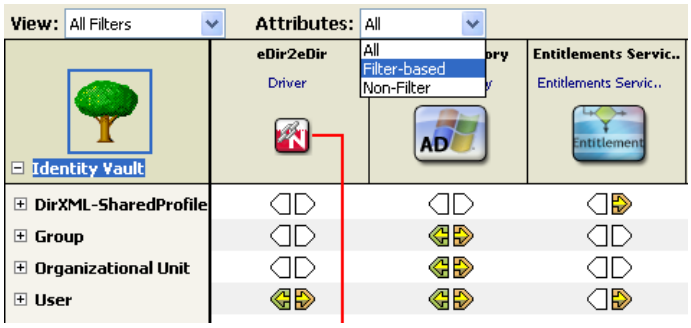


- “Using the All Filters View” on page 174
- “Synchronizing Passwords” on page 174

Using the All Filters View

If you are in the All Filters view, you can further filter with the *Attributes* list. Because the Dataflow editor provides non-filter attributes, you can choose to view regular filter-based attributes, non-filter attributes, or both.

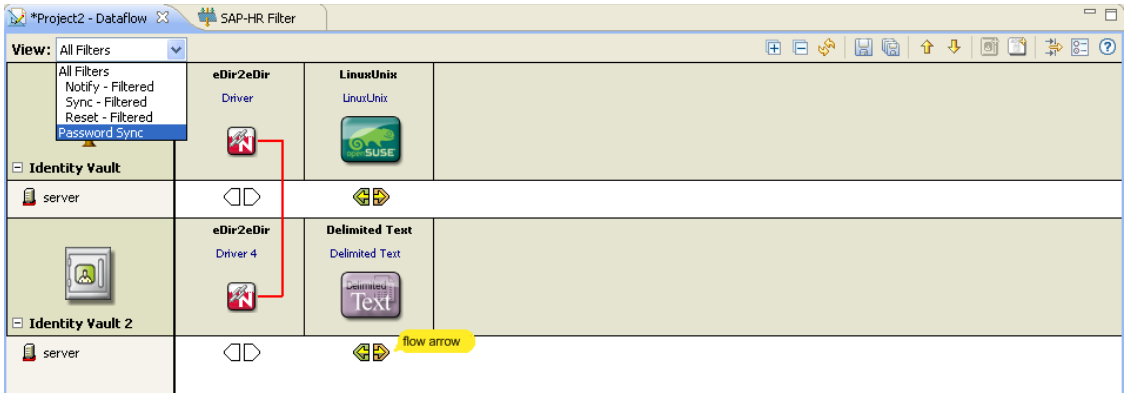
Figure 7-8 *Options in the All Filters View*



Synchronizing Passwords

The Password Sync view enables you to see and edit how all passwords flow in the project. Designer displays the information on a per-server basis and shows how passwords flow among all of the applications.

Figure 7-9 *The Password Flow*

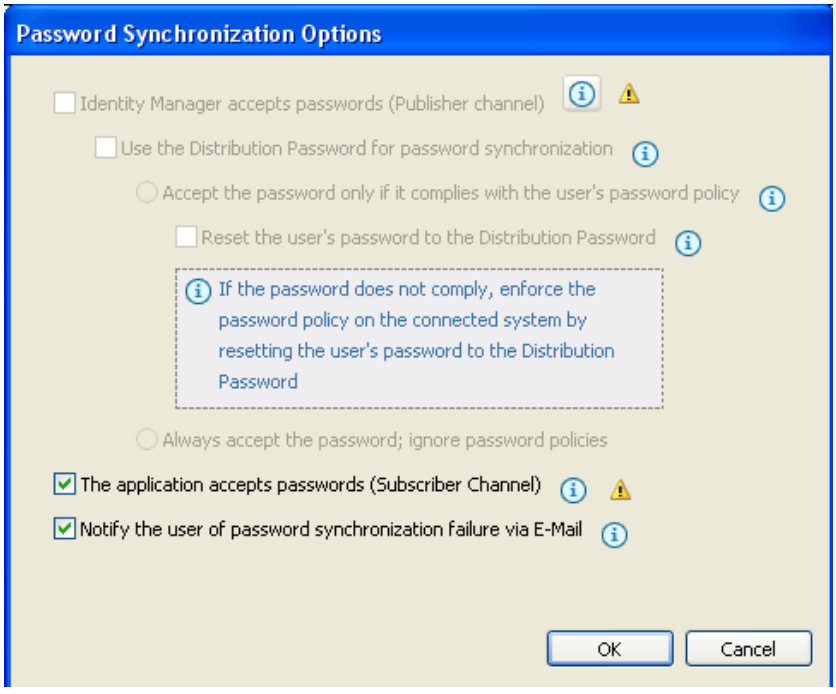


To edit the password flow:

1. Select *Password Sync* in the View filter.
2. Double-click the flow arrow.

   You can also right-click, then select *Password Synchronization*.
3. Edit the password synchronization options.

   For more information about password synchronization, see “Password Synchronization across Connected Systems” in the *Novell Identity Manager Administration Guide* (http://www.novell.com/documentation/idm35/index.html).

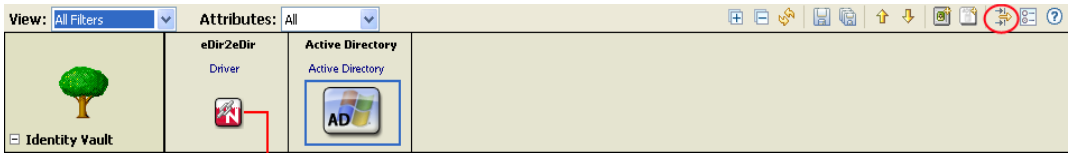


4. Click *OK*.

7.2.2 Filtering Identity Vaults and Applications

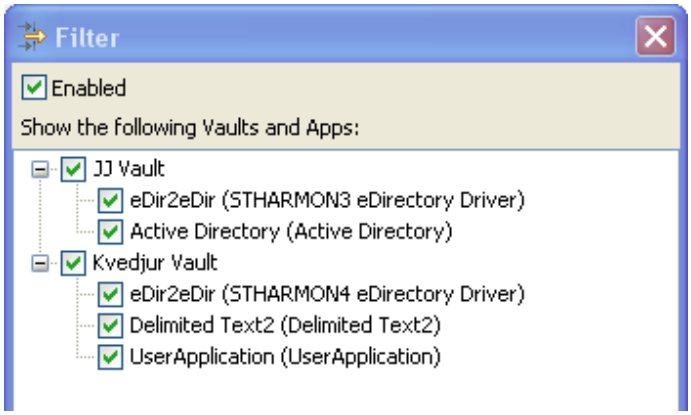
You can select the Identity Vaults and applications that you want to view in the editor.

1 In the Dataflow editor, click the *Filter View* icon.



2 Select *Enabled*.

The Identity Vaults and applications that you select here are included in the HTML reports. For more information, see Section 7.5, “Generating HTML Reports,” on page 189.

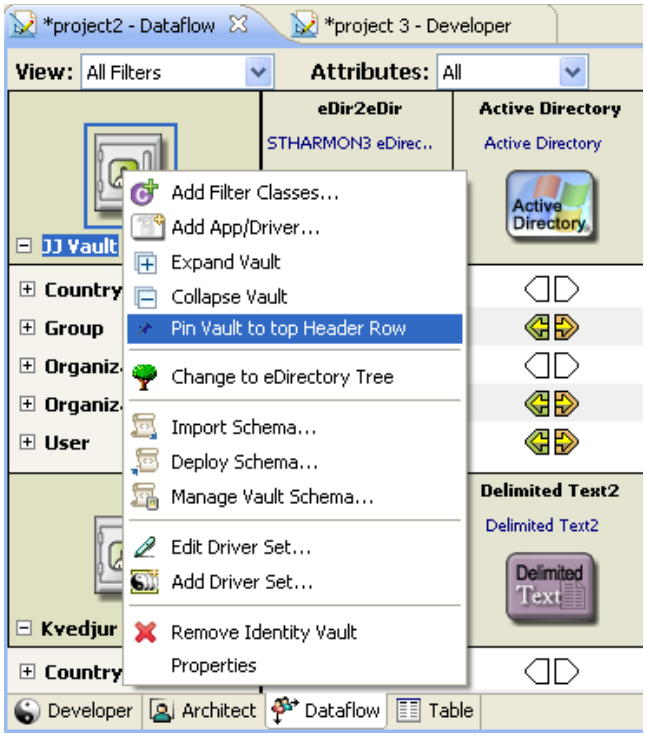


You can scroll and resize the dialog box. Also, you can interact with the Dataflow editor in the background, in any mode. This is convenient if you want to scroll a different section into view while this dialog box is up.

7.2.3 Pinning the Identity Vault

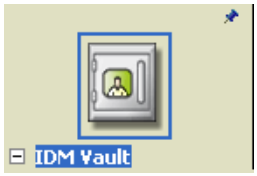
To change the scope of the editor to show a single Identity Vault, right-click the vault, then select *Pin Vault to Top Header Row*.

Figure 7-10 *Pinning an Identity Vault*



With a medium or large-sized project, the dataflow table can contain hundreds of rows and thousands of items. If you have multiple vaults and want to narrow the scope to more easily edit a vault without excessive scrolling, you might want to pin a vault. When an Identity Vault is pinned, a pin icon displays in the upper right corner.

Figure 7-11 *A Pinned Identity Vault*



To unpin the vault, right-click the Identity Vault, then select *Unpin Vault from Top Header Row*.

7.2.4 Expanding and Collapsing the Identity Vault

- “Expanding an Identity Vault” on page 177
- “Expanding All Identity Vaults” on page 178
- “Expanding Classes” on page 179

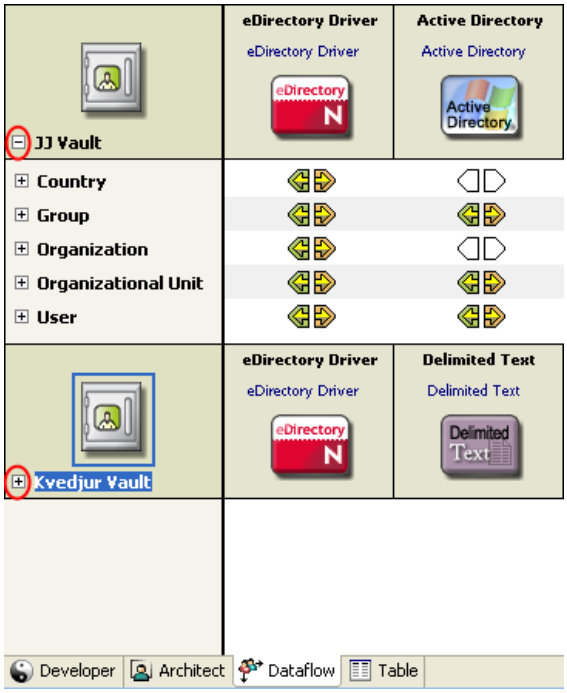
Expanding an Identity Vault

When the editor first loads, all vaults are expanded at the top level by default.

To collapse or expand the list of classes and attributes in an Identity Vault, do one of the following:

- Click the - or + icon below the Identity Vault icon.

Figure 7-12 *Icons to Expand or Collapse the List of Classes*

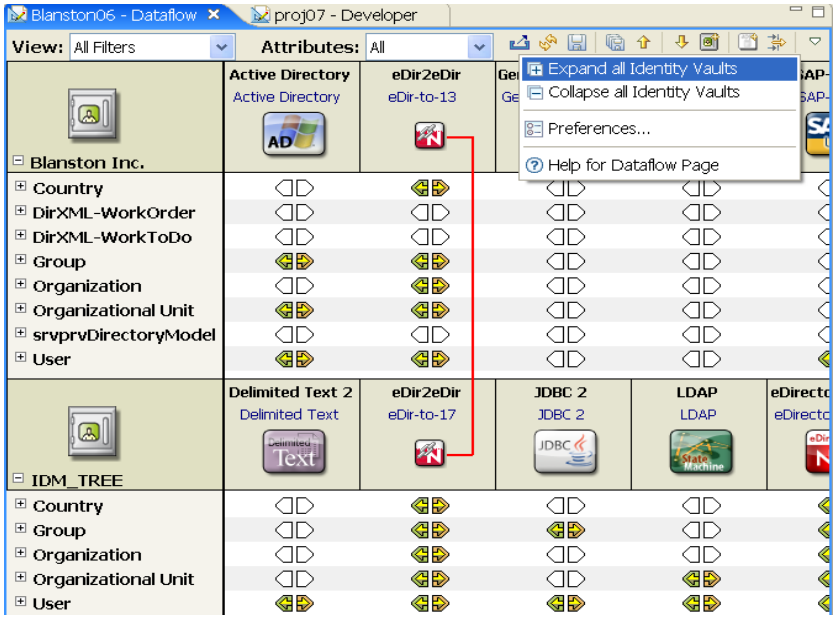


- Select the Identity Vault, then press the Right-arrow key to expand the information, or press the Left-arrow key to collapse the information.

Expanding All Identity Vaults

To expand or collapse the list of classes and attributes for all Identity Vaults, click the *Expand all Identity Vaults* or *Collapse all Identity Vaults* from the pull-down menu.on the toolbar.

Figure 7-13 *Select to Expand or Collapse All Identity Vaults*

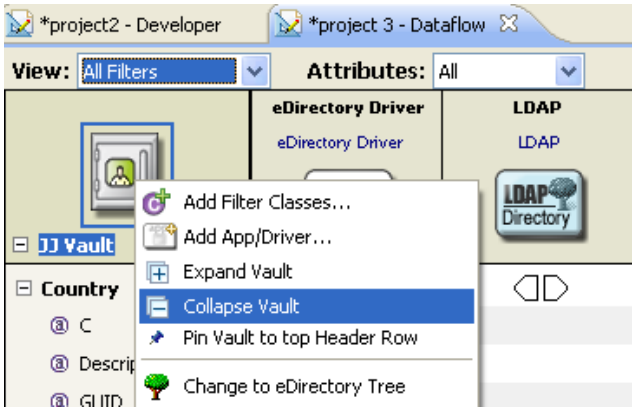


Expanding Classes

To view all attributes in a class, select the class, then press the Right-arrow key. To collapse the list of attributes, press the Left-arrow key.

To view all classes and attributes in an Identity Vault, right-click the Identity Vault icon, then select *Expand Vault*. To list just classes in an Identity Vault, right-click the Identity Vault, then select *Collapse Vault*.

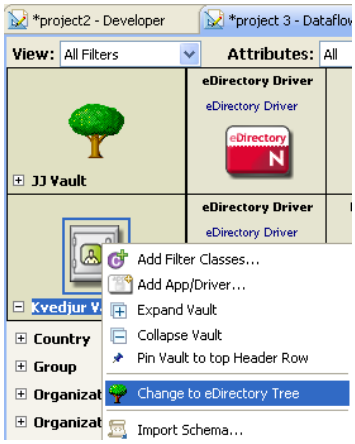
Figure 7-14 *Menu Options to Expand an Identity Vault*



7.2.5 Switching to an eDirectory Tree Icon

To switch from an Identity Vault icon to an eDirectory tree icon, right-click the Identity Vault, then select *Change to eDirectory Tree*.

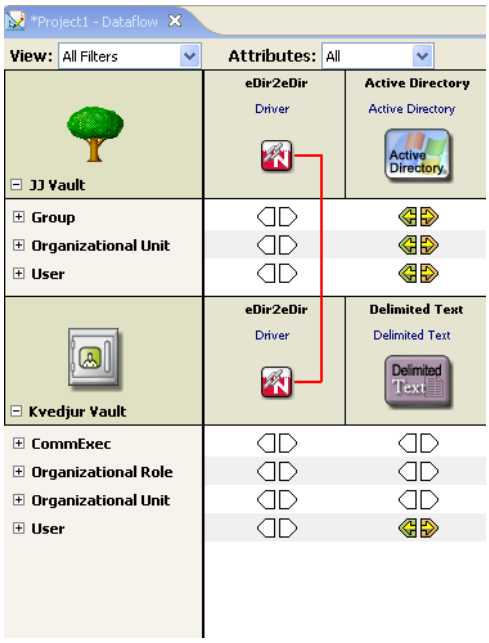
Figure 7-15 *Changing to an eDirectory Tree*



7.2.6 Viewing an eDir-to-eDir Driver

You can easily view both ends of an eDir-to-eDir connection so that you can configure the dataflows on both sides. Designer automatically detects the two eDirectory applications and aligns them in the same table column. A red line connects them.

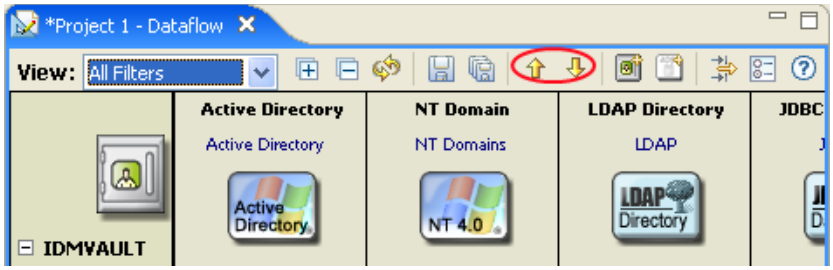
Figure 7-16 *An eDir-to-eDir Connection*



7.2.7 Keyboard Support

You can navigate by using the Up-arrow, Down-arrow, Left-arrow, and Right- arrow keys as well as PageUp, PageDown, Home, and End. In addition, you can navigate from one Identity Vault to another by clicking the up-arrow or down-arrow on the toolbar.

Figure 7-17 *Navigation Icons*

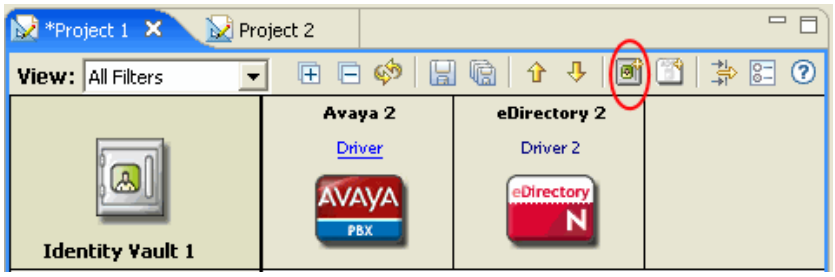


7.3 Adding Items in the Dataflow Editor

- Section 7.3.1, “Adding an Identity Vault in the Dataflow Editor,” on page 181
- Section 7.3.2, “Adding a Driver in the Dataflow Editor,” on page 182
- Section 7.3.3, “Adding an Application in the Dataflow Editor,” on page 182
- Section 7.3.4, “Adding Classes and Attributes,” on page 183
- Section 7.3.5, “Adding Non-Filter Attributes,” on page 183

7.3.1 Adding an Identity Vault in the Dataflow Editor

Figure 7-18 *The Dataflow Editor's Toolbar*



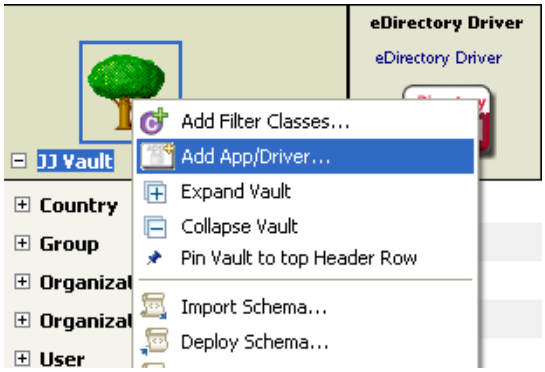
To add an Identity Vault, click the *Add Identity Vault* icon on the toolbar.

To configure the Identity Vault, double-click it.

To delete an Identity Vault, select it, then press Delete.

7.3.2 Adding a Driver in the Dataflow Editor

Figure 7-19 *An Identity Vault in the Dataflow Editor*

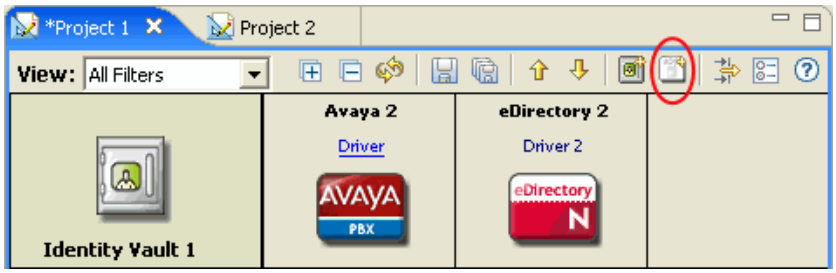


To add a driver while in the Dataflow editor, right-click an Identity Vault, then select *Add App/Driver*.

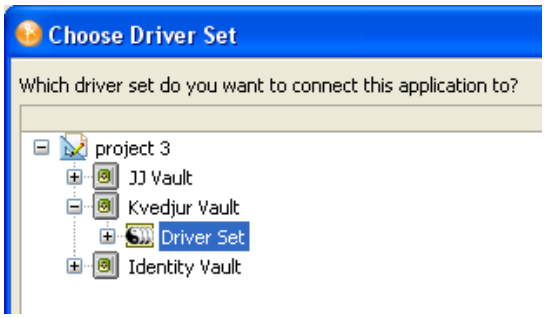
To delete an Identity Vault or driver, select it, then press the Delete key.

7.3.3 Adding an Application in the Dataflow Editor

1 On the toolbar, click the *Add Application* icon.



2 Browse to and select the driver set that you want this application to connect to, then click *OK*.



3 Select the driver you want to create, then click *OK*.

Designer creates a skeleton of the driver. It does not launch the Driver Configuration Wizard. If you want to configure the driver, right-click the connection icon in the Modeler, then select *Run Configuration Wizard*.

7.3.4 Adding Classes and Attributes

You can add classes and attributes to the dataflow.

To add a class:

1. Right-click an Identity Vault, then select *Add Classes*.
2. Select the class that you want to add, then click *OK*.

   If you want to add more than one class, press Ctrl and select the classes.

To add an attribute:

1. Right-click a class, then select *Add Attributes*.
2. Select the attribute that you want to add, then click *OK*.

   If you want to add more than one attribute, press Ctrl and select the attributes.

7.3.5 Adding Non-Filter Attributes

The Dataflow editor provides non-filter attributes. By default, all classes and attributes in the Dataflow editor come directly from all of the filter policies of the drivers. However, in production environments, it is common to cause data to flow a certain way directly in your Policy Script code, XSLT, or in external code that you call out to.

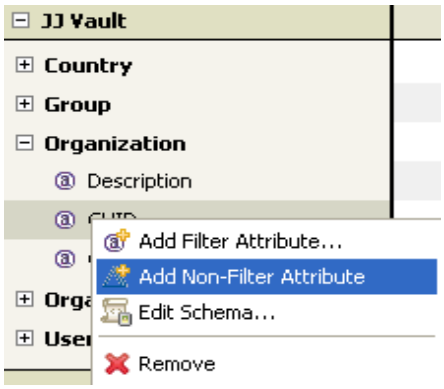
Usually, these non-filter attributes aren't defined in a policy filter (unless you're describing "augmented" processing) and aren't in the schema map. This is because they are generated outside of normal driver operations and you need them in the schema mapping rule only if the engine processes them.

Normally, non-filter attributes are operated on in the Publisher Command Transformation policy set or the Subscriber Output Transformation policy set.

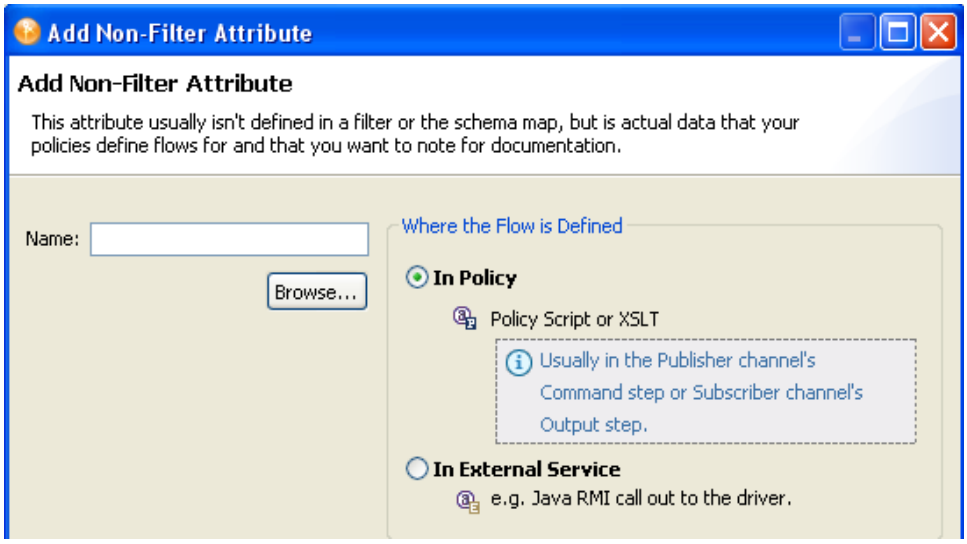
The Dataflow editor lets you add the non-filter attributes to the table for documentation purposes so that you can capture the attributes and have an accurate picture of your actual enterprise dataflows.

To add a non-filter attribute:

1 Right-click the class or attribute name, then select *Add Non-Filter Attribute*.



2 Specify the name of the attribute or class, or click *Browse*, then browse to and select the attribute or class.



3 Click *OK*.

4 Select where the flow of the attribute or class is defined.

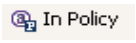
In Policy: The dataflow is defined in a policy script or an XSLT style sheet.

In External Service: The dataflow is defined in a Java RMI call to the driver.

5 Click *OK*.

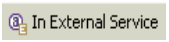
If the non-filter attribute is defined by a policy, a small P is added to the icon. This icon distinguishes a non-filter attribute from a regular filter attribute.

Figure 7-20 *A Non-Filter Attribute*



If the attribute is defined by an external service, a small E is added to the icon.

Figure 7-21 *A Non-Filter External Attribute*



7.4 Editing Items

- Section 7.4.1, “Editing within the Dataflow Editor,” on page 185
- Section 7.4.2, “Editing Non-Filter Attributes,” on page 187
- Section 7.4.3, “Managing Schema,” on page 187
- Section 7.4.4, “Removing a Flow,” on page 187
- Section 7.4.5, “Changing How Data Flows,” on page 188

7.4.1 Editing within the Dataflow Editor

As a convenience, you can edit many items within the Dataflow editor. This capability turns the Dataflow editor into a full project editor that allows you to have all the tools you need in one place. You can edit Identity Vault properties, classes, attributes, drivers, and applications.

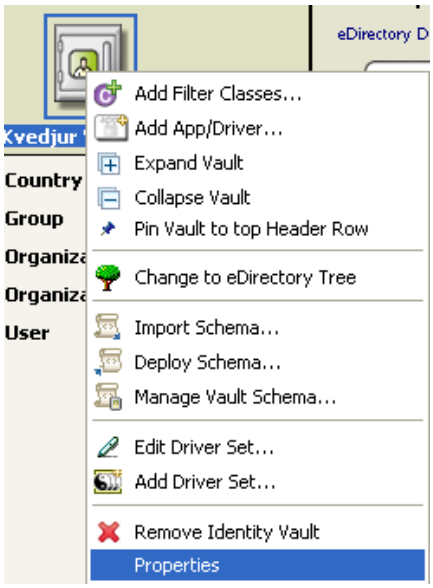
- “Identity Vault Properties” on page 185
- “Classes and Attributes” on page 186
- “Drivers” on page 186
- “Applications” on page 187

Identity Vault Properties

Access the Identity Vault’s properties pages by doing one of the following:

- Double-click the Identity Vault.
- Select the Identity Vault, then press Enter.
- Right-click the Identity Vault, then select *Properties*.

Figure 7-22 *The Properties Option*



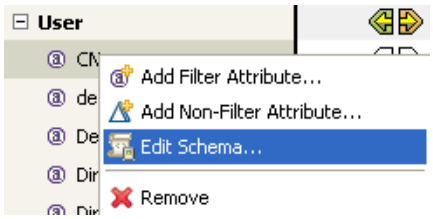
Classes and Attributes

Launch the Manage Schema tool by doing one of the following:

- Double-click the class or attribute.
- Select the class or attribute, then press Enter.
- Right-click the class or attribute, then select *Edit Schema*.

This tool enables you to modify classes and attributes. For more information, see Chapter 6, “Managing the Schema,” on page 137.

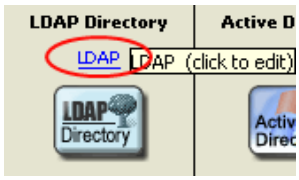
Figure 7-23 *The Edit Schema Option*



Drivers

To access the driver’s property pages, click the driver name below the application name.

Figure 7-24 *Location of a Driver Name*

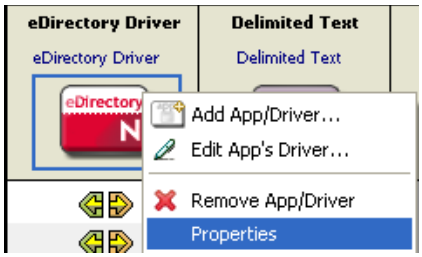


Applications

Access the properties pages for the application by doing one of the following:

- Double-click the application.
- Select the application, then press Enter.
- Right-click the application, then select *Properties*.

Figure 7-25 *The Properties Option for an Application*



7.4.2 Editing Non-Filter Attributes

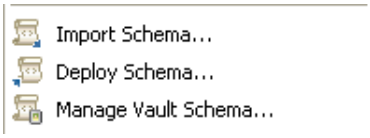
The directional flow of these attributes is edited in the same way as other attributes. Right-click the arrows and select *Publish*, *Subscribe*, *Ignore*, *Reset*, or *Remove from Filter*.

Reset means that you have the value reset under certain conditions. The attribute might be in a policy filter, but in addition, you might have some manual logic that resets the value. Occasionally, resets by manual logic occur in production environments.

7.4.3 Managing Schema

To import, deploy, and edit the schema in the Dataflow editor, right-click an Identity Vault. All schema changes made outside of this editor are synchronized. For more information, see Chapter 6, “Managing the Schema,” on page 137.

Figure 7-26 *The Manage Schema Option*

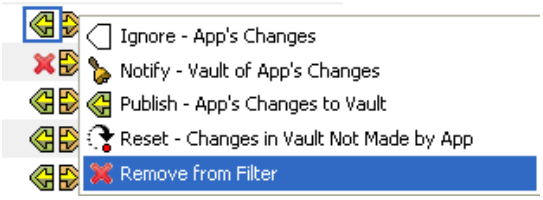


7.4.4 Removing a Flow

If a particular flow (Publisher or Subscriber channel) is not defined in the policy filter’s XML, a red X replaces the Publisher or Subscriber channel arrow. This means that it’s not in the policy and there will be no flow. This scenario is essentially the same as an *Ignore Flow* icon, which is an empty white arrow. However, the distinction is useful so that you know what is actually in your policy’s XML.

To remove the flow from the XML:

1. Right-click the Publisher or Subscriber channel icon.
2. Select *Remove from Filter*.



If a class or attribute is marked to be removed on both channels and nothing references it, Designer removes it from the Dataflow editor's table.

7.4.5 Changing How Data Flows

To change the way data flows, right-click the arrow that displays the dataflow, then select the option that you want.

When you right-click the arrow that displays the dataflow for an attribute, you are presented with five options, as shown below:

- Ignore
- Notify
- Subscribe/Publish
- Reset
- Remove from Filter

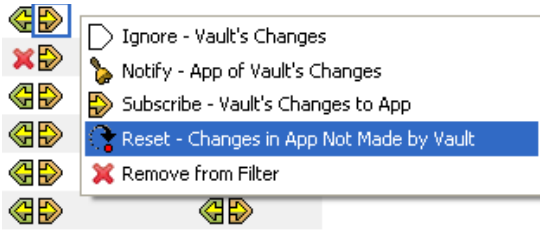
The functionality for these options changes depending on whether you have selected the left (Publisher) channel or the right (Subscriber channel).

For the Publisher Channel:

- *Ignore - App's Changes* instructs the Identity Vault to ignore changes made in the application.
- *Notify - Vault of App's Changes* notifies the Identity Vault about changes made in the application.
- *Publish - App's Changes to Vault* transfers the changes made to the application into the Identity Vault.
- *Reset - Changes in Vault Not Made by App* resets the changes in the Identity Vault that were not made by the application.
- *Remove from Filter* removes the flow from the XML.

For the Subscriber Channel:

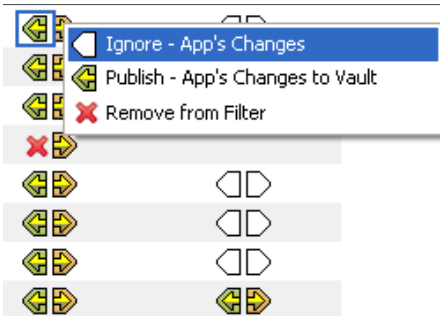
Figure 7-27 *Subscriber Channel Options*



- *Ignore - Vault's Changes* instructs the application to ignore changes made in the Identity Vault.
- *Notify - App of Vault's Changes* notifies the application about changes made in the Identity Vault.
- *Subscribe - Vault's Changes to App* transfers the changes made to the Identity Vault into the application.
- *Reset - Changes in the App Not Made by Vault* resets the changes in the application that were not made by the Identity Vault.
- *Remove from Filter* removes the flow from the XML.

When you right-click the arrow that displays the dataflow for a class, you are presented with three options, as shown below:

Figure 7-28 *Changing the Publisher Flow*



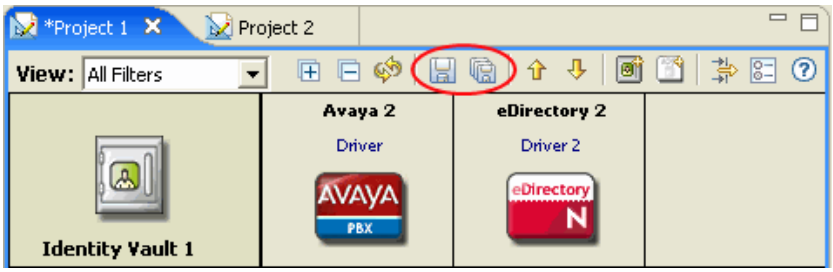
- *Ignore*
- *Publish/Subscribe*
- *Remove from Filter*

The *Reset* and *Notify* options are only available when you select an application.

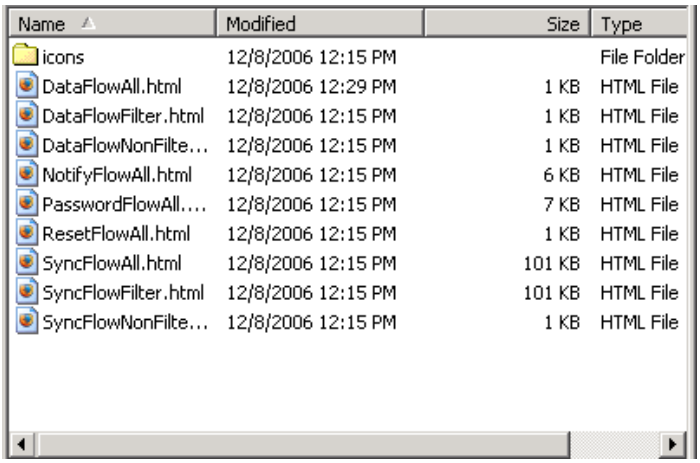
7.5 Generating HTML Reports

Designer allows you to generate HTML reports about your project.

1. Click the *Save Current View to HTML* icon or the *Save All Views to HTML* icon.



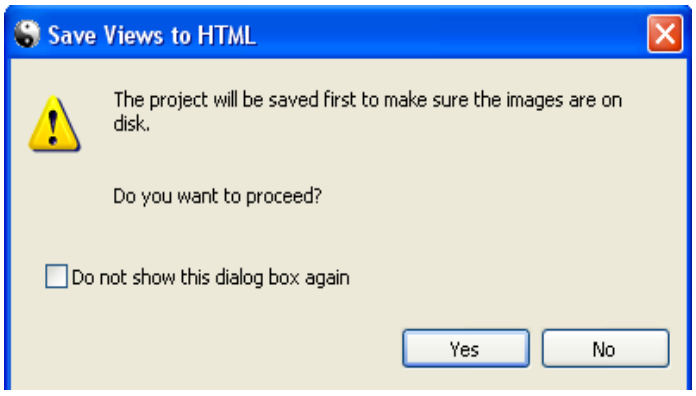
Save Current View to HTML generates a report for the current view. *Save All Views to HTML* generates nine reports.



Name	Modified	Size	Type
icons	12/8/2006 12:15 PM		File Folder
DataFlowAll.html	12/8/2006 12:29 PM	1 KB	HTML File
DataFlowFilter.html	12/8/2006 12:15 PM	1 KB	HTML File
DataFlowNonFilte...	12/8/2006 12:15 PM	1 KB	HTML File
NotifyFlowAll.html	12/8/2006 12:15 PM	6 KB	HTML File
PasswordFlowAll....	12/8/2006 12:15 PM	7 KB	HTML File
ResetFlowAll.html	12/8/2006 12:15 PM	1 KB	HTML File
SyncFlowAll.html	12/8/2006 12:15 PM	101 KB	HTML File
SyncFlowFilter.html	12/8/2006 12:15 PM	101 KB	HTML File
SyncFlowNonFilte...	12/8/2006 12:15 PM	1 KB	HTML File

The HTML files are automatically named. The descriptive names tell what the report is. All images that you need to support the HTML document are copied to an `icons` subdirectory where the HTML is saved. The process includes all of your custom application icons.

You are prompted to save the project to disk.



Saving is necessary to make sure that all of your icon information is in a state where it can be successfully copied.

2 Click *Yes* to save the project.

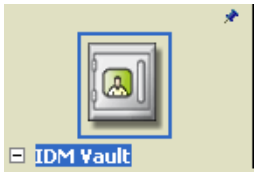
3 Browse to and select the location where you want to save the reports, then click *OK*.

The directory you select for saving is stored in Designer's memory and becomes the default directory the next time you save.

4 Click *OK* in the Information dialog box that indicates where the report is saved.

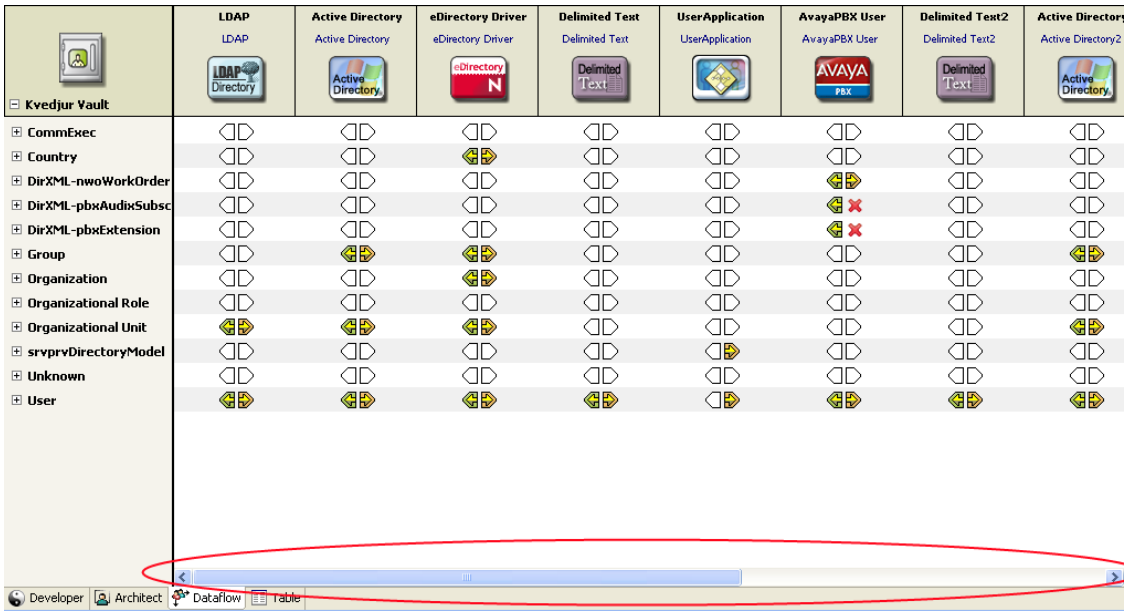
If you pin an Identity Vault and then generate a report, the report is for that Identity Vault. The Identity Vault's name is included in the HTML name.

Figure 7-29 *A Pinned Identity Vault*



If the Dataflow editor has multiple applications, Designer provides a scrollbar to scroll through all the applications within the Dataflow editor.

Figure 7-30 *Continuation Rows in a Report*



7.6 Integrating Passwords

If a driver is synchronizing passwords (in at least one direction), a small password-field icon appears under the driver icon. This icon enables you to know where passwords are being synchronized.

To toggle this icon on or off:

1 Select *Window >Preferences > Designer for IDM > Modeler*.

2 Click the *Display* tab.

3 Select or deselect *Show password icons in Developer mode*.

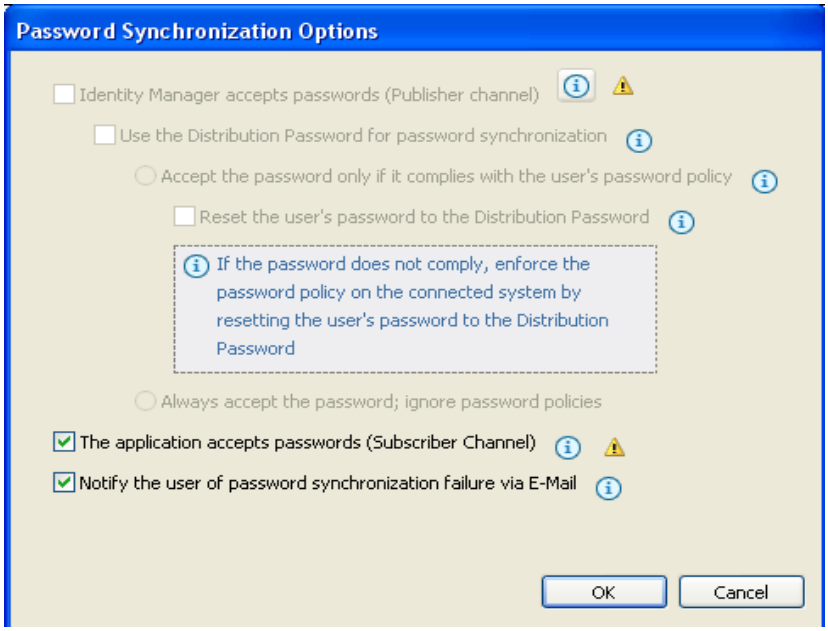
If you mouse over the password icon in Developer mode, a helpful tip explains how your passwords are flowing for each server involved in the flow.

To configure the flow of password synchronization:

1 In Dataflow mode, select *Password Sync* in the *View* drop-down box.

2 Double-click the flow arrow.

3 Select options, then click *OK*.



Creating and Managing Policies

8

Policies enable you to customize the flow of information into and out of Novell eDirectory for a particular environment.

For example, one company might use the inetorgperson as the main user class, and another company might use User. To handle this, a policy is created that tells the Metadirectory engine what a user is called in each system. Whenever operations affecting users are passed between connected systems, Identity Manager applies the policy that makes this change.

Policies also create new objects, update attribute values, make schema transformations, define matching criteria, maintain Novell Identity Manager associations, and many other things.

For more information on policies, refer to the following:

- Understanding Designer for Identity Manager (http://www.novell.com/documentation/idm36/policy/index.html?page=/documentation/idm36/policy/data/bookinfo.html)
- Policies in Designer (http://www.novell.com/documentation/idm36/policy_designer/data/bookinfo.html)
- Novell Credential Provisioning for Identity Manager 3.6 (http://www.novell.com/documentation/idm36/policy_credprov/data/bookinfo.html)
- Identity Manager 3.6 DTD Reference (http://www.novell.com/documentation/idm36/policy_dtd/data/bookinfo.html)

Enabling Staging of Projects

9

Using Identity Manager can involve two or more stages, such as development, the test environment, and the production environment. In an Identity Manager environment, the drivers are the most critical components. When an Identity Manager project is moved from one stage to another, moving drivers becomes critical. Staging provides users the flexibility to validate the appropriate resources in real time, which ensures uniformity across all stages.

Figure 9-1 *Staging Identitiy Manager Projects*

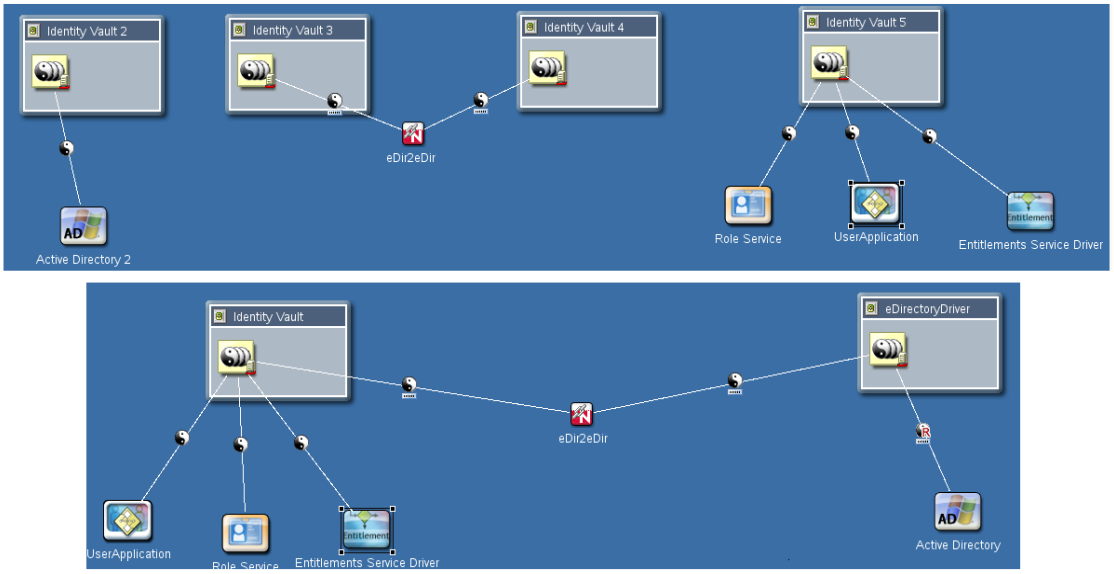


Figure 9-1 shows a basic representation of the movement of Identity Manager projects across different stages. Three projects are developed and tested in three separate setups and then connected in the subsequent stages. The scale of projects grows with stages, but not necessarily the scale of an individual driver.

Staging Identity Manager projects is discussed in the following sections:

- Section 9.1, “Prerequisites,” on page 196
- Section 9.2, “Importing Additional Objects from eDirectory,” on page 196
- Section 9.3, “Deploying Additional Objects into eDirectory,” on page 201
- Section 9.4, “Exporting LDIF Container Objects to an LDIF File,” on page 205
- Section 9.5, “Importing Objects from an LDIF File into an LDIF Container,” on page 207
- Section 9.6, “Setting up Driver Security Equivalences,” on page 208
- Section 9.7, “Editing the LDIF Container Data by Using an Editor,” on page 213
- Section 9.8, “Deleting the LDIF Container,” on page 213
- Section 9.9, “Changing the LDAP Properties,” on page 214

9.1 Prerequisites

Ensure that the following prerequisites are met before attempting staging Identity Manager projects:

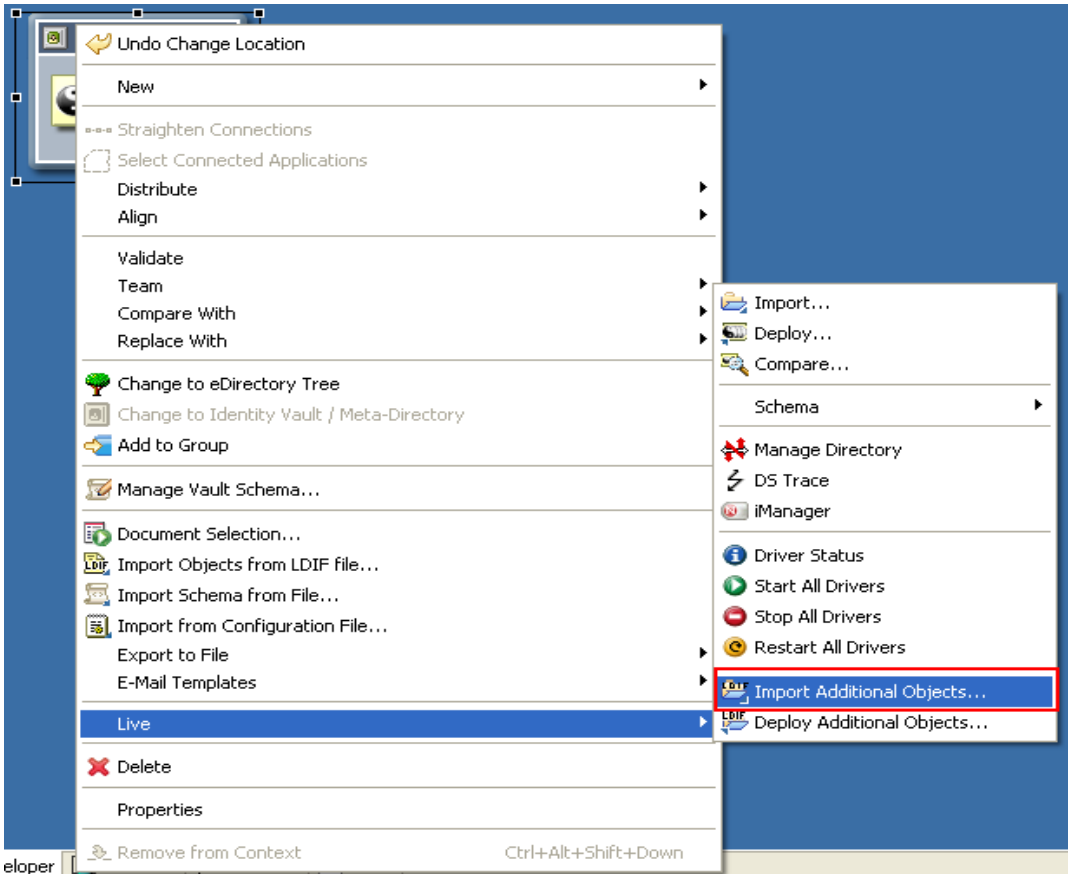
- All the stages should have the same version of eDirectory, Identity Manager, and Identity Manager drivers.
- Designer 3.5 or later is present.
- All the applications and drivers are fully developed and tested in one stage before moving them to the next stage.

  Refer to the *Identity Manager 3.6.1 Staging Best Practices Guide* (http://www.novell.com/documentation/idm36/idm_staging/data/front.html) for more information on pre-staging activities.

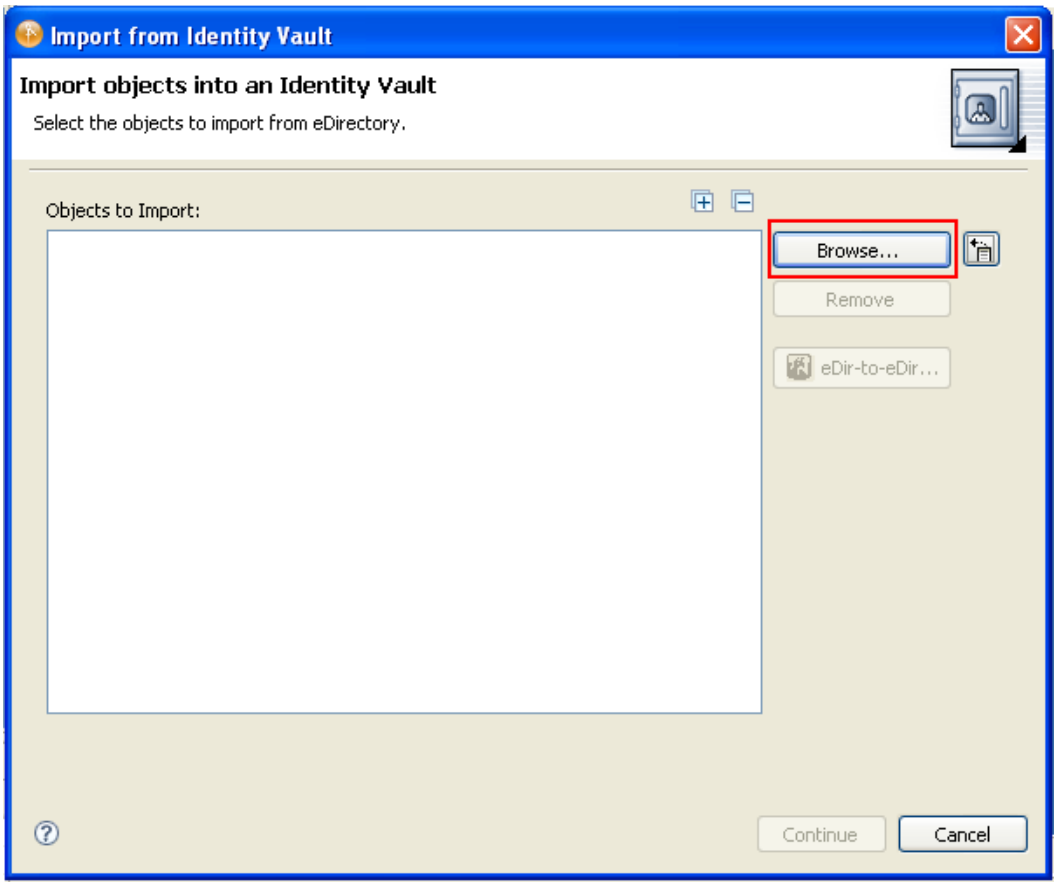
9.2 Importing Additional Objects from eDirectory

Designer provides the functionality to store additional objects (objects that are not modeled in the Designer). These objects are stored in an LDIF container under the Identity Vault. By default, there is no LDIF container. An LDIF container is created when the *Import Additional Objects* or *Import Objects from LDIF File into LDIF Container* option is selected.

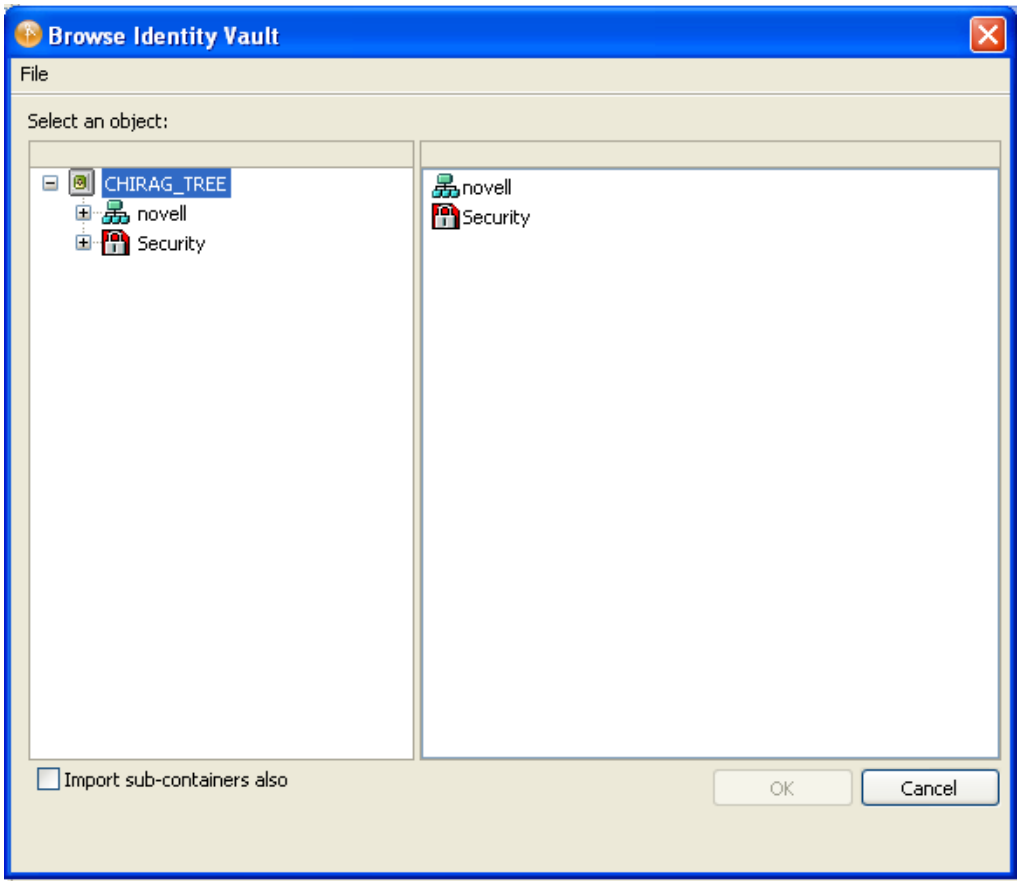
1. In the Modeler, select *ID Vault* > *Live* > *Import Additional Objects*.



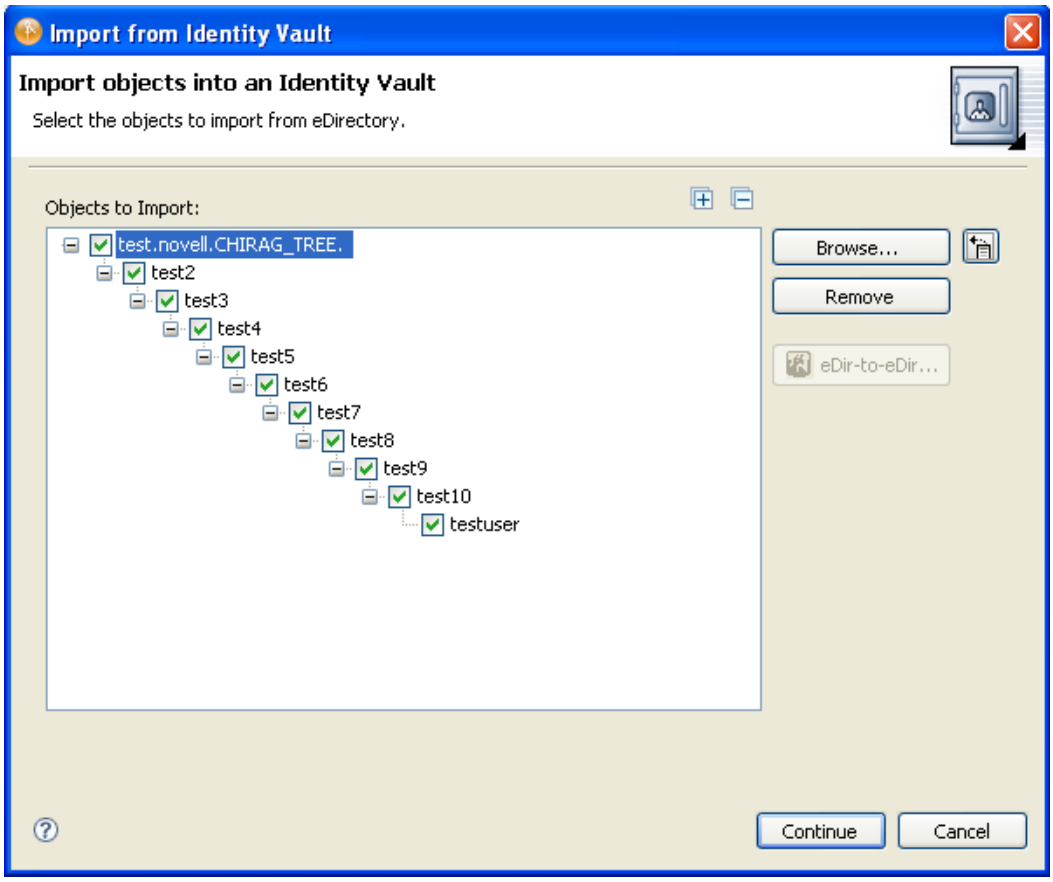
2. In the Import Additional Objects page, click *Browse* to open the eDirectory Browser.


Import from Identity Vault
Import objects into an Identity Vault
Select the objects to import from eDirectory.
Objects to Import:
Browse...
Remove
eDir-to-eDir...
Continue
Cancel

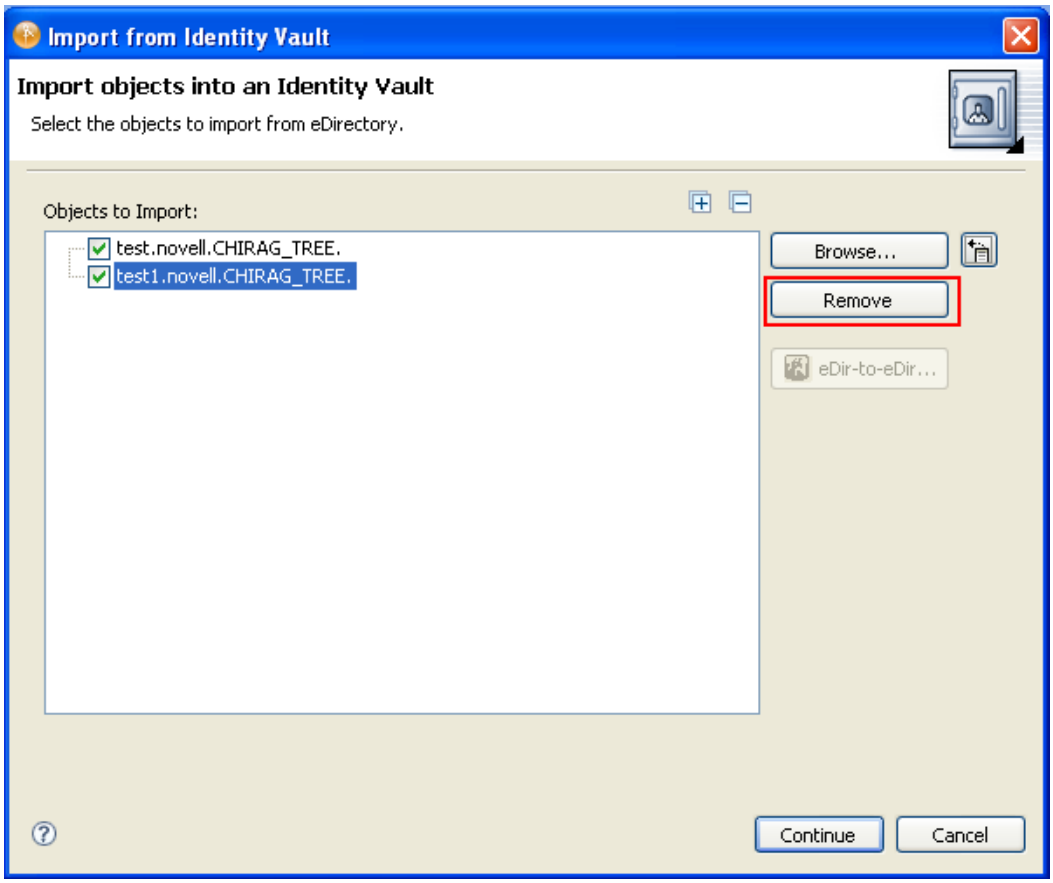
3 In the eDirectory browser page, select the objects that should be directly imported, then click *OK*. Or select the objects, select *Import sub-containers also*, then click *OK*.



The selected objects are displayed in the Import Objects dialog box.


Import from Identity Vault
Import objects into an Identity Vault
Select the objects to import from eDirectory.
Objects to Import:
test.novell.CHIRAG_TREE.
test2
test3
test4
test5
test6
test7
test8
test9
test10
testuser
Browse...
Remove
eDir-to-eDir...
Continue
Cancel

4 If you do not want to import a given object, either deselect it and continue or select it and click *Remove*.



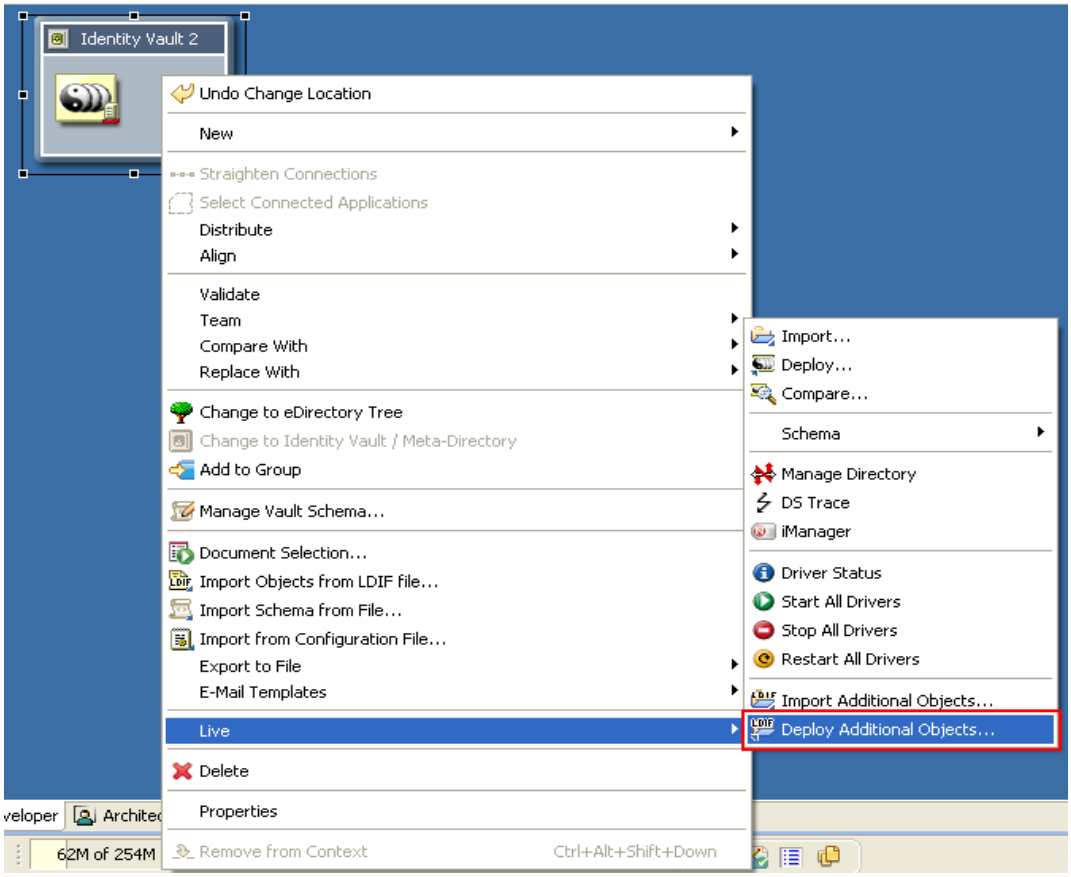
5 Click *Continue* to import the selected objects. These objects are stored in the LDIF container.

These objects are retrieved by an LDAP channel. If you are running the LDAP service on non-default ports, see Section 9.9, “Changing the LDAP Properties,” on page 214 for more information. The objects in the LDIF container are overwritten. To keep a record of the objects stored in the LDIF container, export the information of the LDIF container into an LDIF file. For more information, see Section 9.4, “Exporting LDIF Container Objects to an LDIF File,” on page 205.

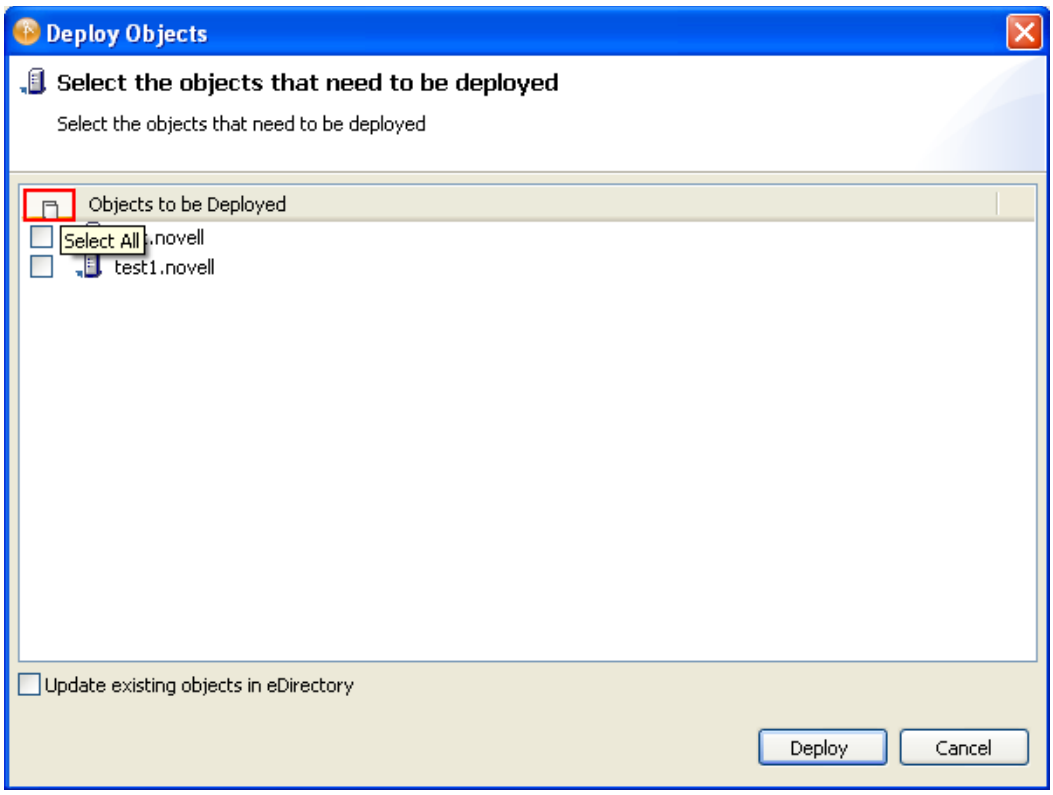
9.3 Deploying Additional Objects into eDirectory

Designer allows you to update objects that are already present in eDirectory. The current functionality does not support the deployment of objects containing references to objects in eDirectory. To deploy these objects, you can either manually create the objects in eDirectory or remove the references (see Section 9.7, “Editing the LDIF Container Data by Using an Editor,” on page 213). Designer allows you to update objects that are in eDirectory.

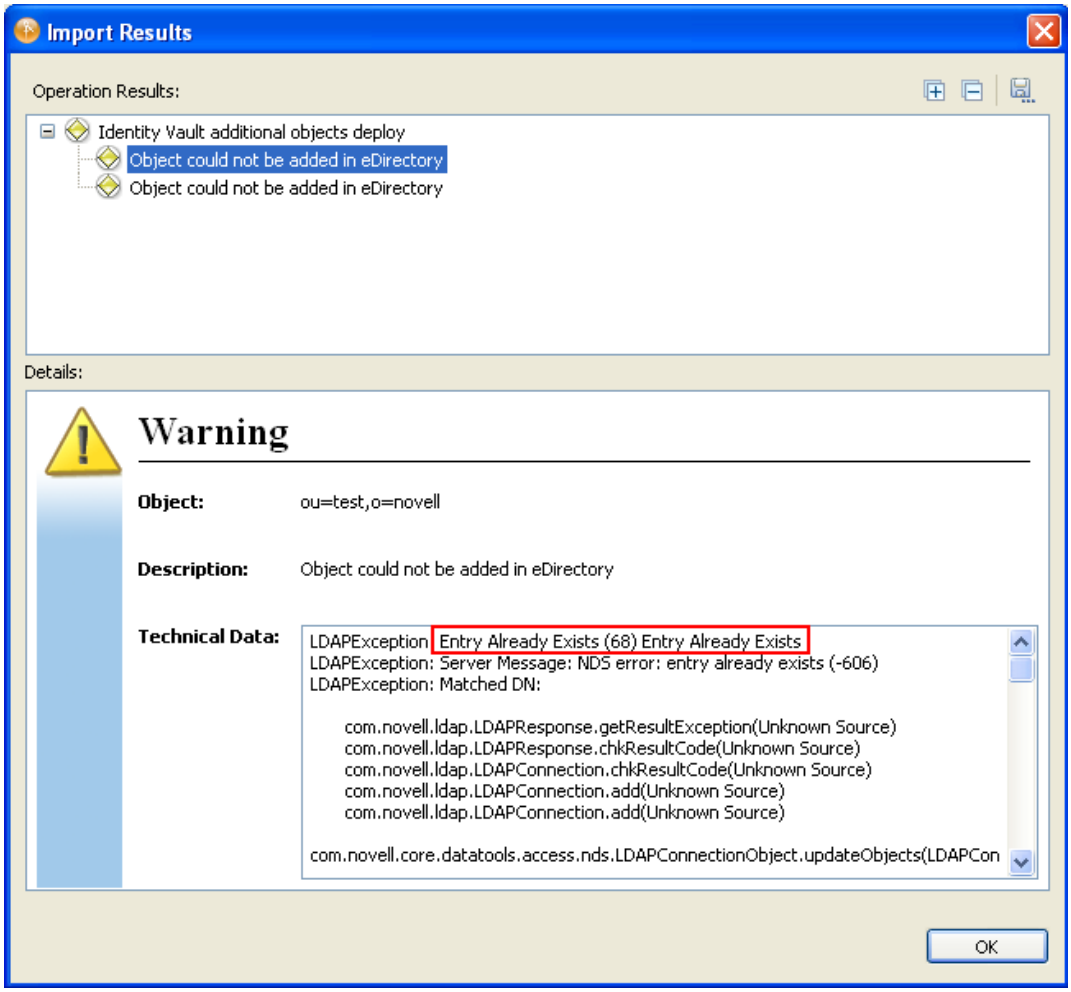
1 In the Modeler, select *ID Vault* > *Live* > *Deploy Additional Objects*.



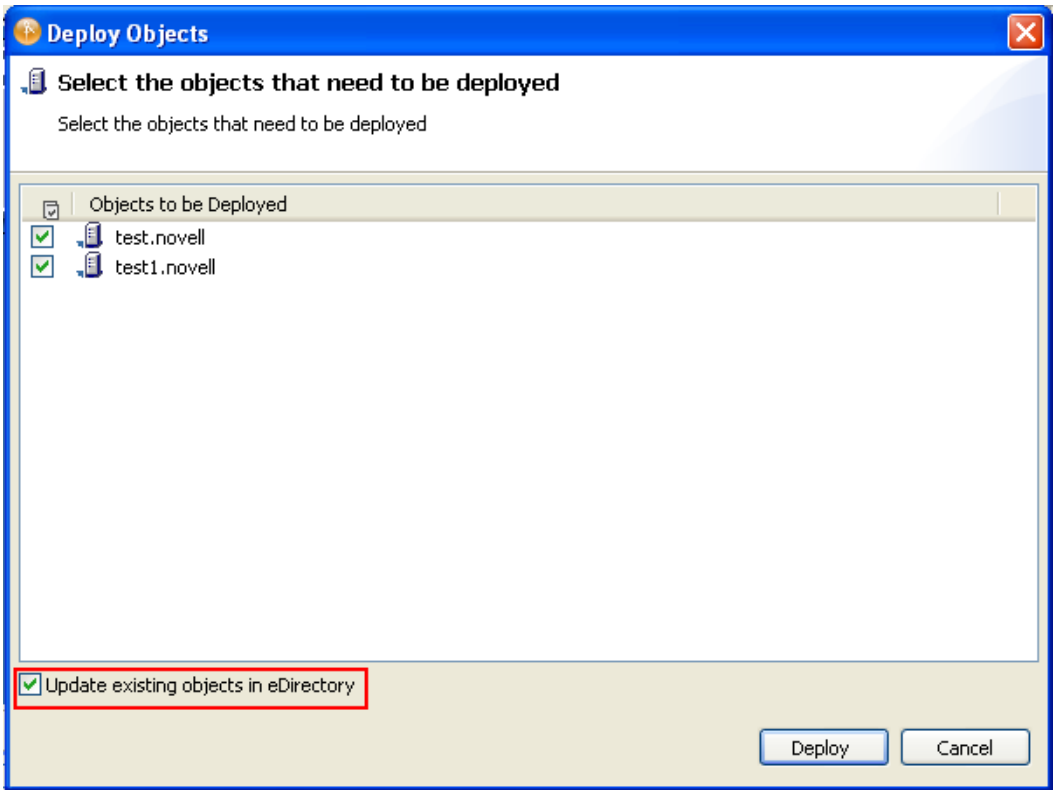
2 Select or deselect the objects by clicking *Select All*, then click *Deploy*.



If the objects are already present in eDirectory, a warning is displayed.



3 If the objects are already in eDirectory, click *Update Existing Objects in eDirectory* to update them, click *Deploy*, then click *OK*.

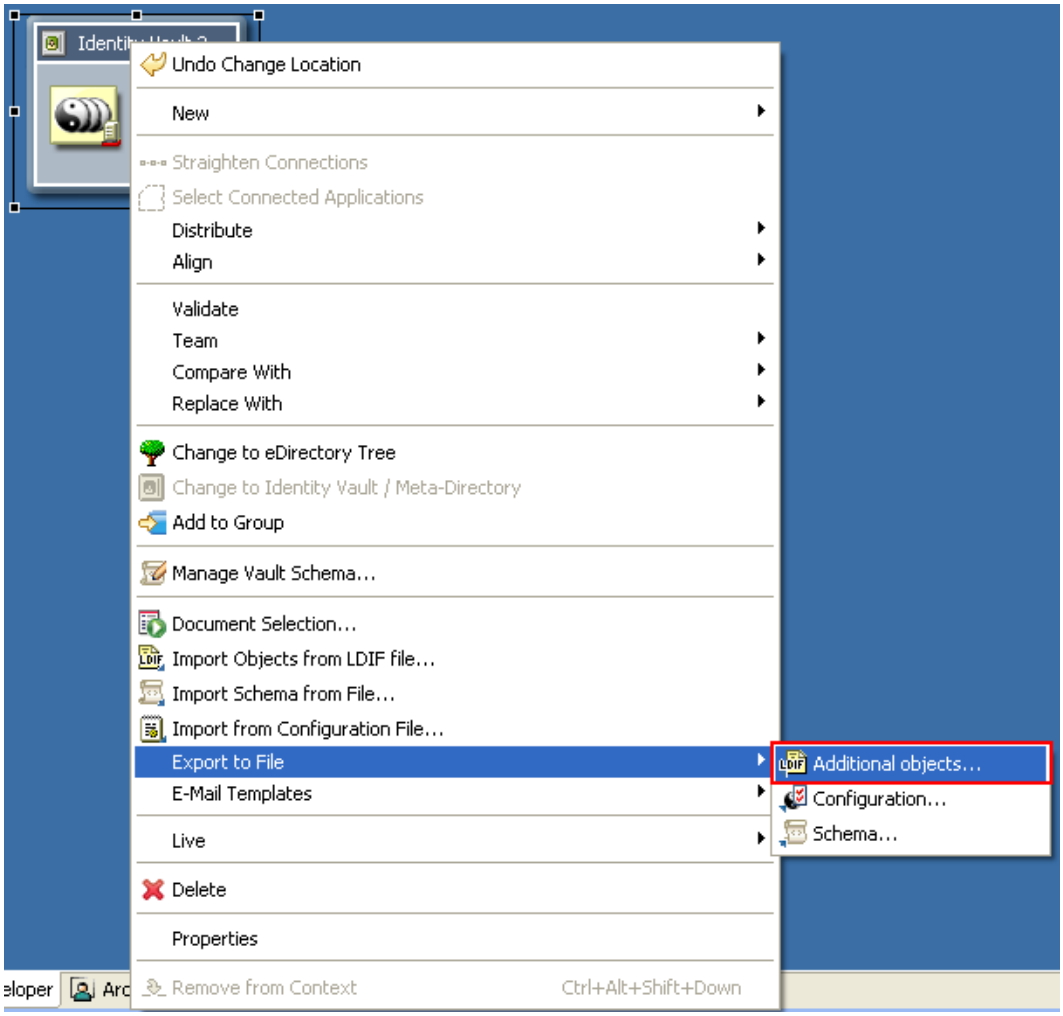


If there are no objects or the information is not in a proper format in the LDIF container, a warning is displayed.

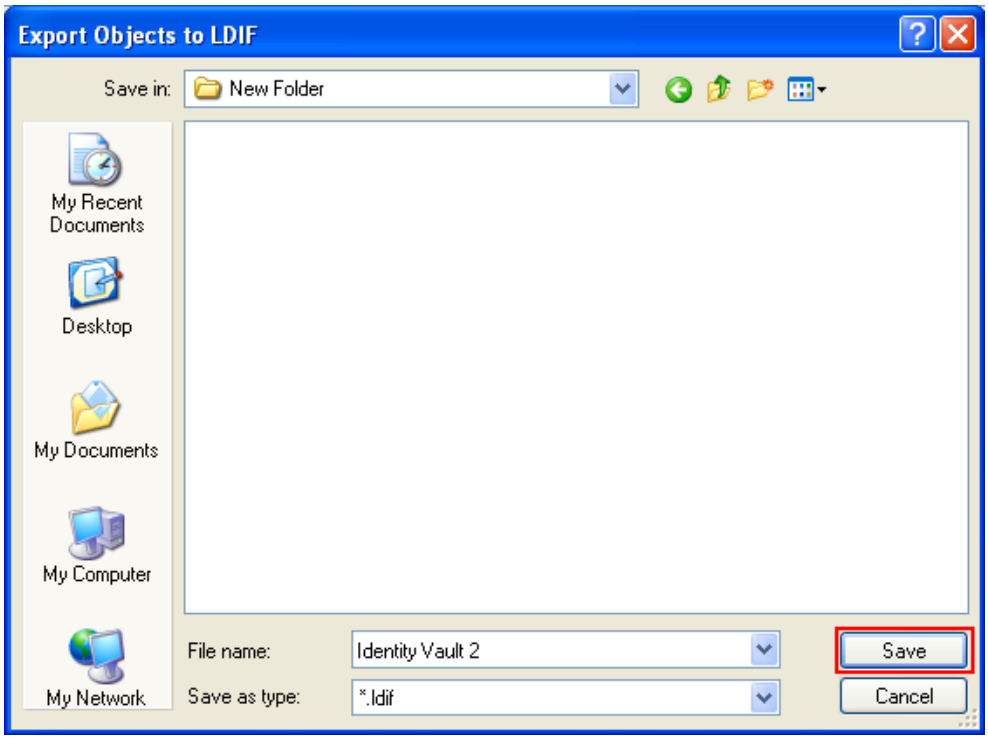
9.4 Exporting LDIF Container Objects to an LDIF File

You can export objects from the LDIF container and store them in the LDIF file.

1 In the Modeler, select *ID Vault > Live > Export to File > Additional Objects*.



2 From the file browser that displays, specify the name of the file into which objects stored in the LDIF container should be exported, then click *Save*.

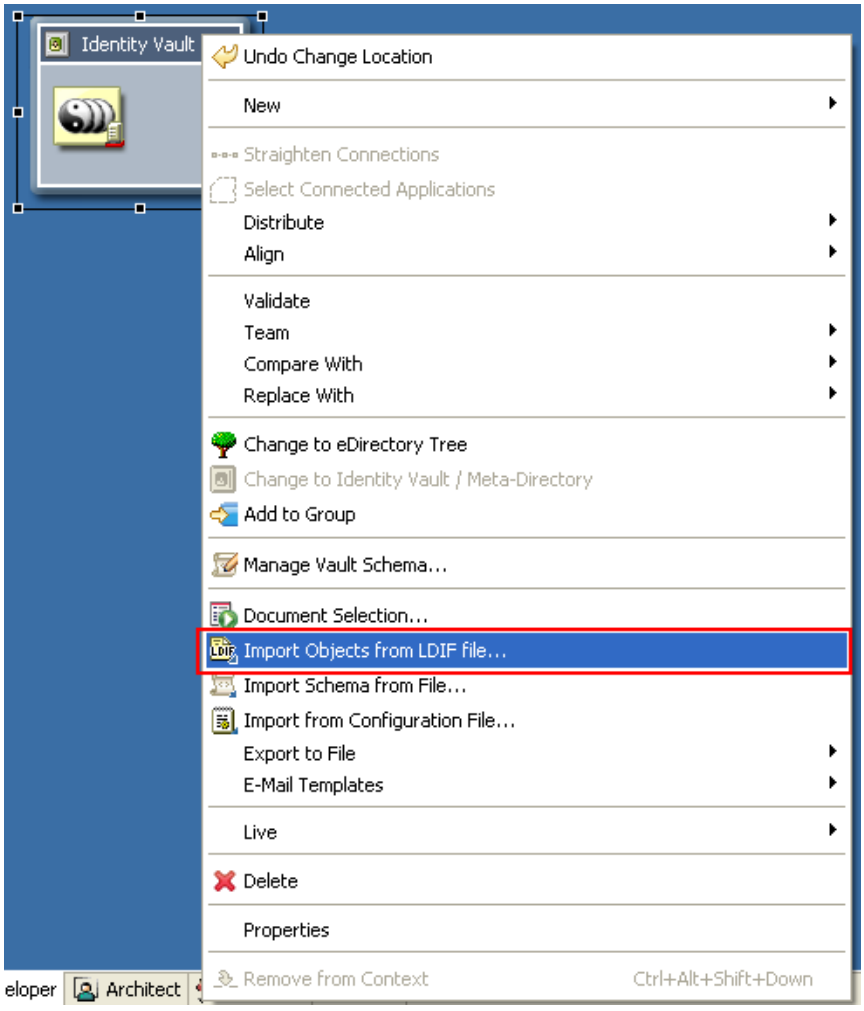


If there is no information in the LDIF container, a warning is displayed.

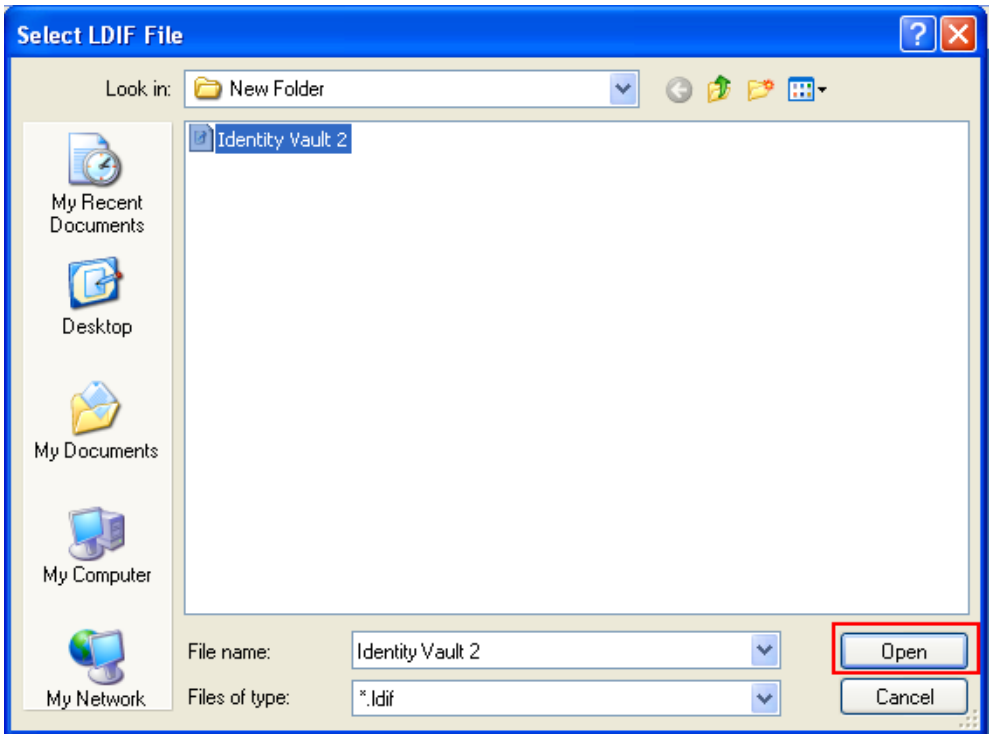
9.5 Importing Objects from an LDIF File into an LDIF Container

You can export objects from an LDIF file to an LDIF container. This overwrites the existing objects of the LDIF container.

1 In the Modeler, right-click the *ID Vault* and select *Import from LDIF File*.



2 From the file browser that displays, select the file from which the LDIF object should be imported, then click *Open*.

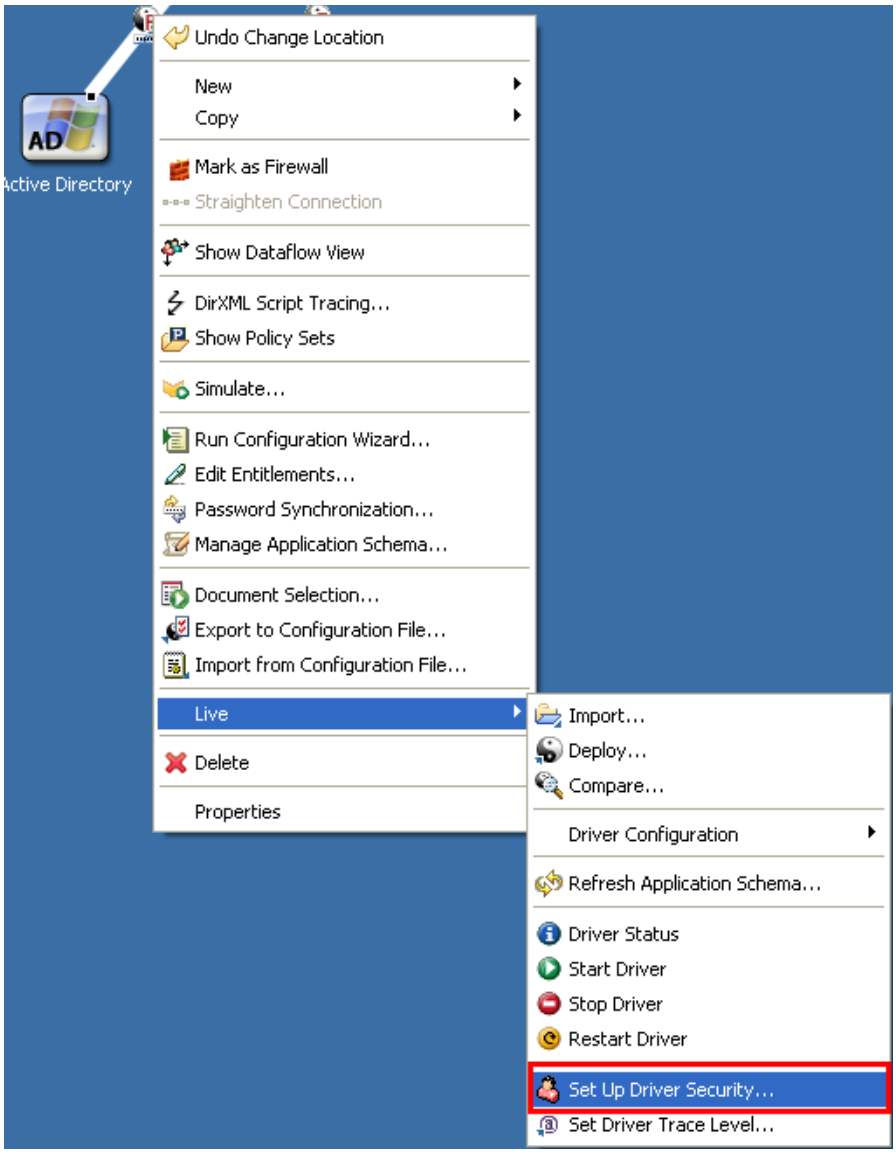


9.6 Setting up Driver Security Equivalences

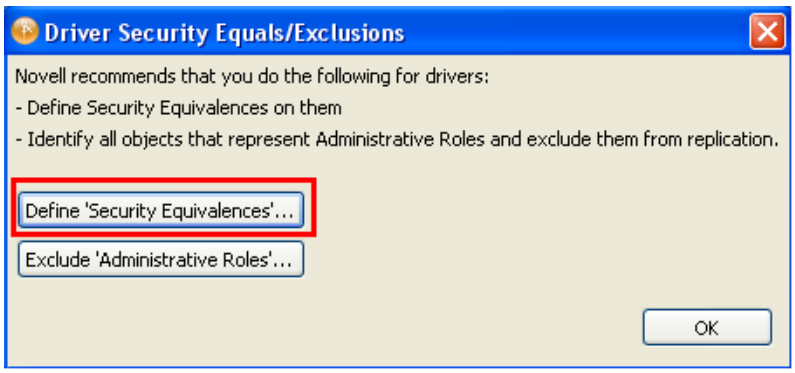
Designer allows you to define and deploy the Security Equivalences objects for the drivers in eDirectory.

To set Security Equivalences for a driver:

1 Right-click a driver and select *Live >Set up Driver Security*.

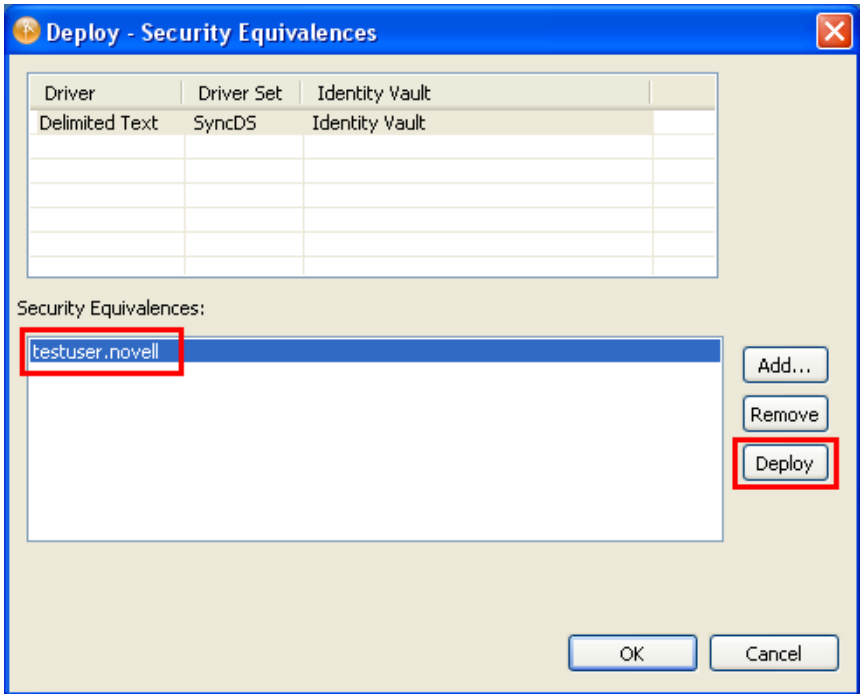


2 On the Driver Security Equivalences/Exclussions page, click *Define Security Equivalences*.



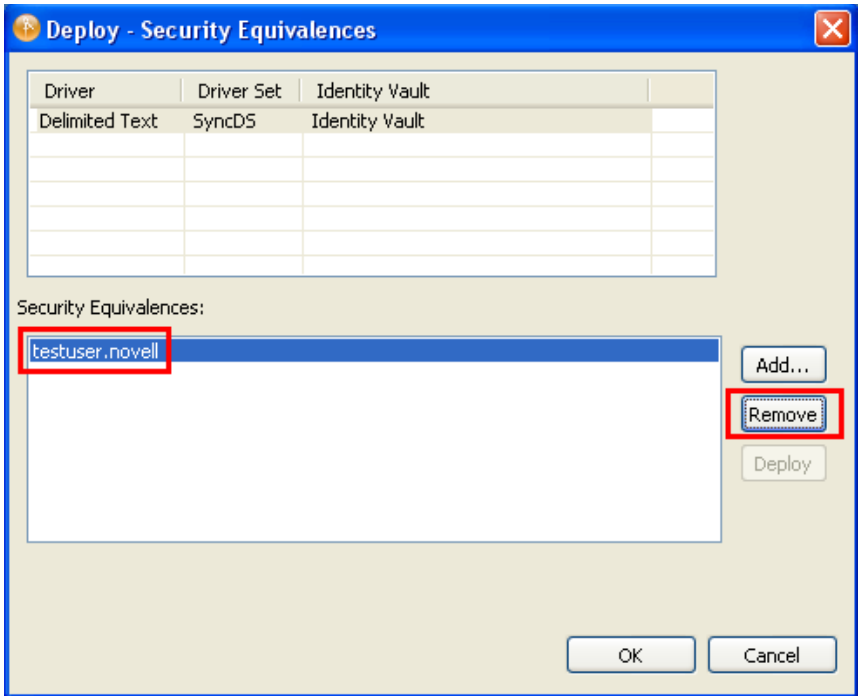
3 On the Deploy Security Equivalences page, select the Security Equivalences object from the list, click *Deploy*, then click *OK*.

For example, testuser.novell is not in eDirectory but in the LDIF container. The Deploy operation picks testuser.novell from the LDIF container and adds it to eDirectory.

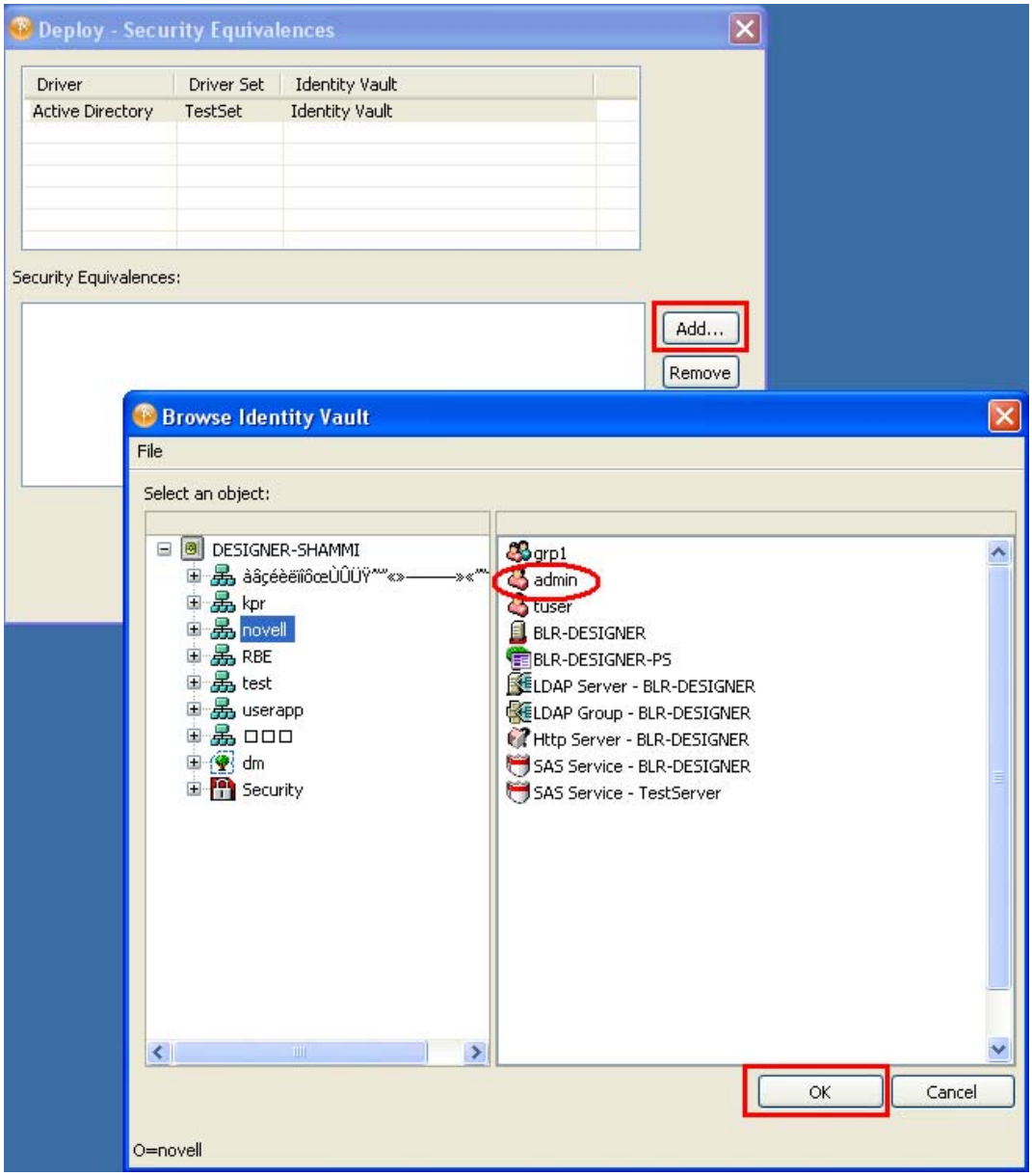


You can remove the existing Security Equivalences object and define new Security Equivalences objects for the Driver.

4 Select the Security Equivalences object, then click *Remove*.

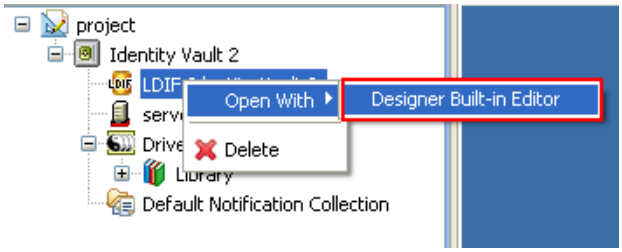


5 To add a new object, click *Add*, select the object, then click *OK*.



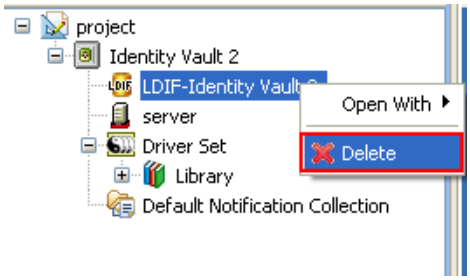
9.7 Editing the LDIF Container Data by Using an Editor

Designer allows you to modify the LDIF objects by using different editors. Under the ID Vault, right-click the LDIF Container, *then click Open With > Select Designer Built-in Editor*. Or double-click the LDIF Container to open it.



9.8 Deleting the LDIF Container

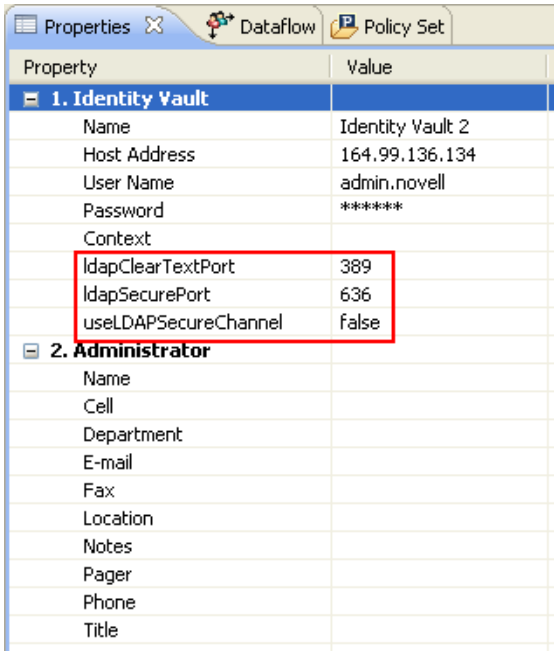
Designer allows you to delete the LDIF Container. Under the ID Vault, right-click the LDIF Container, then click *Delete*.



9.9 Changing the LDAP Properties

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

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



2 Save the project.

Setting Up E-Mail Notification Templates

10

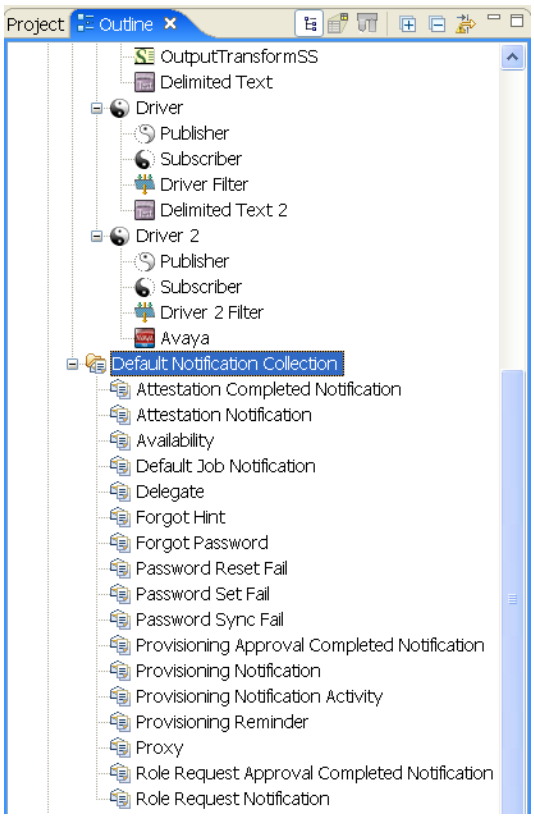
Notification templates enable you to customize and send e-mail messages that users receive when triggers occur.

- Section 10.1, “Viewing Notification Templates,” on page 215
- Section 10.2, “Editing a Notification Template,” on page 218
- Section 10.3, “Adding and Deploying a Notification Template,” on page 221
- Section 10.4, “Policy Builder and Notification Templates,” on page 223
- Section 10.5, “Configuring the E-Mail Server,” on page 223

10.1 Viewing Notification Templates

Designer provides default notification templates, which you can view or edit. To view the templates, perform the following:

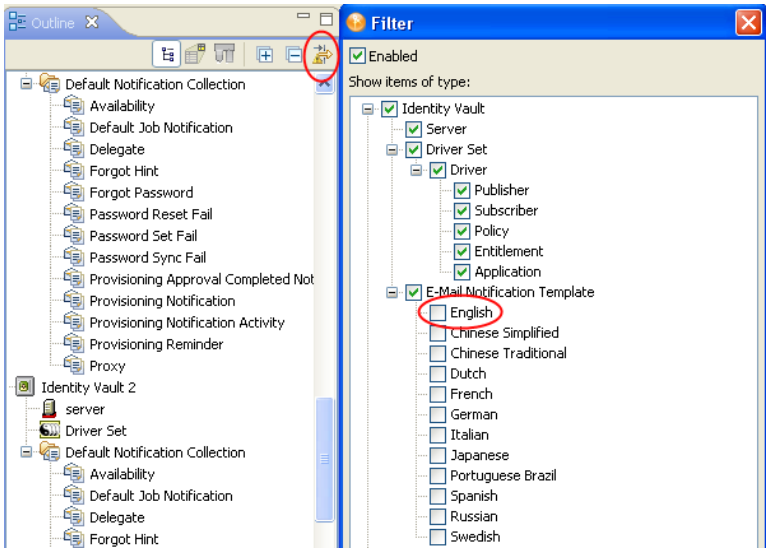
1. Select an Identity Vault in the Modeler.
2. In the Outline view, scroll to and right-click the *Default Notification Collection* for that Identity Vault.
3. Select *Add Default Templates* if you want to add the default English version of the notification templates to the Identity Vault.



If *Default Notification Collection* isn't expanded, expand it. The expanded tree shows the default notification templates. The install program no longer installs all of the notification templates with Designer.

4 Select *Add All Templates* to update all of the notification templates that are installed with Designer to the Identity Vault. You can then use the Filter option in the Outline view to filter out the notification templates that you don't want to see.

To view and edit the internationalized template files, click the *Filter* button in the Outline view, then select languages that you want to see.



5 If you have a certain template that you want to have all of the localized templates, right-click that template and select *Add Localized Templates*. All of the localized templates are added for the selected template. Use the *Filter* button to select the languages you want to see.

6 Use the templates in the Default Notification Collection to send e-mail notifications to users in the Identity Vault. You can customize these templates with your own text.

Right-click a template (for example, *Forgot Hint*), then select *Edit*.

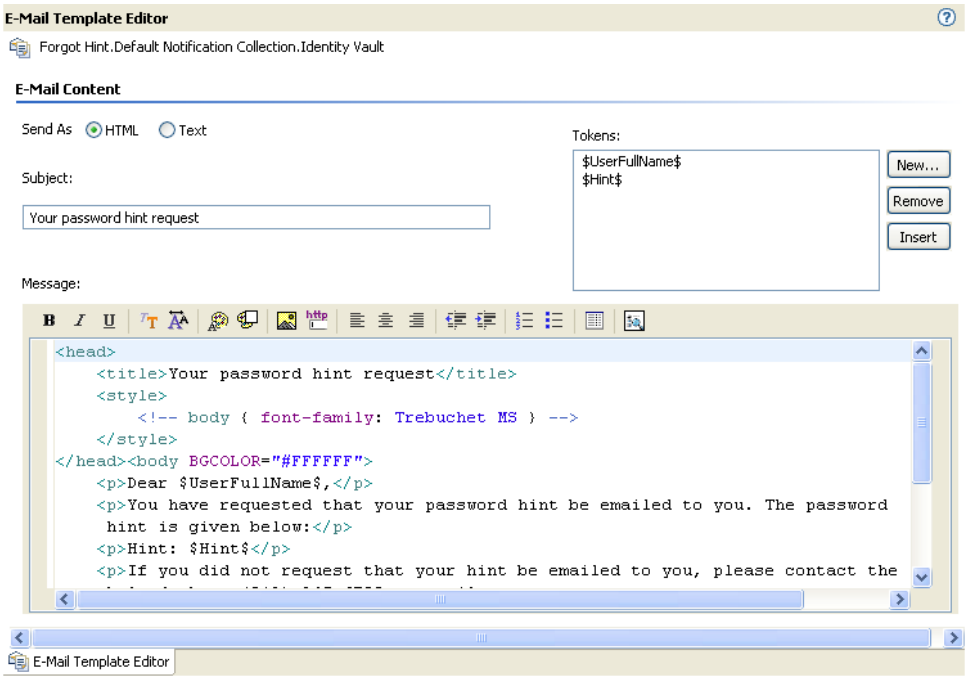
You can also open a template by double-clicking it.

Template Name	Description
Attestation Completed Notification	Sends an e-mail notification when the workflow process for your attestation request is completed.
Attestation Notification	Sends an e-mail notification when a new compliance activity is submitted that requires your attention.
Availability	Sends an e-mail notification when an availability setting has been created or modified.
Default Job Notification	Sends an e-mail notification to report results of the job as configured in the template. Contains the name of a job and any status information from the job.
Delegate	Sends an e-mail notification when a a delegate assignment has been created or modified
Forgot Hint	Sends an e-mail notification when a user forgets a password and requests a hint.
Forgot Password	Sends an e-mail notification when a user incorrectly enters a password.
Password Reset Fail	Sends an e-mail notification when a user tries to reset a password but doesn't meet password policy requirements.
Password Set Fail	Sends an e-mail notification when a user's password cannot be set in the connected system.
Password Sync Fail	Sends an e-mail notification when a user's password fails to synchronize.
Provisioning Approval Completed Notification	Sends an e-mail notification when a workflow is completed. Indicates the overall workflow and provisioning decision.
Provisioning Notification	Sends an e-mail notification to a user or manager for approval. Indicates that action is required from the user or manager.
Provisioning Notification Activity	Sends an e-mail notification to a user or manager about the activity of the provisioning notification.

Template Name	Description
Provisioning Reminder	Sends an e-mail notification when a user activity timeout expires. Reminds the user or manager to act.
Proxy	Sends an e-mail notification when a proxy assignment has been created or modified.
Role Request Approval Completed Notification	Sends an e-mail notification to a user or manager that the approval process is completed.
Role Request Notification	Sends an e-mail notification to a user or manager that a new role request requires approval.

10.2 Editing a Notification Template

1. Select an Identity Vault.
2. In the Outline view, right-click a template (for example, *Forgot Hint*), then select *Edit*.
3. Select a format, specify a subject, add tokens, customize the message that users receive, then save and close the template.



- Section 10.2.1, “Selecting a Format,” on page 219
- Section 10.2.2, “Specifying a Subject,” on page 219
- Section 10.2.3, “Working with Tokens,” on page 219
- Section 10.2.4, “Editing a Template Message,” on page 220

10.2.1 Selecting a Format

Select whether users receive this e-mail notification in HTML or text format.

Figure 10-1 *Options for Sending the Notification*

10.2.2 Specifying a Subject

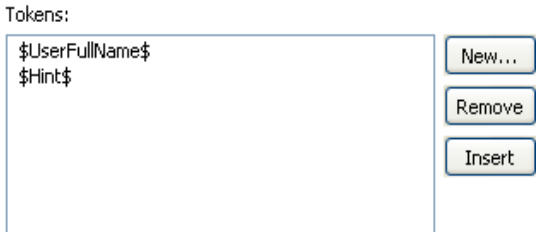
The subject is the text that a user views in an e-mail's Subject heading or field. You can change the text in the Subject field. You can also use tokens here. The text or tokens don't determine the name of the template.

Figure 10-2 *The Subject Field*

10.2.3 Working with Tokens

A token is a variable or replacement tag for items such as the user's name. Tokens help you personalize the message to the user.

Figure 10-3 *The Tokens Dialog Box*

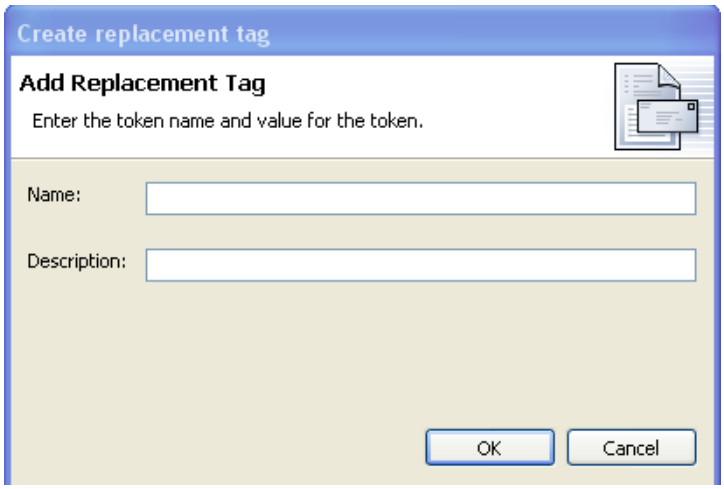


Each template includes default tokens. For example, the Forgot Password e-mail template for sending a password to the user includes the default replacement tag named $CurrentPassword$.

You can define other tokens for use in the body of the message or in the subject. Your ability to do so depends on the application that uses the templates. To find out how to define additional replacement tags, see the documentation for the application. For example, Identity Manager Password Synchronization can't use a replacement tag that you create unless the policy in the driver configuration that uses the template also contains the definition of the replacement tag.

Adding a Token

1. Click *New*.



2 In the *Create a Replacement Tag* dialog box, type a name for the token.

   You don't need to type the $ characters. Designer provides them.

3 Type a description for the token.

4 Click *OK*.

When you add a token, the tag is automatically added to the XML source for the template. After you add a tag, you can edit it only in the XML Source view.

Removing a Token

To delete a token, select it, then click *Remove*.

Make sure that you don't remove tags that are needed for the body of the message.

Inserting a Token

1 In the template, click where you want to insert a token.

2 Select a token.

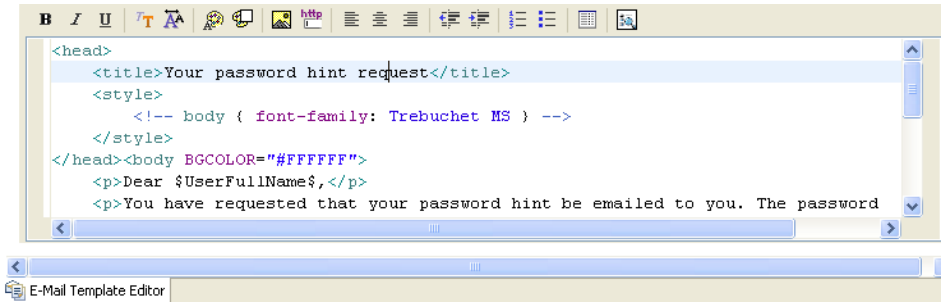
3 Click *Insert*.

   Designer inserts the selected token into the e-mail template.

10.2.4 Editing a Template Message

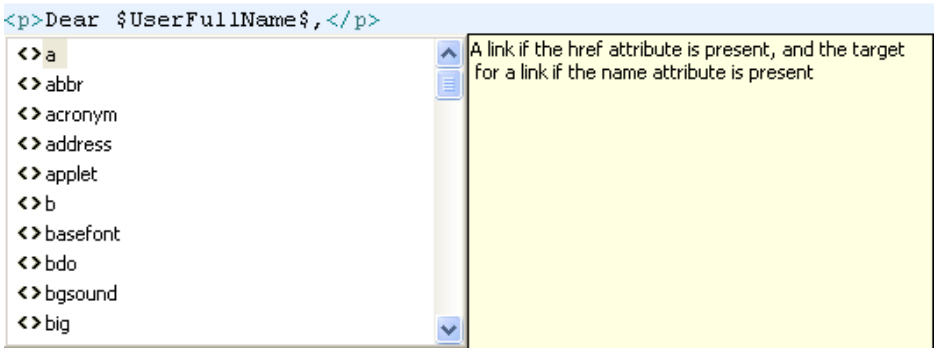
The text of the e-mail message appears in the *Message* field. Customize the text so that it suits your environment. Use tokens to personalize the e-mail message.

Figure 10-4 *The Message Edit Box*



1 In the *E-Mail Template Editor*, place your cursor in the Message edit box, then press Ctrl+Spacebar.

2 Select an HTML tag by double-clicking a tag in the drop-down list.



3 Format text by using the toolbar.

4 Preview the text by clicking the Preview icon .

5 Save the template by selecting *File* > *Save*.

You can also click the *Save* icon.

If the code isn't valid, you can't save the template.

10.3 Adding and Deploying a Notification Template

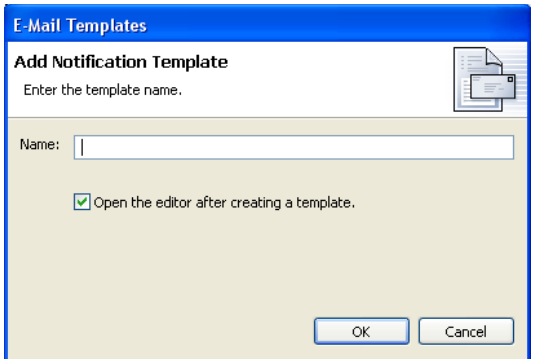
- Section 10.3.1, "Adding a Notification Template," on page 221
- Section 10.3.2, "Importing a Notification Template," on page 222
- Section 10.3.3, "Deploying a Template," on page 222

10.3.1 Adding a Notification Template

1 Select an Identity Vault in the Modeler.

2 In the Outline view, scroll to *Default Notification Collection* for that Identity Vault.

3 Right-click, then select *New Template*.



4 Name the template.

5 If you want to automatically open the template editor so that you can view or edit the template, select *Open the editor after creating a template*.

6 Click *OK*.

7 Customize the text by editing the template message.

8 Click *Save* on the Designer tool bar.

10.3.2 Importing a Notification Template

To import a notification template from a file:

1 In the Outline view, scroll to *Default Notification Collection* for an Identity Vault.

2 Right-click, then select *Import Template from File*.

3 Browse to and select the template.

4 Customize the text for your environment by editing the template message.

To import a notification template as a live operation:

1 In the Outline view, scroll to *Default Notification Collection* for an Identity Vault.

2 Right-click, then select *Live* > *Import*.

3 Specify the host name (IP address) for the tree.

4 To authenticate, specify the user name and password.

5 Browse to and select the template, then click *OK* > *Continue* > *Import* > *OK*.

6 Customize the text for your environment by editing the template message.

10.3.3 Deploying a Template

After you add or import a template, deploy it.

1 Customize the text for your environment by editing the template message.

2 Right-click the template.

3 Select *Live* > *Deploy*.

10.4 Policy Builder and Notification Templates

For information on using the Policy Builder interface to send e-mail notifications, see Send Email (http://www.novell.com/documentation/idm36/policy_designer/data/dosendemail.html) and Send Email from Template (http://www.novell.com/documentation/idm36/policy_designer/data/dosendemailfromtemplate.html) in the *Policies in Designer 3.5* guide.

10.5 Configuring the E-Mail Server

The e-mail server sends notification e-mails from applications that use the Notification Configuration templates.

1. Select an Identity Vault in the Modeler.
2. In the Outline view, scroll to *Default Notification Collection* for that Identity Vault.
3. Right-click, then select *Properties*.
4. Specify the host name, *From*, and authentication settings for your SMTP e-mail server.

   Host Name: The host name of your SMTP e-mail server. This can also be an IP address.

   From: When a user opens the e-mail, the text that you enter in the *From* edit box is displayed in the *From* field of the user's e-mail heading. Depending on your mail server settings, the text in this field might need to match a valid sender in the system (for example, helpdesk@company.com instead of descriptive text such as The Password Administrator). Such a match allows the mail server to do reverse lookups or authentication.

   Authenticate to the server by using credentials: Use for a secured SMTP server.

   If your server requires authentication before sending e-mail, specify the username and password here.

   Although the authentication information is specified here, you might also need to specify it separately for the application that is sending the notification e-mails.

   For example, Forgotten Password e-mail notifications can be sent by using the authentication information you specify here. However, notification e-mails for Identity Manager Password Synchronization require the authentication information to be provided in the driver policy that is used to send notification e-mails.
5. Click *OK*.

Importing into Designer

11

Designer's Import feature allows you to import the following items into defined projects:

- Projects from the File System or from the Identity Vault (eDirectory)
- Libraries
- Driver sets
- Individual drivers
- Channels
- Policies
- Schema

Depending on a project's complexity, importing can save you time in building and rebuilding drivers, channels, and policies. For instance, after a driver, channel, or policy is built, you can import it into new projects and modify it to run in the new environment instead of starting from scratch on each new driver, channel, or policy.

You import projects, drivers, channels, schema, and policies from an existing eDirectory tree running the Identity Manager system (Identity Vault) or from an exported project located in the file system. In Designer, use the Deploy feature to save drivers, channels, and policies into a Metadirectory server in an Identity Vault. Use the Export feature to save project, drivers, channels, and policies to a local, removable, or network directory.

What you can import depends on the level you are at within a project. For instance, at the *File* > *Import* level, you can import a project from the file system or from an Identity Vault. At the project level, you can import driver sets, drivers, channels and policies; at the driver level, you can import channels and policies, and at the channel level, you can import policies.

During import, Designer does not import anything that is encrypted. This includes named passwords, e-mail notifications, existing certificates, and the Secure Login administrator password.

- Section 11.1, "Importing Projects," on page 225
- Section 11.2, "Importing a Library, a Driver Set, or Driver from the Identity Vault," on page 236
- Section 11.3, "Importing a Driver Configuration File," on page 245
- Section 11.4, "Importing Channels, Policies, and Schema Items from the Identity Vault," on page 250
- Section 11.5, "Using the Compare Feature When Importing," on page 259
- Section 11.6, "Error Messages and Solutions," on page 267

11.1 Importing Projects

Designer's Import feature allows you to import projects from the File System or from an Identity Vault (eDirectory). What you are able to import depends on the level you are at within a project. To import an eDirectory object, you must have sufficient rights to access the eDirectory tree that is associated with the Identity Vault you are designing.

When you initially open the Designer utility and close the Welcome view, you have no projects that are currently available.

For information on creating a new project, see Chapter 2, “Creating a Project,” on page 23. For information on importing projects, see the following sections:

- Section 11.1.1, “Importing a Project from the Identity Vault,” on page 226
- Section 11.1.2, “Importing a Project from the File System,” on page 231
- Section 11.1.3, “Importing a Project from iManager,” on page 233
- Section 11.1.4, “Importing a Project from a Version Control Server,” on page 234

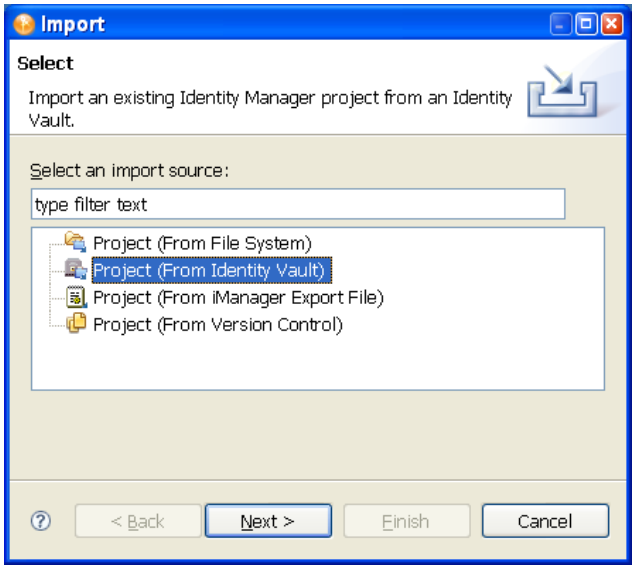
11.1.1 Importing a Project from the Identity Vault

1. In Designer, click *File* > *Import*.

   or

   Click *Import Project From Identity Vault* from the No Projects Available page in the Projects view, then skip to Step 3.

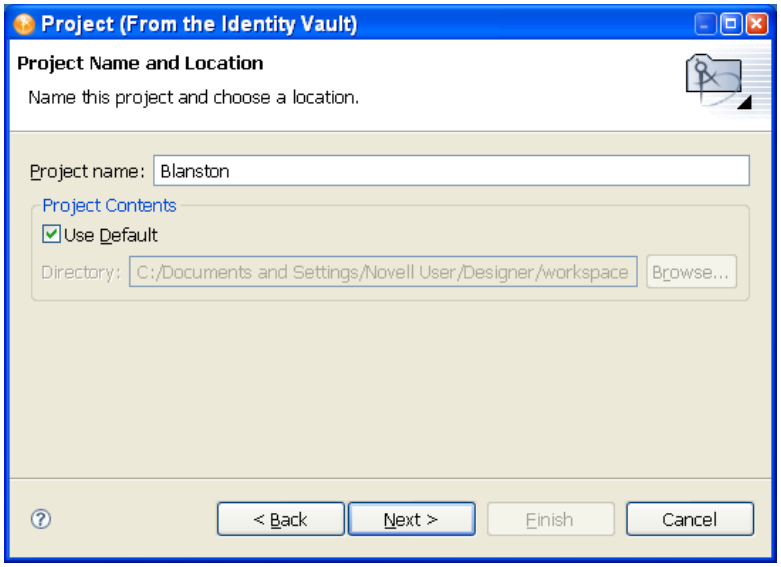
2. From the Import window, select *Identity Manager Project (From Identity Vault)* under the Designer for Identity Manager heading. Click *Next*.



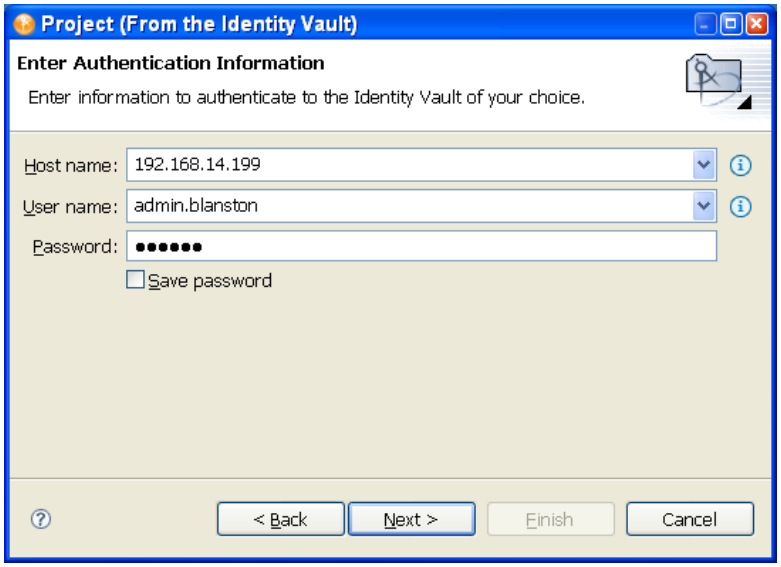
   The Import window allows selections under a number of tabs, including *Designer for Identity Manager*. Selections under the *Designer for Identity Manager* tab include:

   - Importing an existing Identity Manager project from the file system (the project must have a valid `.project` file).
   - Importing an existing Identity Manager project from an Identity Vault.
   - Importing an existing Identity Manager project from an iManager export file (Driver Set or Driver).
   - Importing an existing Identity Manager project from a version control server.

3 In the Project (From Identity Vault) window, give the new project a name. Select where the contents of this project are to reside (for Windows workstations, the default is `C:\Documents and Settings\Username\designer_workspace`). Click *Next*.



4 In the Import Project From Identity Vault window, specify the information needed to authenticate to the Identity Vault (eDirectory) of your choosing. In the *Host Name* field, you can use either a tree name or the IP address of the Identity Vault.



5 Fill in the *User Name* and the *Password* fields.

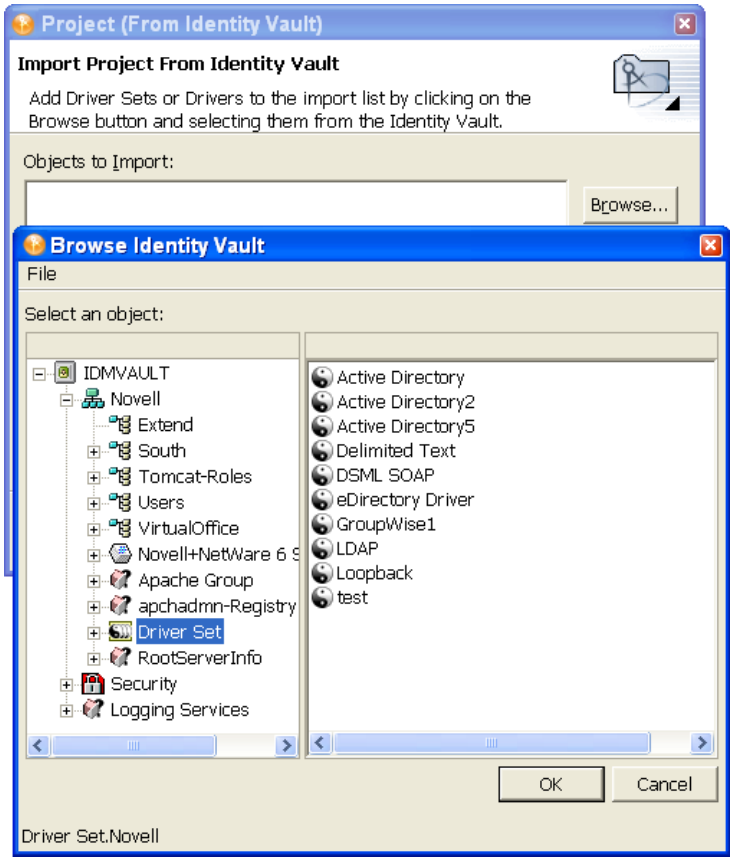
If you select *Save Password*, Designer remembers your password. Otherwise, the next time you close Designer, you need to re-authenticate when you import, deploy, or compare Designer objects with the Identity Vault.

You can use history drop-down lists to choose a previously entered value from a list.

6 Click *Next*.

7 In the Import Project From Identity Vault page, the Identity Vault Schema and the Default Notification Collection are added as import options. If you do not want to import one of these options, select the option and click the *Remove* button.

8 In the Import Project From Identity Vault page, click the *Browse* button to select the object you want to import within eDirectory. Click *OK* to return to the Import Project From Identity Vault page.



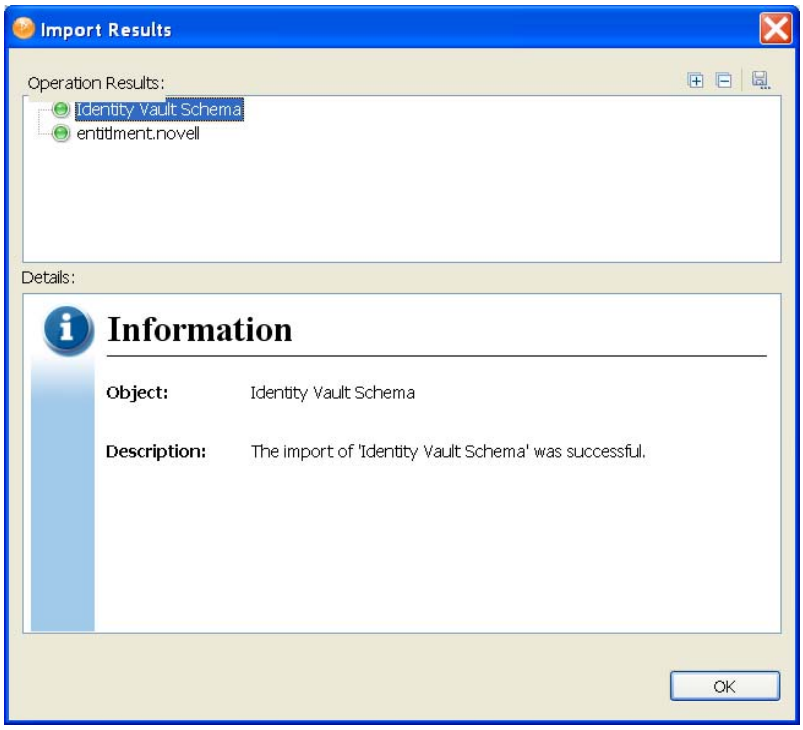
9 If there are drivers you do not want to import with the driver set, select the driver and press the *Remove* button. You can import multiple driver sets during the import operation. Just browse to the various objects that you want to import and add them to the list.

Driver sets that are not associated with a server have a red minus sign in the lower right portion of the driver set icon. These driver sets need a server association in order to be deployed.

10 (Conditional). You can also import policy libraries. Select the *Browse* button to browse to and select the library you want to import, then click *OK* to add the library to the Import Project from Identity Vault page.

11 After you have selected the objects you want to import, click *Finish*.

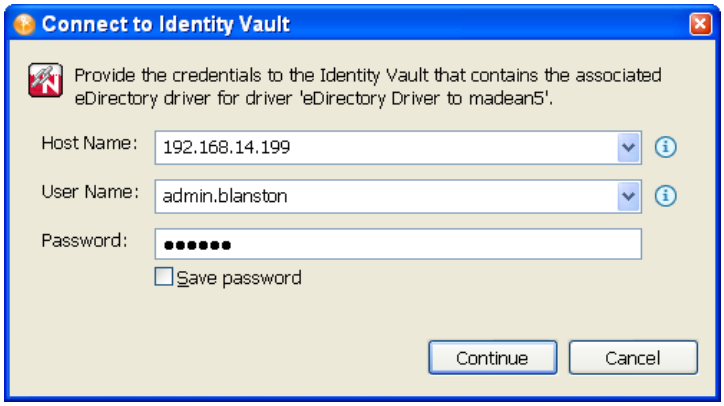
When the driver set imports, you see the Import Results window showing you if there were any problems with the import procedure.



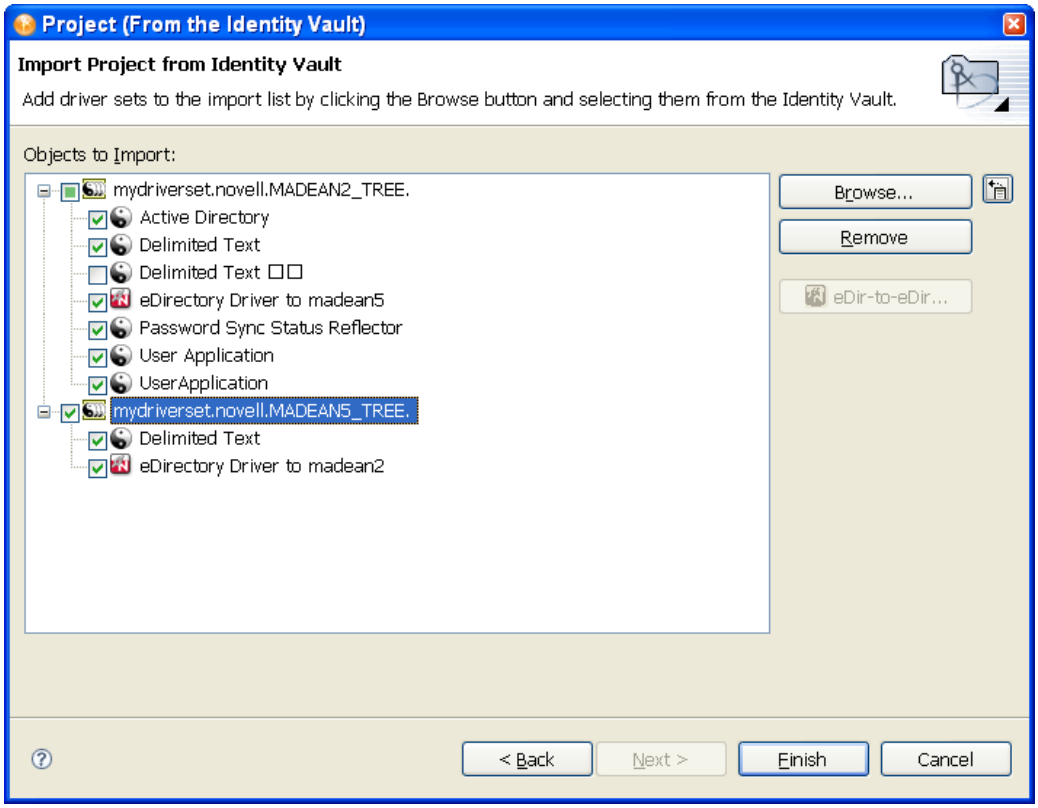
Errors during the import procedure are displayed with a red icon, and you see an Error description that is related to the operation results. If you have multiple errors, selecting an error displays the error's description in the *Details* > *Description* field. See "Error Messages and Solutions" on page 501 for further information.

12 To close the Import Results page, click *OK*.

13 (Conditional) If you are importing more than one eDirectory driver, select the eDirectory driver in the Objects to Import window and click the *eDir-to-eDir* button in the Import Project From Identity Vault page to display the Connect to Identity Vault window, where you can import the associated driver in the other eDirectory trees. Novell recommends that you import both eDirectory drivers, especially if you have SSL/TLS enabled. Provide the username and password, then click *Continue*. (Skip this step if you only want to import one eDirectory driver.)



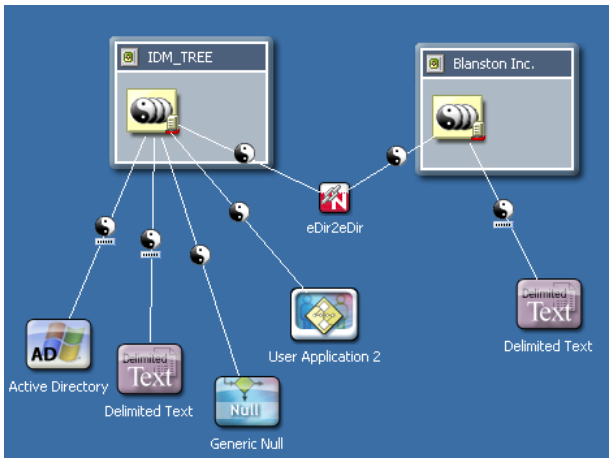
14 (Conditional) If you specify the username and password and select *Continue* to import both eDirectory drivers, you then see a Browse Identity Vault window where you select the corresponding eDirectory driver. Select the driver and click *Finish*.



You are returned to the Project (From Identity Vault) window, where you can select or deselect the drivers, allowing you to choose only the drivers in a driver set that you want to import.

15 Click *Finish*.

By having both drivers available, you can view the complete data flow between the two eDirectory drivers, as well as the other drivers you selected.



11.1.2 Importing a Project from the File System

Earlier Designer workspaces are not compatible with Designer 3.5 and above. Designer stores projects and configuration information in a workspace. These workspaces are not compatible from one version of Designer to another. You need to point Designer 3 to a new workspace, and not to a workspace used by a previous version of Designer.

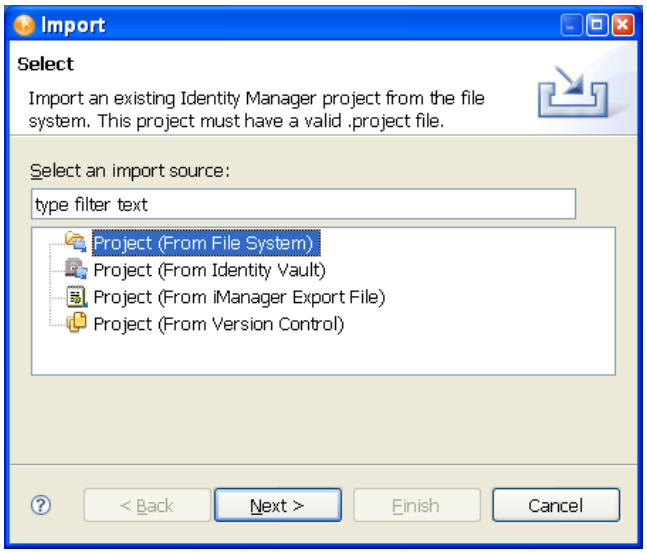
If you have Designer 2.x or 3.5 Milestone projects, import the projects into Designer 3 (*File* > *Import* > *Project from File System*). Be sure *Copy project into the workspace* is selected. Importing the project runs the Converter Wizard, making the project compatible with Designer 3.5 architecture and placing it under your designated Designer 3 workspace directory (`designer_workspace` by default).

1. In Designer, click *File* > *Import*.

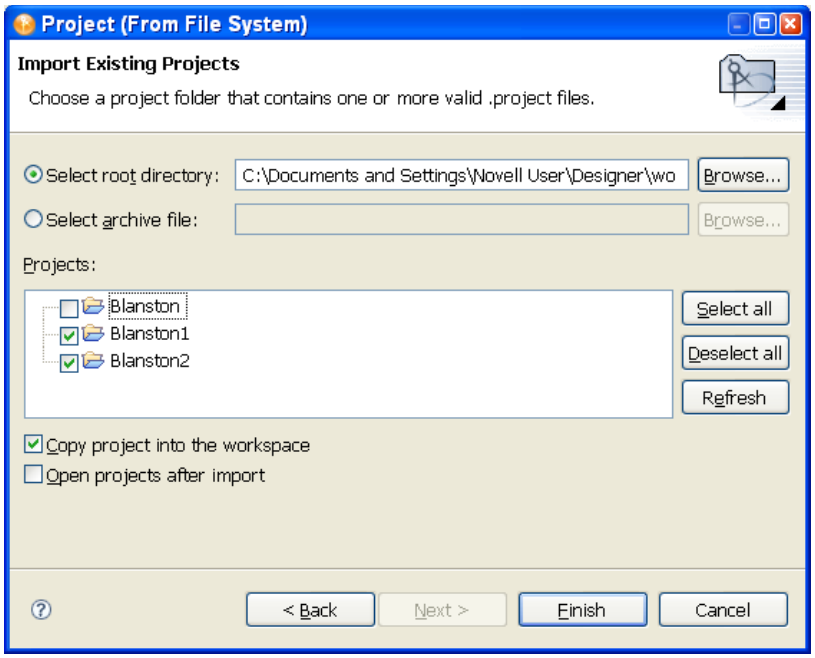
   or

   Click *Import from file system* from the No Projects Available page in the Project's view, then skip to Step 3.

2. From the Import window, select *Designer for Identity Manager* > *Project (From File System)*, then click *Next*.



3. From the Import Existing Projects page, select between the root directory or archive file. The default is *Select root directory*. Browse to the directory containing valid projects.



4 Select the directory where the exported project is saved with the `.project` extension.

   There must also be a valid project file in the selected directory, or the project does not display in the Projects window. If you have multiple projects you want to import and they are under the same directory, select the directory above the projects and click *OK*. In the Import Existing Projects window, you can select or deselect any of the projects, then click *Finish*.

5 Make sure the *Copy Project into Workspace* option is selected to copy the contents of the project into the workstation's local workspace. (Do not use previous Designer workspaces for Designer 3.5.)

   You can also import multiple projects at the same time by specifying the base or root directory where you want to start your search. The Import Wizard searches the selected directory and all of its subdirectories for valid Designer projects to import. You can then select the projects that you want to import into Designer by using the check boxes. If a project with the same name already exists in Designer, you can't import that project and you won't be able to select it from the list.

6 (Optional) If you have selected multiple projects, select whether to open these project's directories in the Model view. Designer won't open all of the projects that are imported from the file system unless you select *Open projects after imports*.

7 (Optional) You can also import projects from ZIP or TAR archives. Click the *Select archive file* selection and select the directory where the exported project is saved with the `.zip` or `.tar` extension. The whole archive is searched for valid Designer projects to import.

   If the Projects you are importing need to be converted to this version of Designer, you will see the Project Converter window. See Section 17.1, "Converting Earlier Projects," on page 373 for more information.

8 In the Import Existing Projects window, you can select or deselect any of the projects, then click *Finish*.

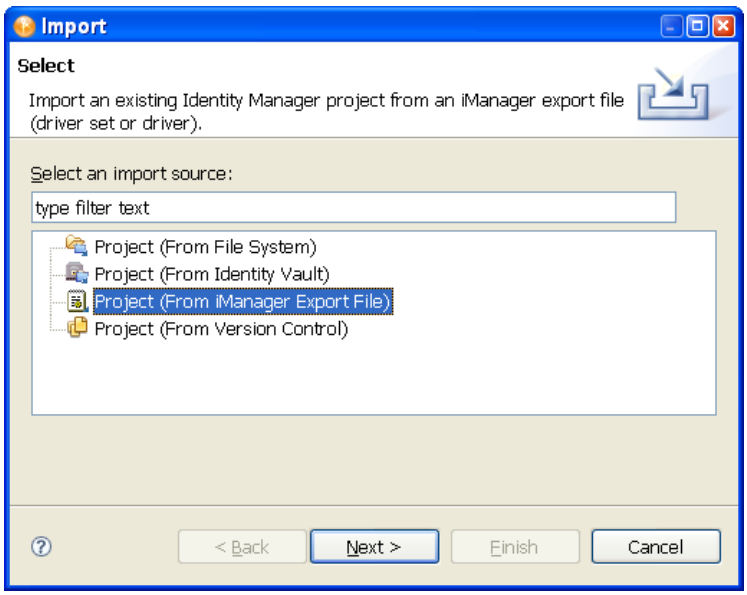
11.1.3 Importing a Project from iManager

You can create a new Designer project by selecting an iManager `.xml` export file. These include driver set and driver exports as well as Novell sample configuration files.

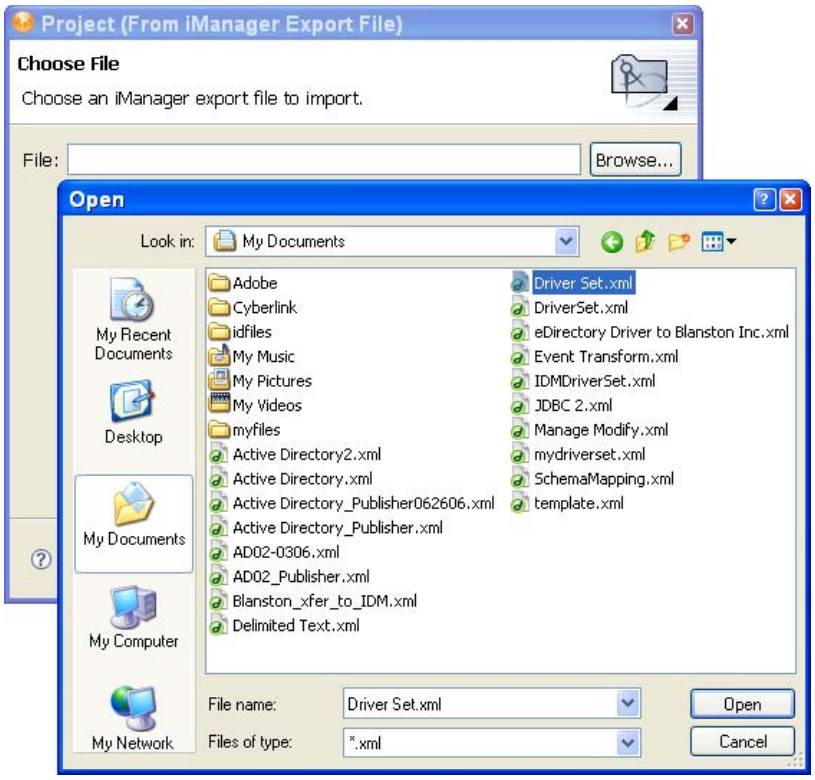
1 Click *File* > *Import* > *Designer for Identity Manager* > *Project (From iManager Export File)*.

or

Click *Project (From iManager Export File)* from the No Projects Available view, then skip to Step 3.



2 Type a project name. Use the default `designer_workspace` directory for the project, or type or browse to the directory where you want to import the project. Click *Next*.



3 Select the directory where the exported project is saved with the .xml extension, click *Open*, then click *Finish*.

4 When importing a driver set or driver into a project, you are asked if you want to validate the values within the drivers you are importing. If you do not want to validate the driver configuration at this time, click *No*.

   Otherwise, click *Yes* and continue importing the project.

   You can only import the driver functionality that you saved to the .xml file. This file does not contain default driver configurations unless that is what you have saved.

5 Fill out the information requested in all of the Import Information Requested windows that you see for each driver in the driver set, or for each driver selected. The Import Information Requested windows contain different driver information from each selected driver.

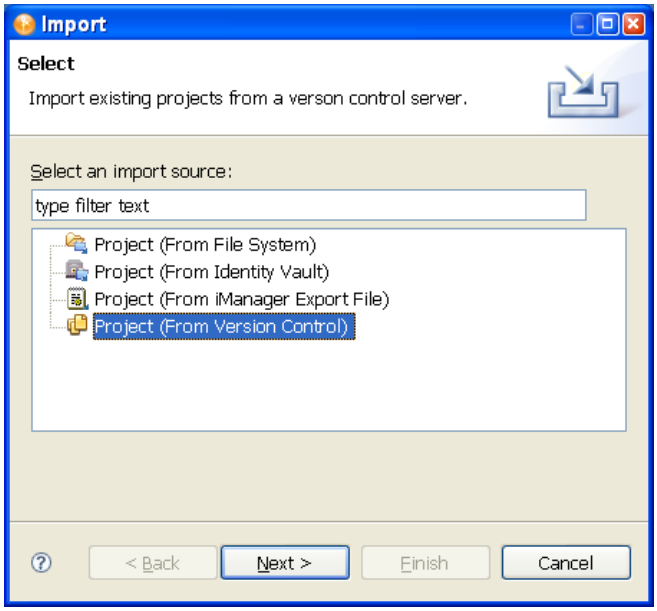
6 Click *Next* or *Finish* (depending on the number of pages presented).

7 Click *OK* to close the Import Configuration Results window.

11.1.4 Importing a Project from a Version Control Server

The Import dialog box lists projects and enables you to select projects that you want to import. There are a number of ways to access the Import dialog box in order to import projects from a version control server, and this example covers one of those methods.

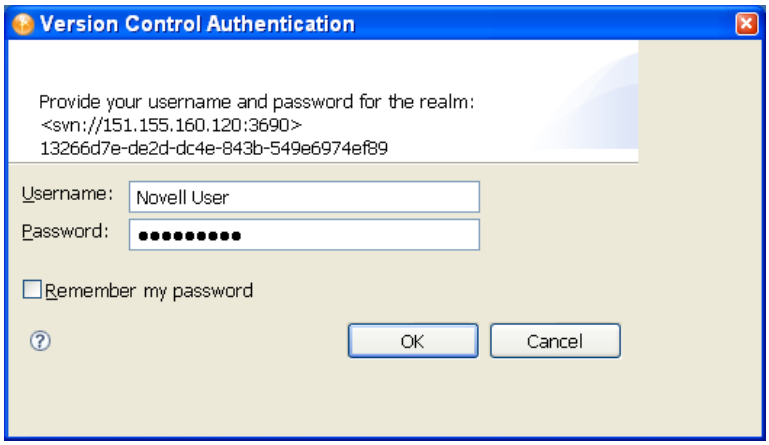
Figure 11-1 *The Import Wizard*



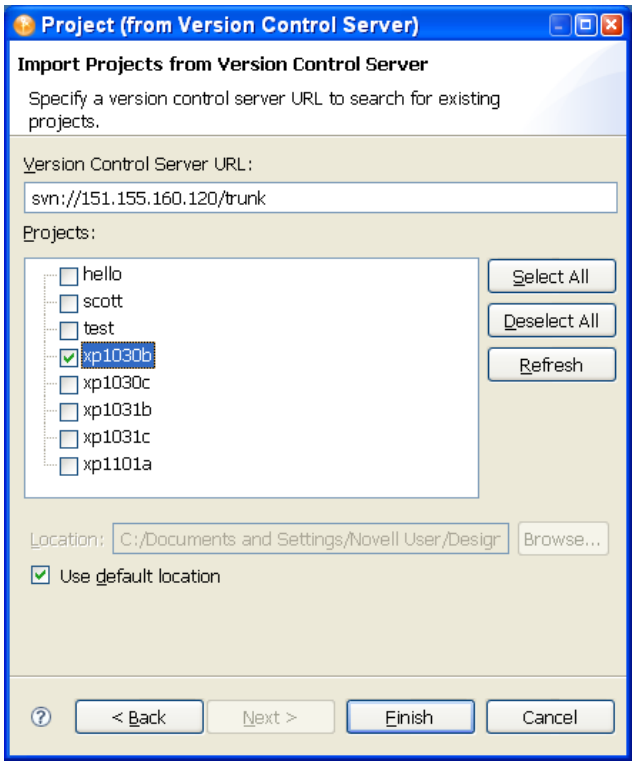
1. Select *File* > *Import*.
   or
   If no projects are available, select *Import from version control* from the Project view.
2. Click *Project (From Version Control)* > *Next*.
3. Type a URL in the *Version Control Server URL* field, then press *Enter*. For example:

   ```
   https://sun.provo.novell.com/svn
   ```

   ```
   svn://123.123.131.120/trunk
   ```
4. Provide authentication to the Subversion* server if required. Depending on the type of security you have set up, you might need to supply SSH authentication, SSL client certificate authentication, or basic HTTP authentication.



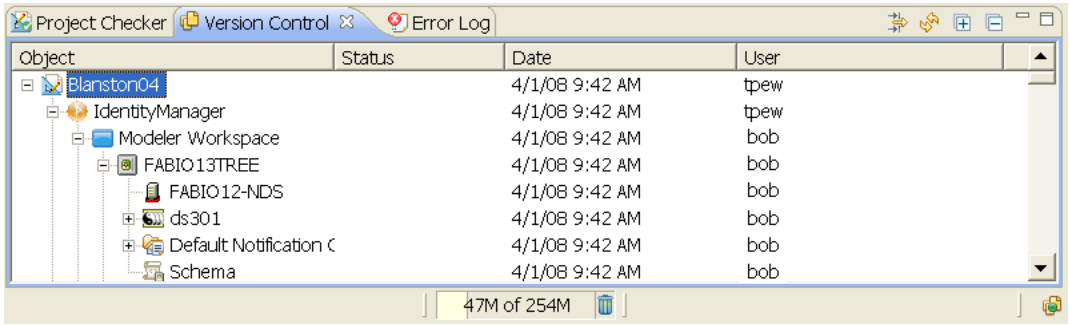
5. Select a project or projects.



Version control searches for projects three levels deep from the directory specified in the *Version Control Server URL:* entry.

6 Click *Finish*. At the Version Control page that shows you the version control server status, click *OK*.

The projects are imported into Designer and are added to the Project view and the Version Control view.



11.2 Importing a Library, a Driver Set, or Driver from the Identity Vault

To import an eDirectory object, you must have access to the eDirectory tree that is associated with the Identity Vault.

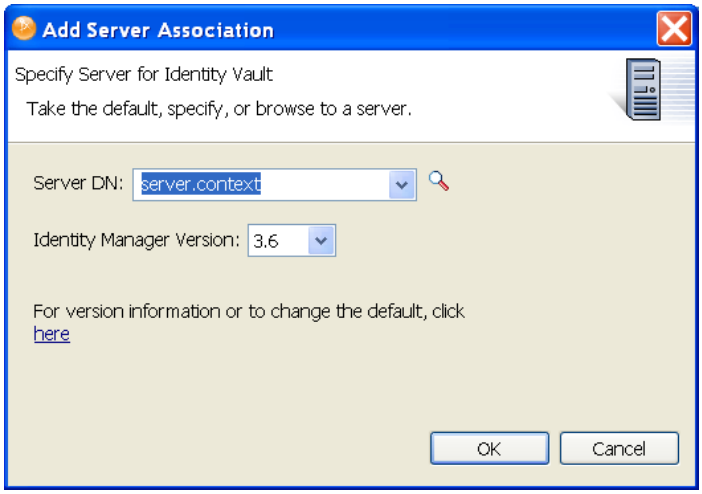
- Section 11.2.1, “Associating a Server to the Identity Vault,” on page 237
- Section 11.2.2, “Importing a Library from the Identity Vault,” on page 238

- Section 11.2.3, “Importing a Driver Set from the Identity Vault,” on page 239
- Section 11.2.4, “Importing a Driver from the Identity Vault,” on page 242

11.2.1 Associating a Server to the Identity Vault

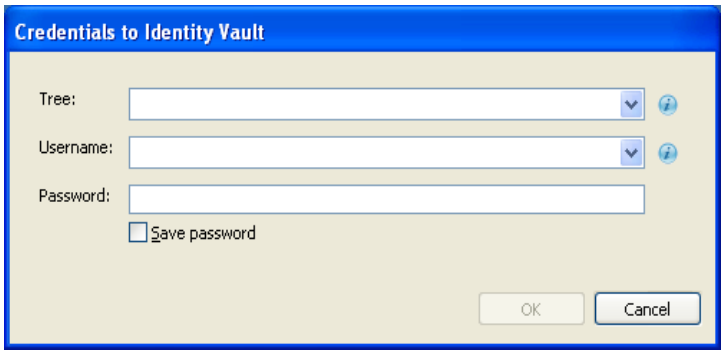
When you add a new Identity Vault to a project, you see the Add Server Association window, where you can accept the default server, specify a server, or browse to a server. The import and deploy features use the server association for later identification. To do this:

1. In the Modeler view, drag an Identity Vault icon from the Palette to the Modeler view to bring up the Add Server Association window.



   Type the server’s DN context in the *Server DN* field, or click *Browse*.

2. If you select *Browse*, fill in the appropriate host name, user name, and password in the Credentials to Identity Vault window. Click *OK*.



3. In the Browse for Server Object window, select the server you want to associate with this driver set and click *OK*.

   In the Add Server Association window, you also see the Identity Manager version displayed. This is important when importing and exporting driver sets and drivers, because you must match driver sets and drivers to the correct Identity Manager version.

4 Click the *For version information or to change the default, click here* entry for more information.

5 Click *OK* to close the Add Server Association window and add an Identity Vault to your Modeler view.

11.2.2 Importing a Library from the Identity Vault

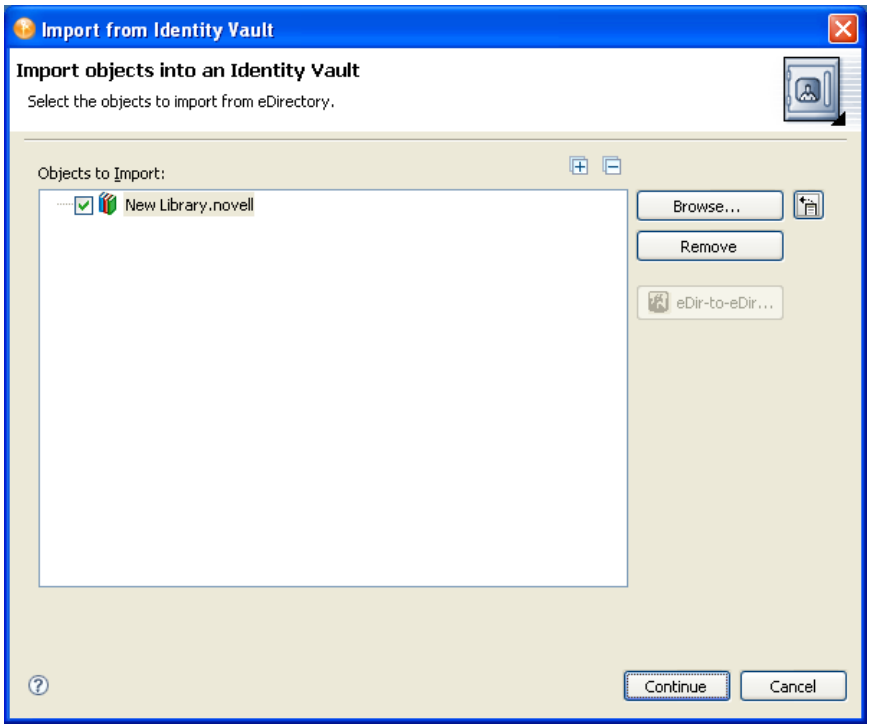
1 Right-click the Identity Vault in the Modeler view, then click *Live* > *Import*.

2 (Conditional) If you have not yet provided authentication information, specify it now. In the Identity Vault Credentials window, fill in the host name, the user name and password information, then click *OK*.

Use the drop-down lists if they apply to your connection and user information. The *Save Password* option allows Designer to keep password information for future connections to this Identity Vault. Otherwise, you will see the Identity Vault Credentials page the next time you open Designer.

3 On the Import from Identity Vault page, browse to the Library object by clicking the *Browse* button.

4 Select the Library object and click *OK*.



The library is added to the Import from Identity Vault page.

5 Click *Continue*, then click *Import* to import the library.

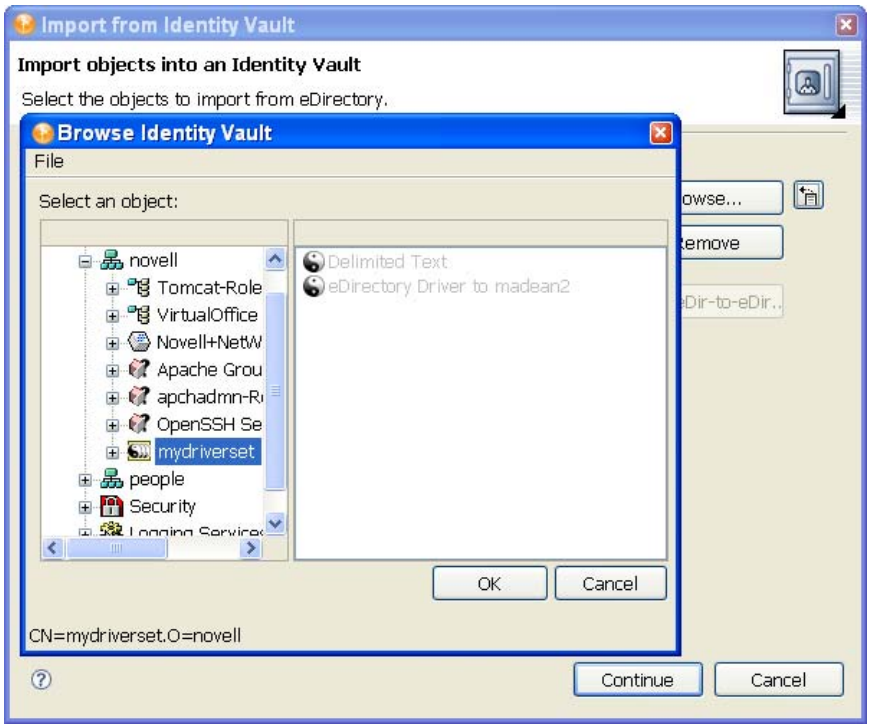
6 On the Import Results page, click *OK*.

11.2.3 Importing a Driver Set from the Identity Vault

To import an Identity Manager Driver Set object (and all contained drivers) into an Identity Vault object in the Modeler view:

1. Right-click the Identity Vault in the Modeler view, then click *Live* > *Import*.
2. (Conditional) If the Driver Set that comes with the Identity Vault creation is empty, you are asked if you want to remove the default Driver Set icon from the selected Identity Vault. Click *Yes*.
3. (Conditional) If you filled out the authentication information when you initially created an Identity Vault icon in the Modeler view, go to the Properties view under the Project view. Specify authentication credentials for the selected Identity Vault, then skip to Step 5.
4. (Conditional) If you have not yet provided authentication information, specify it now. In the Identity Vault Credentials window, fill in the host name, the user name and password information, then click *OK*.

   Use the drop-down lists if they apply to your connection and user information. The *Save Password* option allows Designer to keep password information for future connections to this Identity Vault. Otherwise, you will see the Identity Vault Credentials window the next time you open Designer.
5. In the Import from Identity Vault window, browse to the driver set by clicking the *Browse* button.



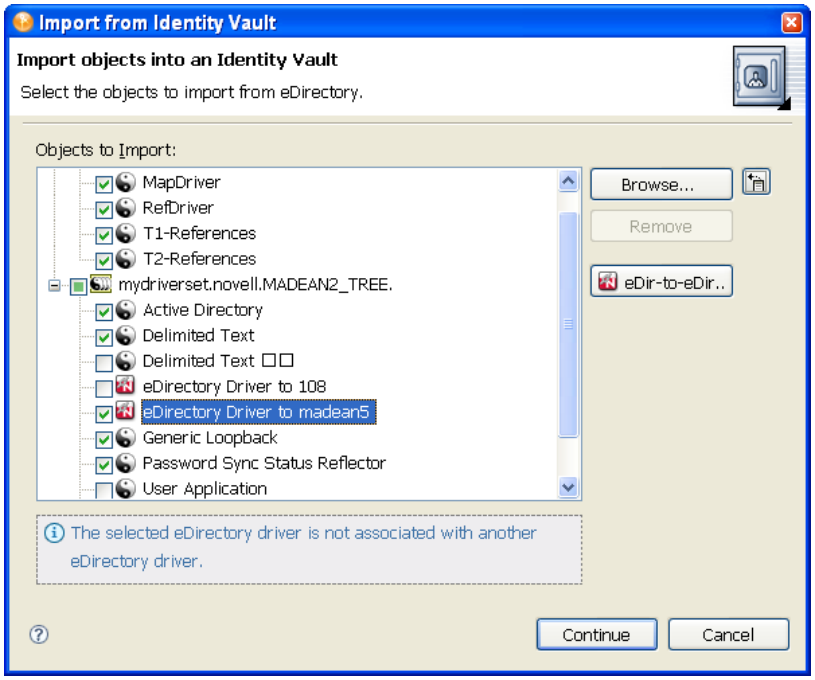
6. Select the driver set you want to import, click *OK* to place the driver set in the Objects to Import list in the Import Driver Set from eDirectory window. You can then deselect the drivers you do not want to import by deselecting the box next to the driver name. If you chose the wrong driver set, select the driver set and click *Remove*. Otherwise, click *Continue*.

You can import multiple driver sets during the import operation. Just browse to the various objects that you want to import and add them to the *Objects to Import* list.

Driver sets that are not associated with a server have a red minus in the lower right portion of the driver set icon. These driver sets need a server association in order to be deployed. An error displays if the application can't authenticate to the eDirectory tree you have selected.

7 (Conditional) If you are importing one or more eDirectory drivers, select the eDirectory driver in the Objects to Import window, then click the *eDir-to-eDir* button.

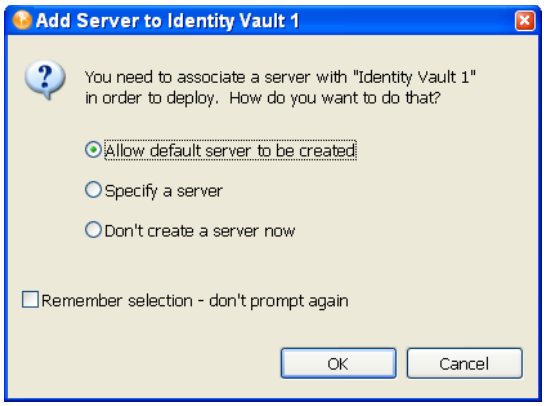
8 (Conditional.) If you fill out the user name and password and click *Continue* to import both eDirectory drivers, you then see a Browse Identity Vault window where you select the corresponding eDirectory driver. Select the driver and click *OK*.



You are returned to the Import Driver Sets from eDirectory window, where you can select or deselect the drivers, allowing you to choose only the drivers in a driver set that you want to import.

9 Click *Continue*. This brings up the Import Summary window, where you can see all of the Driver Set objects that are being imported into Designer. This summary uses the same format as the Compare window (see Section 11.5, "Using the Compare Feature When Importing," on page 259 for further information). Click *Import* to continue.

10 (Optional) As the import operation progresses, you are asked to associate a server with the Identity Vault. Select the option that best suits your needs.

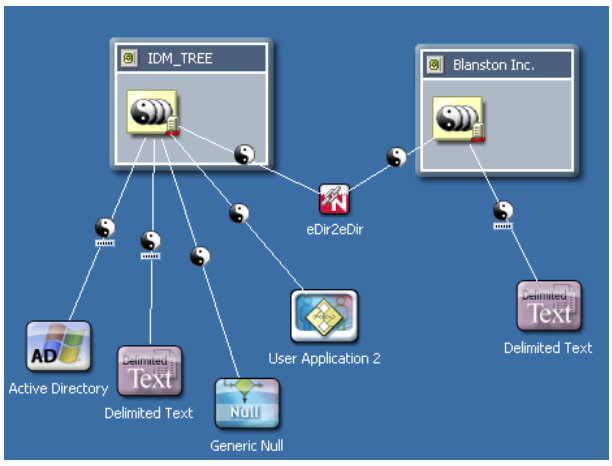


- **Allow default server to be created:** Creates a dummy server with global configuration values and other elements that are associated with this project until you specify an actual server for the project. Make sure you have designated a correct Identity Vault server when you deploy the driver set.
- **Specify a server:** Brings up the credentials screen, allowing you to designate a host server, a user name, and password for the Identity Vault server for this project.
- **Don't create a server now:** Skips all associations for this project. You need to fill in the host information before you deploy this driver set.
- **Remember selection - don't prompt again:** Continues to use whatever server option you chose the next time Designer needs to associate a server with an Identity Vault.

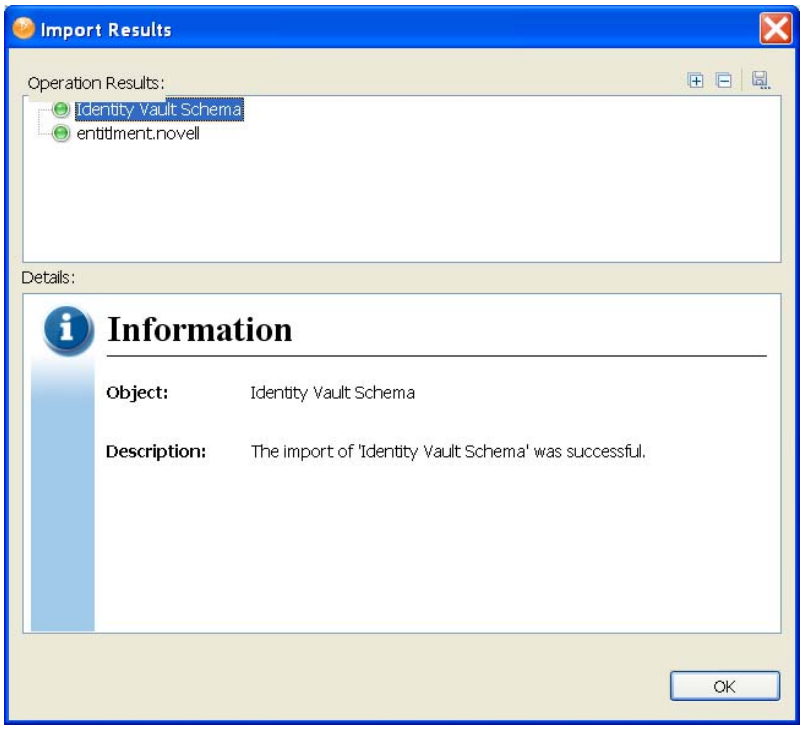
11 After you decide your plan of action and select the option you want, click *OK* to continue the import procedure.

12 Click *Finish*.

If you selected in Step 7 to connect eDirectory drivers, you can view the complete data flow between the two eDirectory drivers, as well as the other drivers you selected.



When the driver set imports, you see the Import Results window, showing you if there were any problems with the import procedure.



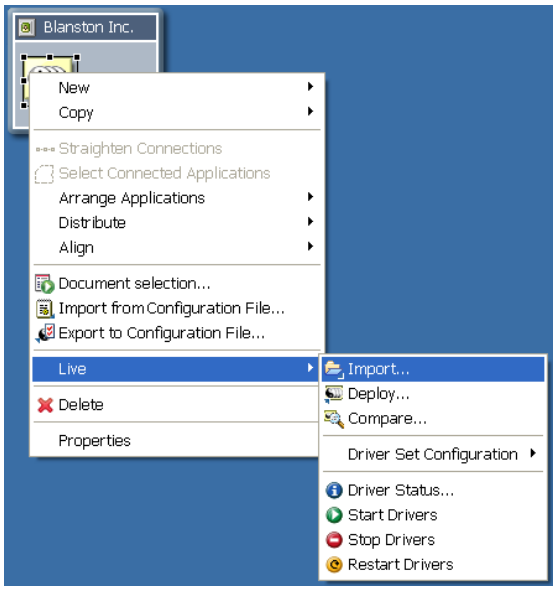
Errors during the import procedure are displayed with a red icon, and you see an Error description that is related to the operation results. If you have multiple errors, selecting an error displays the error's description in the *Details* > *Description* field. See "Error Messages and Solutions" on page 501 for further information.

13 Click *OK* to finish the import process.

11.2.4 Importing a Driver from the Identity Vault

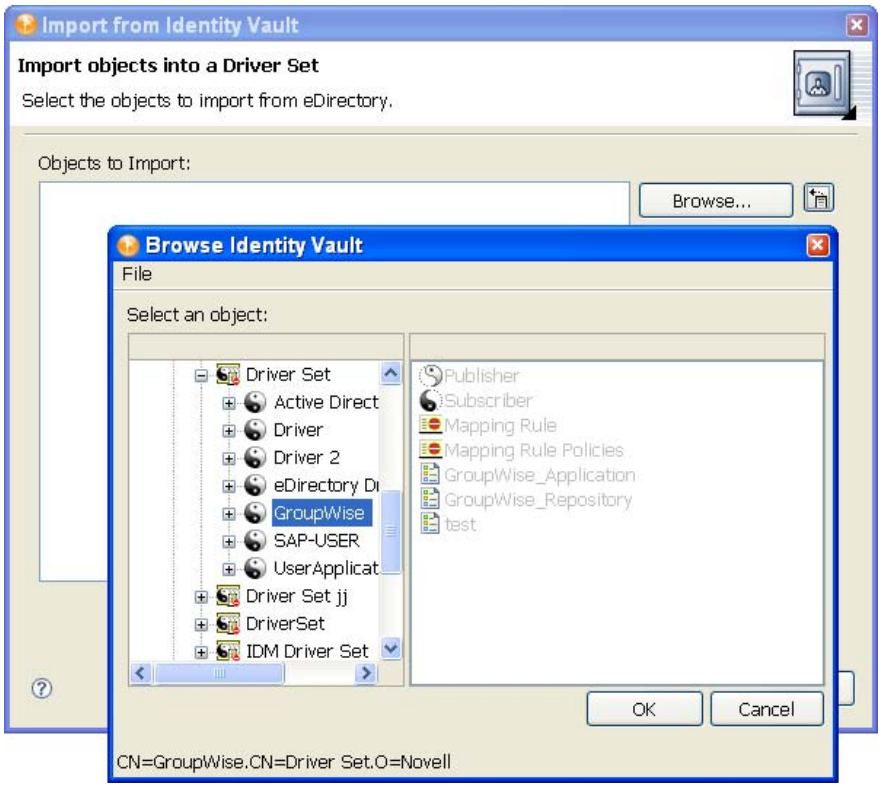
To import an Identity Manager Driver object (and all contained channels and policies) into a driver set:

1. Select an Identity Vault in the Modeler view. If you have added a new Identity Vault to a project, see Section 11.2.1, "Associating a Server to the Identity Vault," on page 237 first, then return to Step 2.
2. Verify that the authentication credentials in the Properties view for the Identity Vault are correct.
3. Right-click a DriverSet object within the Identity Vault, then select *Live* > *Import*.



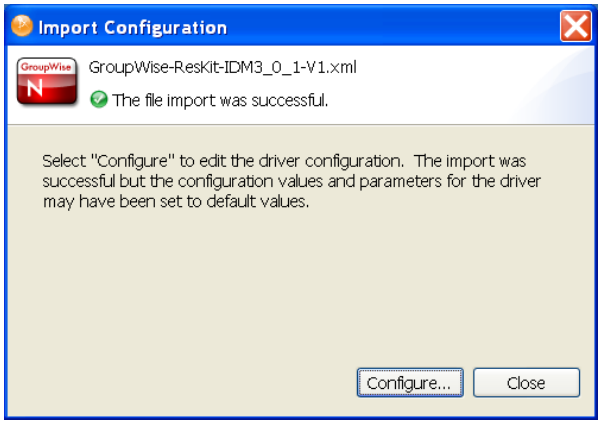
4 (Conditional) If the Identity Vault is not authenticated to the eDirectory tree, you see the Identity Vault Credentials window asking for the hostname, username and password. Provide the information, then click *Next*.

5 In the Import from Identity Vault window, click *Browse* to select a Driver object from the Identity Vault.



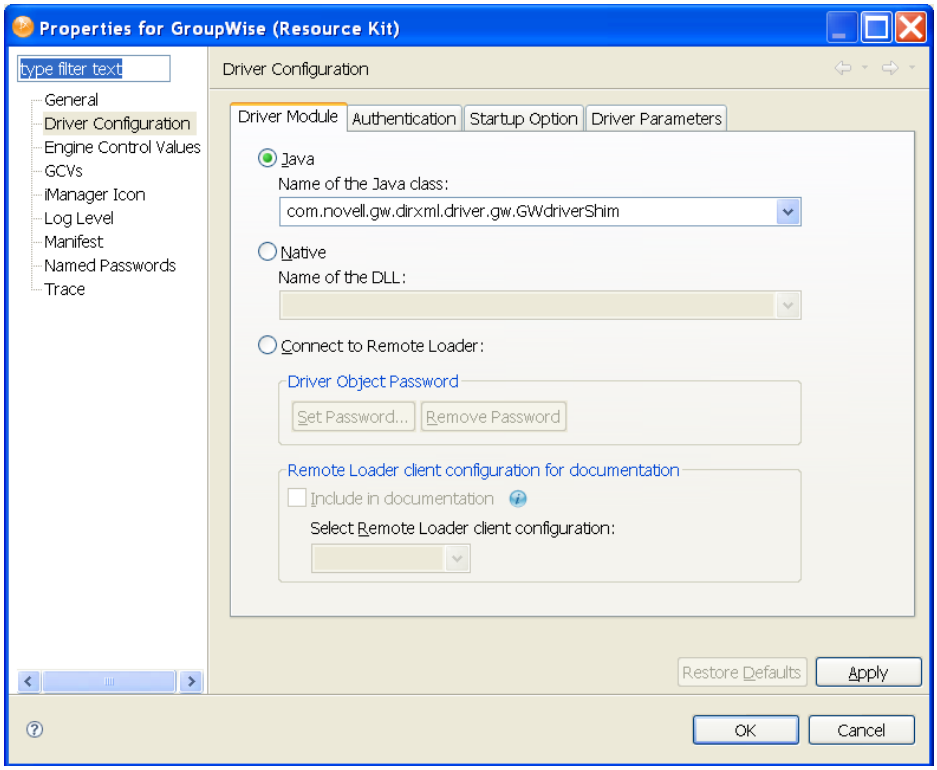
6 Click *OK* to place the driver in the Import from Identity Vault window, then click *Continue* to install the driver and bring up the Import Configuration window.

7 In the Import Configuration window, select *Configure* to edit the driver configuration, or select *Close* to close the Import Configuration window. Most drivers cannot run with default values. You need to modify the driver configuration values and parameters so the drivers can work properly in your network environment.



NOTE: You also see the Import Configuration window when you drag an application from the Palette to a driver set in the Modeler view.

8 When you select *Configure*, the driver's Property page with the *Driver Configuration* option is displayed. Fill out the required values and parameters that are necessary to have the driver run in your network environment.



The two required options for every driver are *Driver Configuration* and *GCVs*. However, because each driver contains different values and parameters, you need to consult the driver manual for specific values. Go to the Identity Manager Drivers web site (http://www.novell.com/documentation/idm36drivers/index.html), then select the manual for the driver you are configuring.

9 (Conditional) If you are importing one or more eDirectory drivers, Novell recommends that you connect to those eDirectory drivers during the import process. Select the eDirectory driver in the Objects to Import window, then click the *eDir-to-eDir* button.

10 (Conditional) Fill out the user name and password for the other eDirectory tree and select *Continue* to import both eDirectory drivers.

11 (Conditional) In the Browse Identity Vault window, select the corresponding eDirectory driver, then select the driver and click *OK*. You are returned to the Import Drivers from eDirectory window.

When the driver imports, you see the Import Results window showing you if there were any problems with the import procedure.

Errors during the import procedure are displayed with a red icon, and you see an Error description that is related to the operation results. If you have multiple errors, selecting an error displays the error's description in the *Details* > *Description* field. See "Error Messages and Solutions" on page 501 for further information.

12 Click *OK* to finish the import process.

11.3 Importing a Driver Configuration File

You can import an Identity Manager driver configuration file into the selected driver set for a project by using the Import from Configuration File option, which imports an XML configuration file that can be a driver set, an individual driver, driver channels, or policies. If you import a driver configuration file that contains only a policy, Designer creates the framework for parent containment objects, such as a channel, a driver, or a driver set. Such parent containment objects do not contain attributes; they are only the framework of the channel, driver, or driver set from where the policy came.

These configuration files are in XML format, which you can save to a local, removable, or network directory through the Export to Configuration File option. After you select the driver configuration file to import, you are prompted to provide driver-specific information for setting up the driver and building policies. The exported `.xml` files are compatible with those used by the iManager driver configuration file plug-ins for Identity Manager 2.7. This allows you to export configuration files from Designer and import those files through iManager or through Designer's Import feature.

You can also use the *Import from Configuration File* option as a backup method. For example, if you are working on a group of policies in an Active Directory driver, you can use the *Export to Configuration File* to back up your work, adding dates to the exported filenames so you can select a particular driver configuration that you want to use as the basis for your Active Directory driver. If something goes wrong, you can re-import the base driver. You can also use this export/import feature to export and import just the policies you are working on.

Of course, a better method for backing up your projects is through the version control feature. See Chapter 19, "Version Control," on page 411.

You can import a configuration from a file in three ways:

- Section 11.3.1, “Importing an Identity Manager Project from the File System,” on page 246
- Section 11.3.2, “Importing a Driver Configuration from a File in the Modeler View,” on page 246
- Section 11.3.3, “Importing from a File through the Outline View,” on page 248

11.3.1 Importing an Identity Manager Project from the File System

The *Import an Identity Manager Project from File System* option allows you to import an Identity Manager project that has been saved to the file system through the *Export* > *File System* option. The project must have a valid `.project` file and accompanying file structure for the project to completely import. For information about importing a project, see Section 11.1.2, “Importing a Project from the File System,” on page 231.

11.3.2 Importing a Driver Configuration from a File in the Modeler View

You can import a previously exported configuration file or one of the sample `.xml` driver configurations that are included with Designer. To import a configuration file into a driver set:

1. Select an Identity Vault in the Modeler view.
2. Right-click a DriverSet object within the Identity Vault, then select *Import from Configuration File*.
3. In the Import a Driver Configuration File window, type the name of the configuration file, or browse to and select the `.xml` file.

   If you use the Browse feature, by default Designer takes you to the following directories:

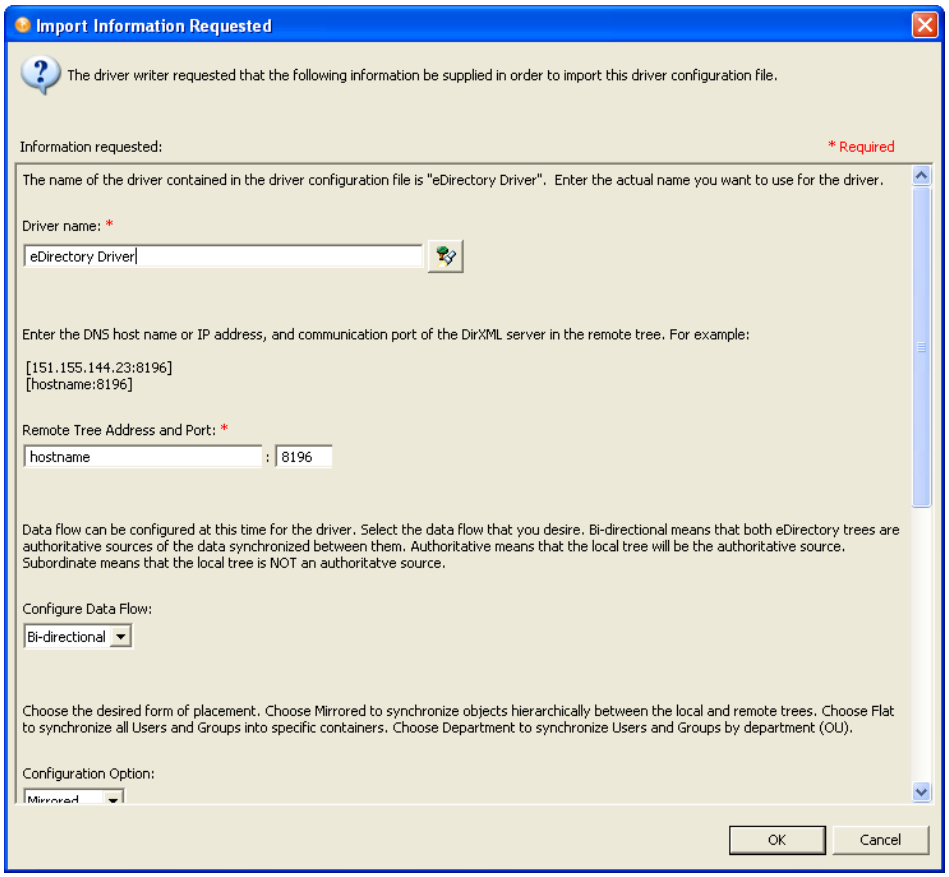
   - for Windows:

     ```
     C:\Program
     Files\Novell\Designer\eclipse\plugins\com.novell.idm_3.5.x\defs\drive
     r_configs\current\drivername
     ```

   - For Linux:

     ```
     /home/username/designer/eclipse/plugins/com.novell.idm_3.5.x/defs/
     driver_configs/current/drivername
     ```

   (The “x” after the idm_3.5 number refers to Designer build numbers). You can use one of the `.xml` configuration files in a selected directory or you can browse to a directory containing an exported configuration file.
4. Click *OK*.
5. Complete the import by filling out the Import Information Requested prompts for the driver configuration file.



- The template varies, depending on the driver configuration file selected and the state in which the file was saved. Saved files might only prompt for a driver name, but need other values in order to work in a network environment.

  NOTE: The pre-Identity Manager 3.6 Driver Configuration Wizard adds different policies to the driver, depending on which options you select when you initially import the driver. If you want to change an option that you did not initially choose, it is not possible, because the information is not included in the driver. You must delete the driver and create a new one through the Driver Configuration Wizard.

  WARNING: Do not use the Driver Configuration Wizard on the `.xml` file that you are importing. The Wizard brings up the Import Information Request forms, but these forms are pulled from the default driver that come with Designer and will overwrite the driver you are importing. Use this method only if you need to start over.

- Identity Manager 3.6 Driver Configuration Wizard adds all policies when the driver is imported, and are not lost if you did not select an option in the Import Information Request forms. If this is a driver configuration file that came with Identity Manager 3.6, you can change the driver's values through the Properties page.

  If the driver needs other values and parameters in addition to what appears on the Import Information Requested template, close the template, right-click the driver line in the Modeler view and select *Properties* > *Driver Configuration* and *GCVs* to fill out what you need. You may also need to fill out GCVs at the driver set level.

Since each driver contains different values and parameters, consult the driver manual that matches the installed driver at the Identity Manager Drivers web site (http://www.novell.com/documentation/idm36drivers/index.html). Then select the manual for the driver you are configuring.

6 Click *OK*, then click *OK* in the Import Configuration Results window.

7 You might have imported a single driver or a collection of drivers (a driver set). If you are importing a driver set, repeat Steps 4 and 5 for each driver in the driver set. When you are finished with each imported driver, click *OK* at the Import Configuration Results window.

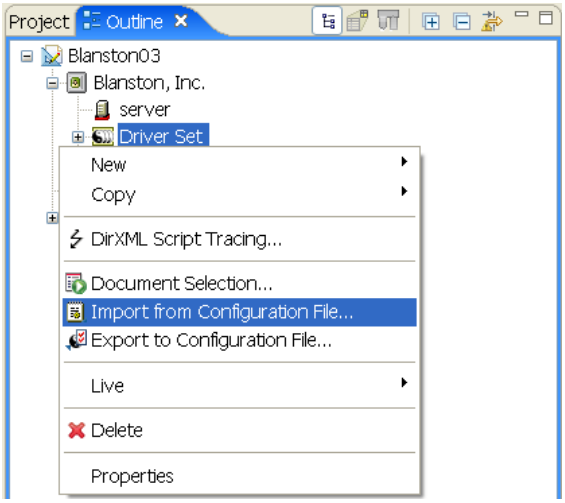
11.3.3 Importing from a File through the Outline View

You can use the Outline View to import driver sets, drivers, channels, and policy configuration files from the file system. The following example demonstrates how to import a driver, but the procedure also works for the other files.

1 Double-click the *System Model* icon under a project name in the Project view. This brings up the project model in the Modeler view.

2 Click the *Outline* tab.

3 Right-click the Driver Set object and select *Import from Configuration File*.



4 In the Import a Driver Configuration File window, type the name of the configuration file, or browse to and select the `.xml` file.

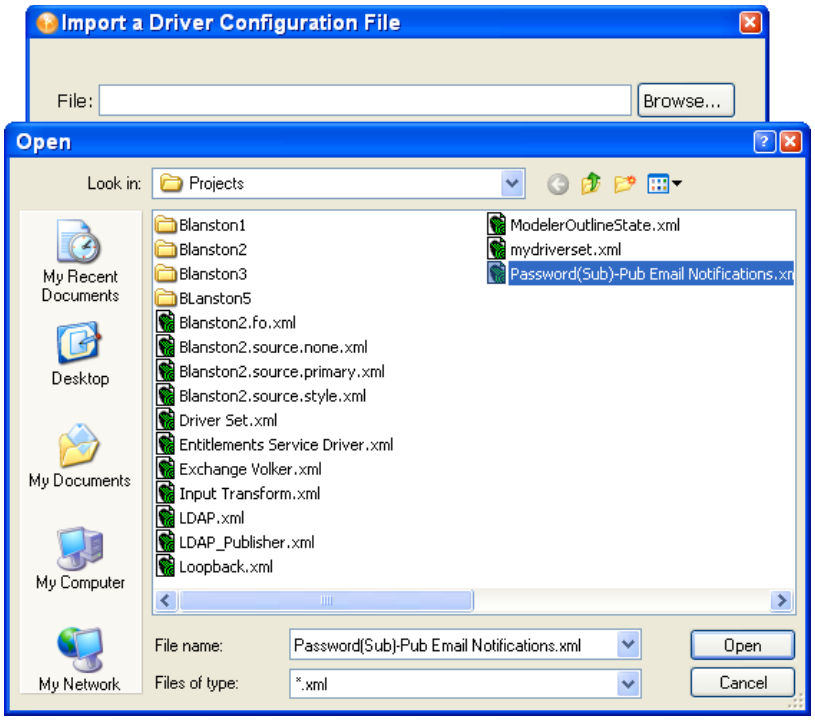
When a driver is exported, Designer uses the name of the driver set or driver object, to which you can add dates. In this example, it is an Active Directory driver that was exported June 26th and is now being imported.

5 Click *Open*, then click *OK* to import the configuration file.

To import a policy into a driver set:

1 In the Outline view, click the *Driver Set* icon, then click *Import from Configuration File*.

2 In the Import a Driver Configuration File window, browse to or specify the XML configuration file name.



3 Click *Open*, then click *Save* to import the selected policy.

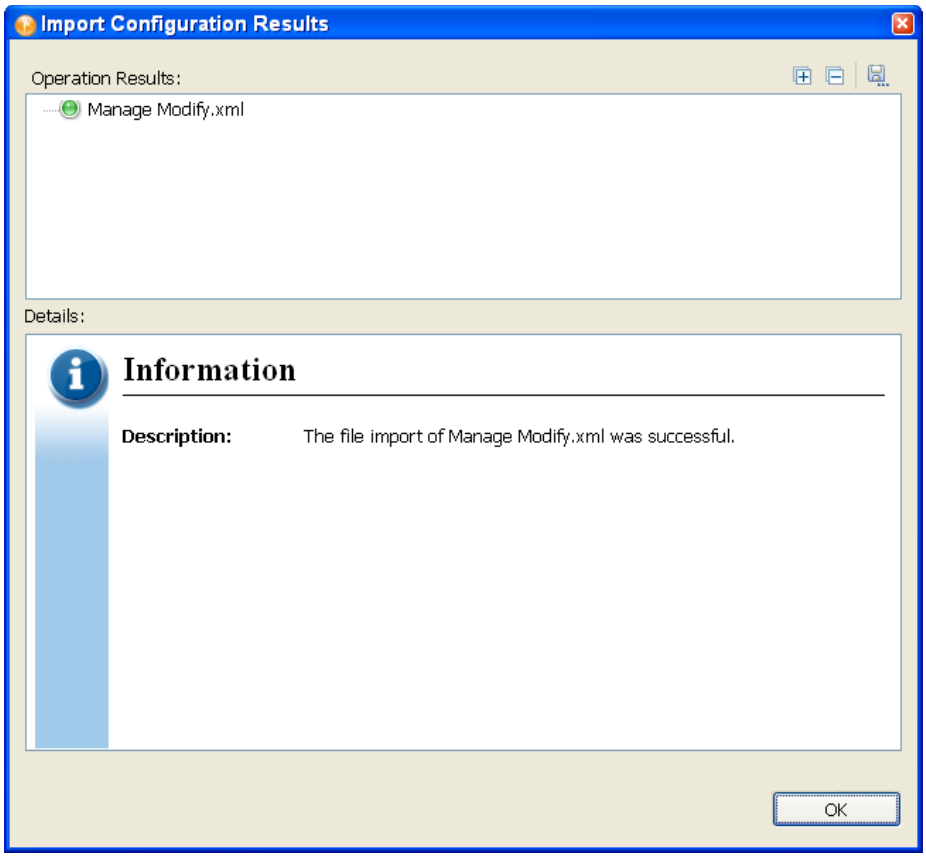
Each policy is saved to its own `.xml` file. By default, Designer uses the name of the policy or rule selected.

4 In the Perform Prompt Validation window, you are asked if you want to fill in required driver information. If you answer *Yes*, you see the Import Information Requested dialog box as described in Step 5 and you must provide values to all of the required fields. If you answer *No*, you still see the Import Information Requested dialog box, but it isn't necessary to fill in the required information.

5 Complete the import by filling out the Import Information forms for the driver configuration file as necessary. Policies are saved with a skeleton driver configuration structure, which designates where the policy resides. In this case, the driver already existed and the imported policy was initially written for that driver.

6 Click *OK*.

When the policy or rule is imported, you see the Import Configuration Results window, which indicates if there were any problems with the import procedure.



Errors during the import procedure are displayed with a red icon, and you see an Error description that is related to the operation results. If you have multiple errors, selecting an error displays the error's description in the *Details* > *Description* field. See "Error Messages and Solutions" on page 501 for further information.

7 Click *OK* to finish the import process and close the Import Configuration Results window.

11.4 Importing Channels, Policies, and Schema Items from the Identity Vault

A channel is a combination of rules and policies, and Designer allows you to import a channel instead of the entire driver. The Subscriber and Publisher channels describe the direction in which the information flows. The Subscriber channel takes the event from the Identity Vault and sends that event to whatever the receiving system is (application, database, CSV file, etc.) The Publisher channel takes the event from the application, database, CSV file, etc., and sends that event to the Identity Vault. The Subscriber and Publisher channels act independently; actions in one are not affected by what happens in the other.

- Section 11.4.1, "Importing a Channel," on page 251
- Section 11.4.2, "Importing a Policy," on page 254
- Section 11.4.3, "Importing a Schema," on page 256

11.4.1 Importing a Channel

To import an Identity Manager channel (a Subscriber channel or a Publisher channel) object (and all contained policies) into a driver:

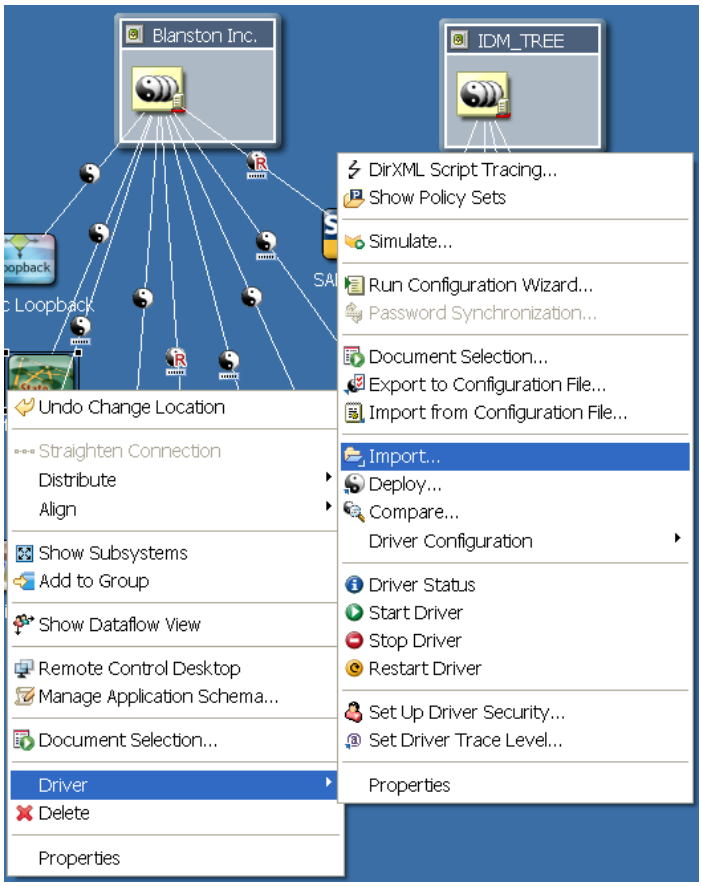
1. Select either a Driver object or an Application object in the Modeler view.

   The Driver object is represented by the line between the Identity Vault and the Application object and has a circle icon to represent it. The Application object connects to the Identity Vault through the Driver object.

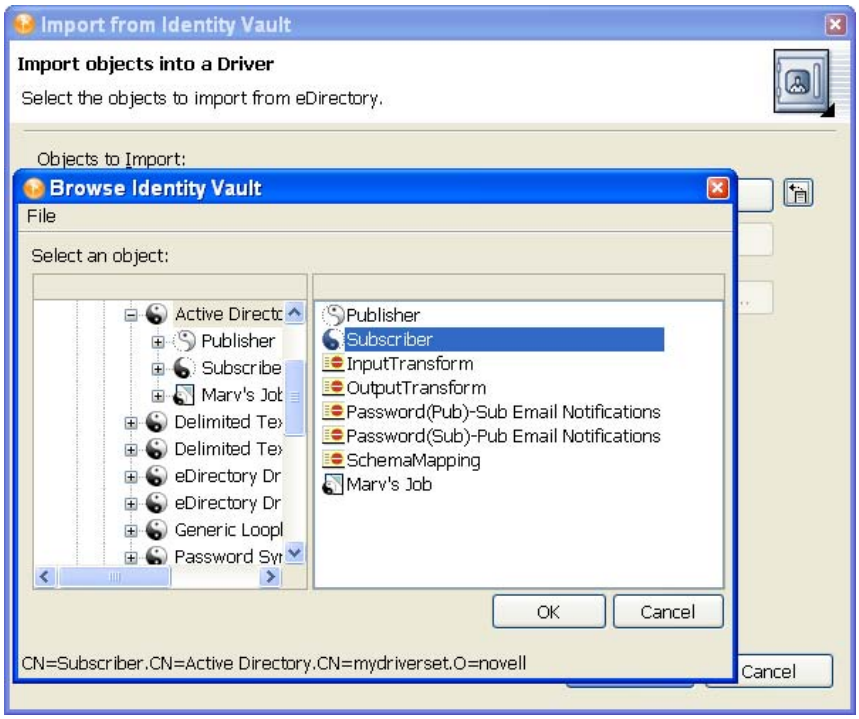
2. Right-click a Driver object, then click *Live* > *Import*.

   or

   Right-click an Application object and click *Driver* > *Import*.



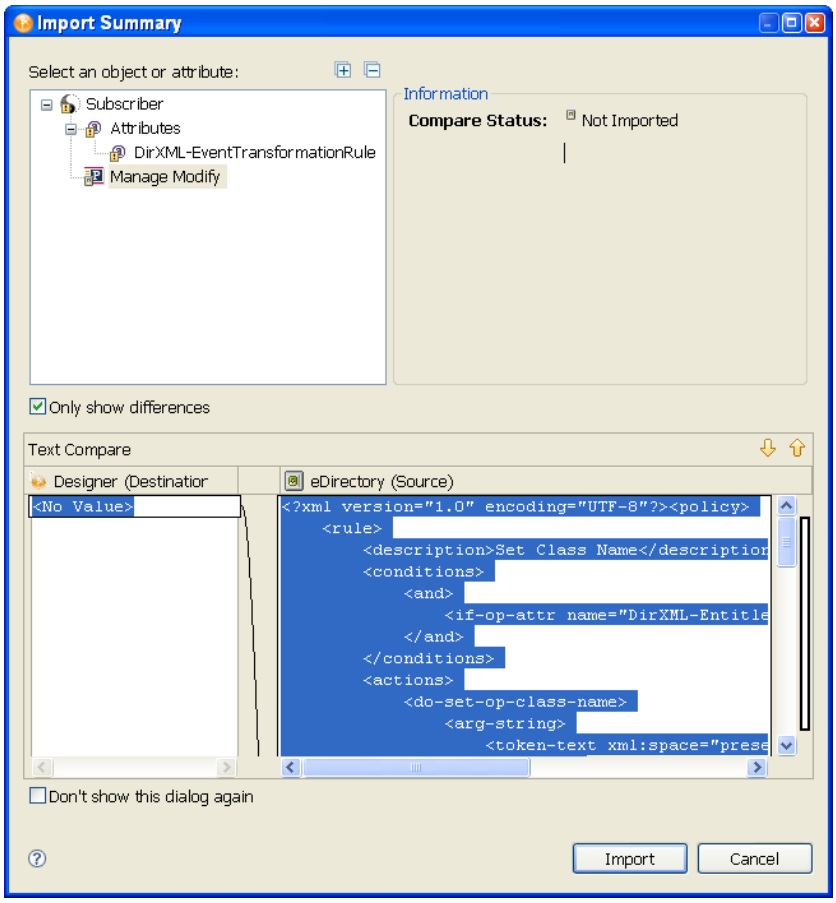
3. If Designer can't authenticate to the eDirectory tree specified in the Identity Vault, you see the Identity Vault Credentials window if you have not previously specified the authentication credentials or if you do not save the password. Fill in the appropriate information and click *OK*.
4. In the Import from Identity Vault window, browse to and select either a Publisher or a Subscriber Channel object from the eDirectory tree under the corresponding driver.



5 You can import more than one channel at a time; if you want to import both channels, select one channel, click *OK*, then browse to the next channel, select it, and click *OK*.

6 Click *Continue*.

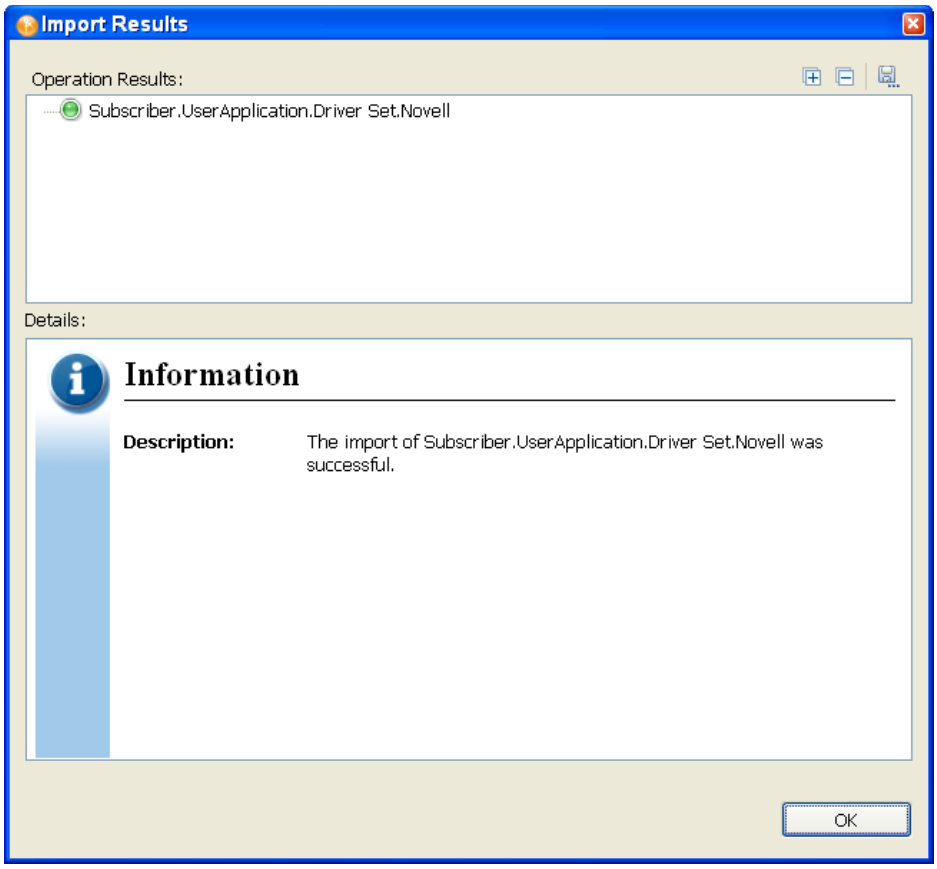
As the channel imports, you see the Import Summary window showing you the differences between eDirectory (the source of the import) and Designer (the destination).



You can click the different objects in the channel view differences between the two drivers. All channel information is overwritten by the import procedure; however, the rest of the driver is unaffected.

7 Click *Import*.

For more information on Compare, see Section 11.5, “Using the Compare Feature When Importing,” on page 259.



If there are any problems with the import procedure, they are displayed with a red icon in the Import Results window and you see an error description that is related to the operation results. If you have multiple errors, selecting an error displays the error's description in the *Details* > *Description* field. See "Identity Vault Configuration Errors" on page 502 for further information.

8 Click *OK* to finish the import process.

11.4.2 Importing a Policy

A policy is a collection of rules and arguments that allows you to transform the data that an application sends to and receives from eDirectory. You use policies to manipulate the data you receive from eDirectory or from the connected system so they can synchronize the information in their databases. Each driver connects to a different system, and policies tell the driver how to synchronize the data on that connected system to the Identity Vault.

You might use the Import feature for policies more than anything else. For example, you can set up a policy to allow users with the title "Manager" to be placed in a specific container, no matter which application the information is coming from, and you can place this information in multiple connected systems. However, because each application is different, you need to modify the arguments and rules within policies to reflect those differences. For more information about policies, see *Understanding Policies for Identity Manager 3.6* (http://www.novell.com/documentation/idm36/policy/data/bookinfo.html) and *Policies in Designer 3.5* (http://www.novell.com/documentation/idm36/policy_designer/data/bookinfo.html).

To import an eDirectory Policy object (for example, a rule or a style sheet) into a driver or channel (Subscriber or Publisher):

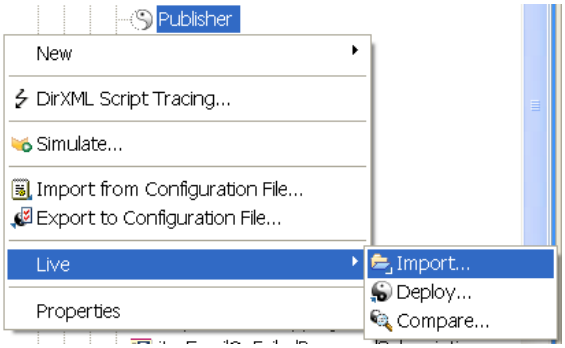
1. Select a driver in the Modeler view.

   or

   Click the *Outline* tab and select a Driver or Channel object from the Outline view.

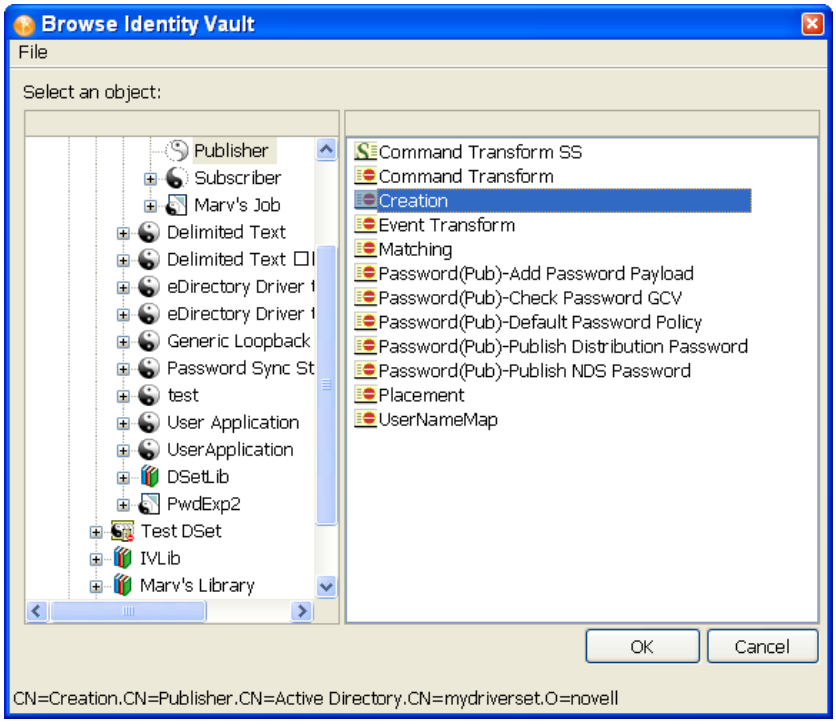
2. Verify that the authentication credentials in the Properties view for the selected Identity Vault are correct.

3. Right-click the Driver or Channel object, then click *Live* > *Import*.



4. If the application can't authenticate to the eDirectory tree, you see the Identity Vault Credentials window asking for the hostname, username, and password if you have not previously specified the authentication credentials or if you do not save the password. Fill in the appropriate information and click *OK*.

5. In the Import from Identity Vault window, click *Browse*, then select a policy object from the channel you specified when you started the import process.



Policies are found under either the Publisher or Subscriber channel of a selected driver or under the driver itself. Be sure to match the proper policy to the proper channel or driver object.

6 Click *OK*, then click *Continue* to import the policy.

You see the Import Summary window showing you the differences between eDirectory (the source of the import) and Designer (the destination). You can click the different objects in the policy to see what is different between the two policies. All selected policy information is overwritten by the import procedure; however, the rest of the driver is unaffected.

7 Click *Import*.

If the importing policy contains the same values as the policy in Designer, you are not allowed to import the policy. (See Section 11.5, “Using the Compare Feature When Importing,” on page 259 for more information on the Compare feature.)

Clicking *Import* brings up the Import Results window. If there are any problems with the import procedure, they are displayed with a red icon, and you see an Error description that is related to the operation results. If you have multiple errors, selecting the different errors displays the error’s description in the *Details* > *Description* field. See Section 21.5.3, “Error Messages and Solutions,” on page 501 for further information.

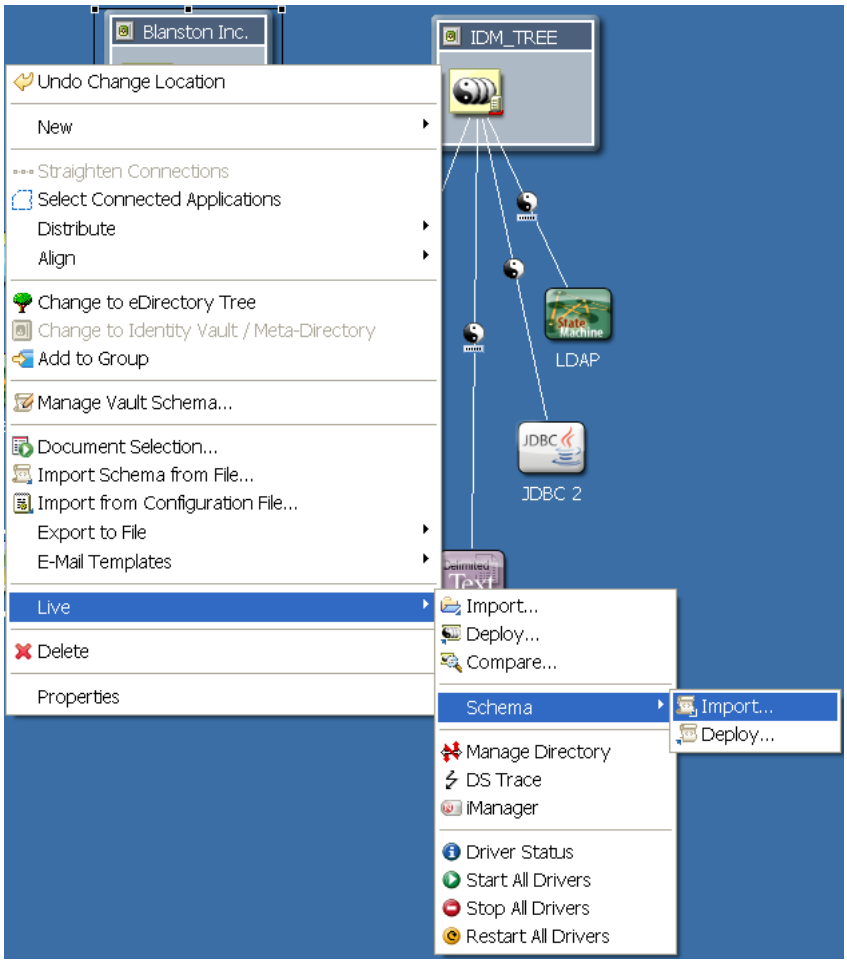
8 Click *OK* to finish the import process.

For policy design, see the Policy Builder and Policy Management Help topics within the Designer utility. Also, see *Understanding Policies for Identity Manager 3.6* (http://www.novell.com/documentation/idm36/policy/data/bookinfo.html) and *Policies in Designer 3.5* (http://www.novell.com/documentation/idm36/policy_designer/data/bookinfo.html).

11.4.3 Importing a Schema

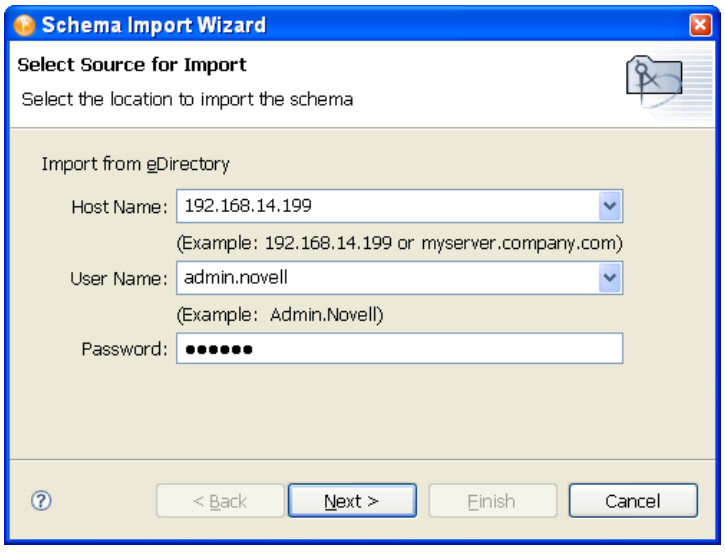
You can import a schema from the Identity Vault or from a `.sch` file into your project. When you import a schema, you can select the whole Identity Vault schema (not recommended) or just the schema differences between the Identity Vault and your project.

1 Bring up the project in Designer’s Modeler view. Right-click the Identity Vault and select *Live* > *Schema* > *Import*.



2 On the Select Source for Import page, select *Import from eDirectory* if you can connect to an actual Identity Vault.

NOTE: The specified user must have administrative rights to the schema.



3 In the Import from eDirectory section, specify the hostname, username and password connection information.

The *Host Name* and *User Name* entries have drop-down menus storing the last information you typed into these fields, which you can use for filling in these entries.

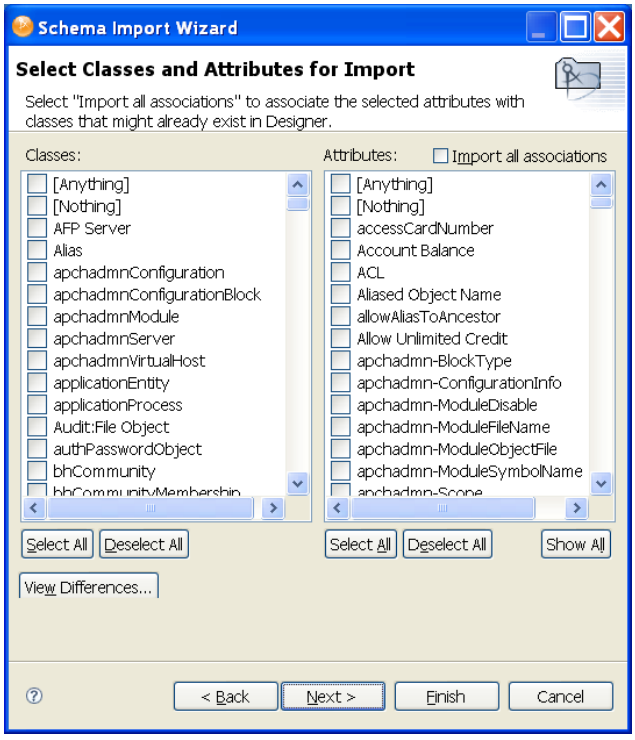
4 Click *Next*.

5 Decide which classes and attributes to import.

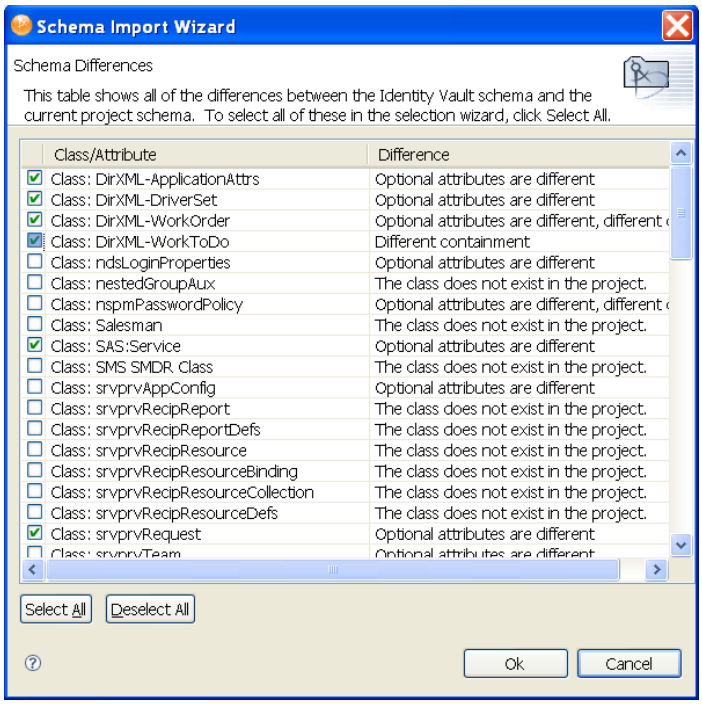
On the Select Classes and Attributes for Import page, you can select all of Identity Vault's schema, including classes and attributes. However, this can create very large documents when you document the project (600 pages or more).

6 If you want to import all the classes and attributes, click *Select All*, click *Finish*, then skip to Step 8.

Select only the classes and attributes that you want to import. If you only want to import the schema differences between the live Identity Vault and the Identity Vault in your project, click *View Differences*, then continue with Step 8.



7 On the Schema Differences page, you see the schema differences between the live Identity Vault and the Identity Vault in your project. Click *Select All* if you only want to import schema differences. Otherwise, click *Cancel*.



8 Selecting *Select All* > *OK* brings you back to the Select Classes and Attributes for Import page with the schema differences now selected under the Classes and Attributes headings. If you select any classes from the Schema Differences page, the *Import all associations* box is selected. Leave it selected, as it enables you to associate the selected attributes with the classes that might already exist in Designer. Click *Finish*.

   If you selected *Cancel* on the Schema Differences page, make your schema selections on the Select Classes and Attributes for Import page, select the *Import all associations* box (recommended), and click *Finish*.

9 (Optional) Click *Next* if you want to see the Import Summary page to see the classes and attributes that you are importing. Then click *Finish*.

10 On the Import Messages page of the Schema Import Wizard, click *OK*.

   If you want to save the differences to a log file, click *Save to Log*. This brings up the Save As dialog box, where you can choose a filename and directory to store the file in.

11 Click *Save*, then click *OK*.

11.5 Using the Compare Feature When Importing

Designer's Compare feature allows you to see differences between the driver sets, drivers, channels, and policies that are stored in projects and those that are running in deployed systems, and reconcile any differences to either Designer or Identity Vault. Previous versions of Designer only provided conflict resolution when importing a Driver. While importing, you could select which policies of the driver you wanted to update, but you could not view any differences between existing and new values.

Designer provides conflict resolution on an object-by-object basis and allows you to view the differences between existing and new values when importing and deploying driver sets, drivers, channels and policies. For example, before importing a driver object in Designer to a driver object

that already exists in the Identity Vault, you can run Compare. Compare shows whether the driver objects are equal (no action is necessary) or unequal. If unequal, you can choose not to reconcile the driver objects, choose to update the driver object in Designer, or choose to update the driver object in the Identity Vault.

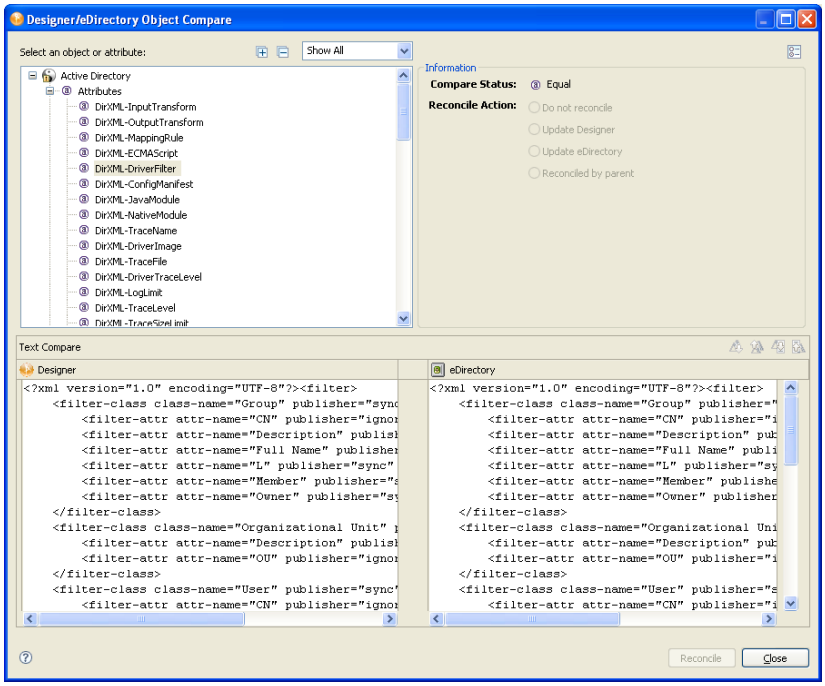
You can run the Compare feature at any time. If you choose to reconcile the differences between drivers objects in Designer and eDirectory while in Compare, you won't need to run Import or Deploy.

- Section 11.5.1, "Using Compare When Importing a Driver Object," on page 260
- Section 11.5.2, "Using Compare on a Channel Object," on page 263
- Section 11.5.3, "Using Compare On a Policy," on page 263
- Section 11.5.4, "Matching Attributes with Designer Properties," on page 263

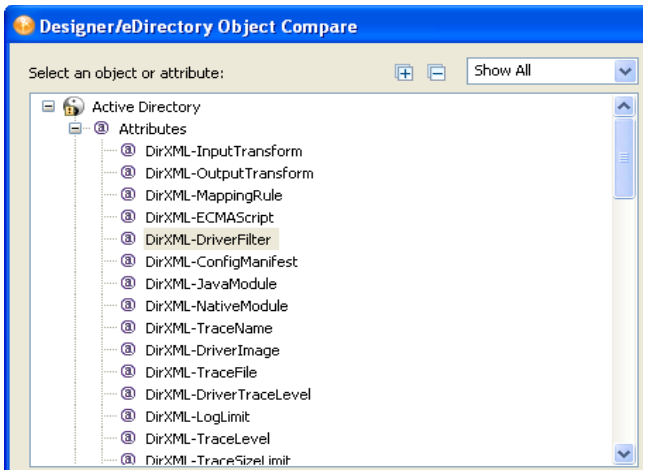
11.5.1 Using Compare When Importing a Driver Object

Use this procedure if you want to import a Driver object from the Identity Vault and the same driver already exists in Designer.

1 Right-click the driver object in either the Modeler view or in the Outline view, then click *Live* > *Compare* to bring up the Designer/eDirectory Object Compare window.



Under the *Select an object or attribute*, you see the selected object with the differences between Designer's and eDirectory's driver object. You can select the attributes and child objects to see the actual differences displayed in the *Text Compare* area.

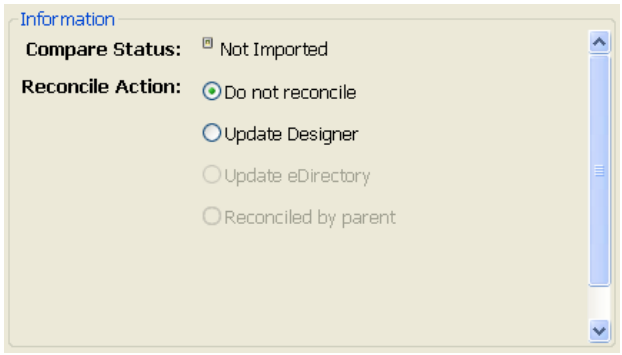


The plus icon at the right side of the *Select an object or attribute* allows you to expand all elements in the parent object, and the minus icon collapses all of the elements. The "?" button in the bottom left portion of the window displays the Summary/Compare dialog box help.

Server-specific attributes are attributes that have a value for each server that is associated with a driver set. These attributes are displayed in the *Attributes* list with the server name in parentheses to the right of the attribute name.

2 By default, the Compare window only displays values that are different between Identity Vault and Designer. To view all of the object values, select *Show All* from the drop down menu.

Values that are equal are shown as *Equal* on the *Compare Status* line under *Information*.



The overlay image displayed in the *Compare Status* entry identifies objects or attributes that need reconciliation. The following table describes what you see in the *Compare Status* line and the overlays that you can see:

Compare Status	Description
Equal	The selected attribute's value or all attributes of the selected object are the same in eDirectory and Designer.
Unequal	The value of the selected attribute, or one or more attributes of the selected object, are different in eDirectory and Designer.

Compare Status	Description
Not Deployed	The selected object or the object containing the selected attribute is not deployed to eDirectory.
Not Imported	The selected object or object containing the selected attribute does not exist in Designer.
Unknown	The selected object or object containing the selected attribute cannot be compared, such as a password.
Deleted	Designer tracks objects that are deployed, then deleted from the Designer project.

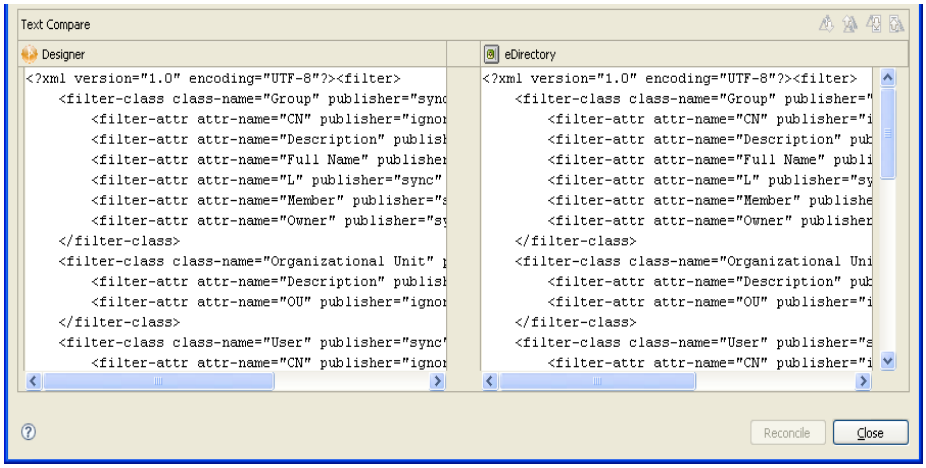
You can also see an *Attribute Note* if you select an attribute.

3 In the *Information* portion of the Compare window, select how you want to reconcile the differences between the Source and Destination. If *Compare Status* shows *Unequal*, you have three choices:

- **Do not reconcile:** To do nothing, keep the default value of *Do Not Reconcile*.
- **Update Designer:** To update the driver in Designer so that it contains the same information as the driver in the Identity Vault, select *Update Designer*.
- **Update eDirectory:** To update the driver in eDirectory to reflect the changes you have just made to the driver in Designer, select *Update eDirectory*.

If you select the parent object to perform the update, then all of the child objects under the parent reflect that choice and you see the *Reconciled By Parent* button selected. If you do not choose a parent object, you can reconcile each child object individually.

4 View the differences displayed in the Text Compare area.



The Text Compare values displayed in the bottom portion of the Designer/eDirectory Object Compare window vary, depending on the object being compared. For instance, Compare shows you changes down to the policy level. The Text Compare dialog box uses the Eclipse Compare editor to compare attributes that contain XML data, such as policy data, driver filters, or configuration data. The differences in the code are highlighted in blue.

5 After you view the differences, click *Reconcile* to perform the reconciliation actions for each object in the tree, or click *Close* to close the Designer/eDirectory Object Compare screen.

After reconciliation, the object matches both locations and has been imported or deployed through the action.

11.5.2 Using Compare on a Channel Object

Use this procedure if you want to import a channel object from the Identity Vault and the same channel already exists in Designer. You can view the differences and decide whether to reconcile them.

1 Right-click the channel object in the Outline view. Click *Live* > *Compare* to bring up the Designer/eDirectory Object Compare window.

All Compare windows behave as described in Section 11.5.1, “Using Compare When Importing a Driver Object,” on page 260.

After reconciliation, the Channel object matches both locations and has been imported or deployed through the action.

11.5.3 Using Compare On a Policy

Use this procedure if you want to import a policy object from the Identity Vault and the same channel already exists in Designer. You can view the differences and decide whether to reconcile them.

1 Right-click the policy object in the Outline view. Select *Live* > *Compare* to bring up the Designer/eDirectory Object Compare window.

All Compare windows behave as described in Section 11.5.1, “Using Compare When Importing a Driver Object,” on page 260.

After reconciliation, the policy object matches both locations and has been imported or deployed through the action.

11.5.4 Matching Attributes with Designer Properties

The attributes of the object are displayed in the single list. Selecting an attribute displays its value below the attribute list with the Designer value on the left and the eDirectory value on the right. The name displayed in the list is the eDirectory attribute name.

The following tables map the eDirectory attribute to the Designer property page or control where you can change or set the attribute (you can’t make changes inside the Compare window).

- Table 11-1 on page 264 shows Driver Set eDirectory attributes
- Table 11-2 on page 264 shows Driver eDirectory attributes
- Table 11-3 on page 265 shows Channel eDirectory attributes
- Table 11-4 on page 265 shows the Job eDirectory attributes
- Table 11-5 on page 266 shows the Resource eDirectory attributes
- Table 11-6 on page 266 shows the ID Policy eDirectory attributes
- Table 11-7 on page 266 shows the Library eDirectory attribute

- Table 11-8 on page 267 shows the Notification Template eDirectory attributes
- Table 11-9 on page 267 shows the Notification Template Collection eDirectory attributes

Table 11-1 *Driver Set eDirectory Attributes*

Driver Set eDirectory Attribute	Designer Property
DirXML-DriverTraceLevel	Driver Set Properties->Trace->Driver Trace Level
DirXML-XSLTraceLevel	Driver Set Properties->Trace->XSL Trace Level
DirXML-JavaEnvironmentParameters	**Driver Set Properties -> Java**
DirXML-JavaDebugPort	Driver Set Properties->Trace->Java Debug Port
DirXML-JavaTraceFile	Driver Set Properties->Trace->Java Trace File
DirXML-Trace File Encoding	**Driver Set Properties -> Trace - Trace File Encoding**
DirXML-TraceSizeLimit	Driver Set Properties->Trace->Trace File Size Limit
DirXML-LogLimit	Driver Set Properties->Driver Set Log Level->Log Limit
DirXML-LogEvents	Driver Set Properties->Driver Set Log Level->Log Specific Events
DirXML-NamedPasswords	**Driver Set Properties -> Named Passwords**
DirXML-ConfigValues	Driver Set Properties->Global Configuration Values

Table 11-2 *Driver eDirectory Attributes*

Driver eDirectory Attribute	Designer Property or View
DirXML-InputTransform	Policy Set View->Input Transformation
DirXML-OutputTransform	Policy Set View->Output Transformation
DirXML-MappingRule	Policy Set View->Schema Mapping
DirXML-Driver Filter	Policy Set View->Driver Filter
DirXML-ConfigValues	Driver Properties->Global Configuration Values
DirXML-DriverTraceLevel	Driver Properties->Driver Log Level->Driver Log Level
DirXML-EngineControlValues	**Driver Properties -> Engine Control Values**
DirXML-LogEvents	Driver Properties->Driver Log Level->Log Specific Events
DirXML-LogLimit	Driver Properties->Driver Log Level->Log Limit
DirXML-ConfigManifest	Driver Properties->Driver Manifest
DirXML-JavaModule	Driver Properties->Driver Configuration->Driver Module: Java
DirXML-NativeModule	Driver Properties->Driver Configuration->Driver Module: Native
DirXML-DriverImage	Driver Properties->iManager Icon
DirXML-ReciprocalAttrMap	**Driver Properties -> Reciprocal Attributes**

Driver eDirectory Attribute	Designer Property or View
DirXML-TraceLevel	Driver Properties->Trace->Trace Level
DirXML-TraceFile	Driver Properties->Trace->Trace File
DirXML-TraceFileEncoding	**Driver Properties -> Trace - Trace File Encoding**
DirXML-TraceSizeLimit	Driver Properties->Trace->Trace File Size Limit
DirXML-TraceName	Driver Properties->Trace->Trace Name
DirXML-DriverCacheLimit	Driver Properties->Driver Configuration->Authentication->Driver Cache Limit
DirXML-ShimAuthID	Driver Properties->Driver Configuration->Authentication->User ID
DirXML-ShimAuthServer	Driver Properties->Driver Configuration->Authentication->Connection Information
DirXML-ShimAuthPassword	Driver Properties->Driver Configuration->Authentication->Set Password
DirXML-ShimConfigInfo	Driver Properties->Driver Configuration->Driver Configuration->Driver Parameters
DirXML-DriverStartOption	Driver Properties->Driver Configuration->Startup Option
DirXML-ECMAScript	Driver Properties->Driver Configuration->ECMAScript
DirXML-NamedPasswords	**Driver Properties -> Named Passwords**

Table 11-3 *Channel eDirectory Attributes*

Channel eDirectory Attribute	Designer View
DirXML-EventTransformationRule	Policy Set View->Event Transformation
DirXML-MatchingRule	Policy Set View->Matching
DirXML-CreateRule	Policy Set View->Creation
DirXML-PlacementRule	Policy Set View->Placement
DirXML-CommandTransformation	Policy Set View->Command Transformation

Table 11-4 *Job eDirectory Attributes*

Job eDirectory Attribute	Designer View
XmlData	Job Editor, XML cannot be edited directly only through Job Editor UI
DirXML-ServerList	Job Editor
DirXML-Scope	Job Editor
DirXML-EMailTemplates	Job Editor
DirXML-EMailServer	Job Editor

Job eDirectory Attribute	Designer View
DirXML-NamedPasswords	Job Editor
DirXML-TraceName	Job Properties -> Trace
DirXML-TraceFile	Job Properties -> Trace
DirXMl-TraceSizeLimit	Job Properties -> Trace
DirXML-TraceFileEncoding	**Job Properties -> Trace - Trace File Encoding**
DirXML-TraceLevel	Job Properties -> Trace

Table 11-5 *Resource eDirectory Attributes*

Resource eDirectory Attribute	Designer View
DirXML-ContentType	Read only, cannot be edited set at creation time of object
DirXML-DirXMLData	Resource Editor
DirXML-NamedPasswords	Resource Editor

Table 11-6 *ID Policy eDirectory Attributes*

ID Policy eDirectory Attribute	Designer View
DirXML-idPolMin	**ID Policy Properties -> Constraints Minimum**
DirXML-idPolMax	**ID Policy Properties -> Constraints Maximum**
DirXML-idPolPrefix	**ID Policy Properties -> Constraints Prefix**
DirXML-idPolArea	**ID Policy Properties -> Constraints Exclude/Include Text Field**
DirXML-idPolFill	**ID Policy Properties -> Constratints Fill Yes/No**
DirXML-idPolAreaEI	**ID Policy Properties -> Constraints Exclude/Include Radio Button**
DirXML-idPolAccessControl	**ID Policy Properties - Access Control enabled**
DirXML-idPolACL	**ID Policy Properties - Access Control ACL**

Table 11-7 *Library eDirectory Attribute*

Library eDirectory Attribute	Designer View
Description	**Library Properties -> Description**

Table 11-8 *Notification Template eDirectory Attributes*

Notification Template eDirectory Attributes	Designer View
notfMergeTemplateSubject	**Template Editor**
notfMergeTemplateData	**Template Editor**

Table 11-9 *Notification Template Collection Attributes*

Notification Template Collection Attributes	Designer View
notfSMTPEmailHost	Notification Template Collection Properties - > Host Name
notfSMTPEmailFrom	**Notification Template Collection Properties - > From**
notfSMTPEmailUserName	**Notification Template Collection Properties - > User Name**

11.6 Error Messages and Solutions

To view error messages along with their possible solutions associated with importing and deploying files, see Section 21.5, “Deploying Identity Manager Objects,” on page 498.

Documenting Projects

12

When you create a project, it's vital to keep track of how the driver works and how it's implemented into a network. The Document Generator helps you quickly generate customized documentation for your Designer projects. These documents can save you weeks or months of gathering and writing driver specifications and their implementations. To generate a document, choose a document style (it can be the default style that comes with Designer or one that you customize) and a Designer project or portion of a project. The Document feature combines the information and structure of the selected style with the project information in order to generate customized project documentation.

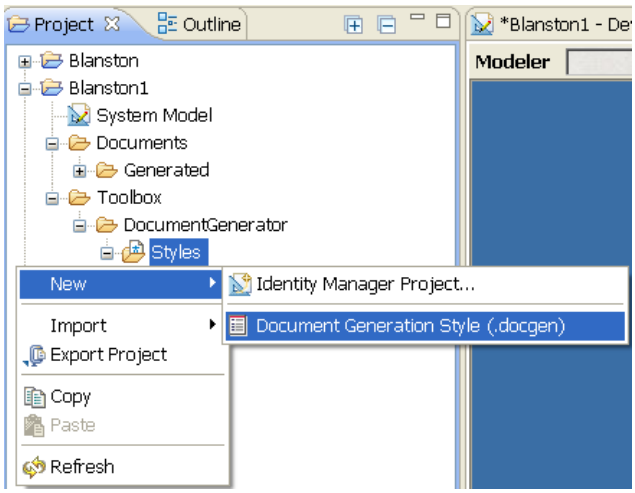
Designer comes with a default document style so you won't need to create a document from scratch. This default style contains everything that you have placed in a project through Designer. You must first use this default style to create your own document style for the project you are working on, then you can either use it as it is or customize it to meet your particular needs, including or excluding information as needed. After you have edited the style to your liking, you can also use it to document your other projects. There is an advanced editing feature that allows you to create your own sections for adding information that you did not create in Designer.

- Section 12.1, "Creating a Document Style," on page 269
- Section 12.2, "Editing a Document Style for Your Needs," on page 271
- Section 12.3, "Generating a Document," on page 272
- Section 12.4, "Using Your Style Template for Other Projects," on page 275
- Section 12.5, "Customizing Styles to Include or Exclude Information," on page 278
- Section 12.6, "Advanced Editing of a Document Style," on page 281

12.1 Creating a Document Style

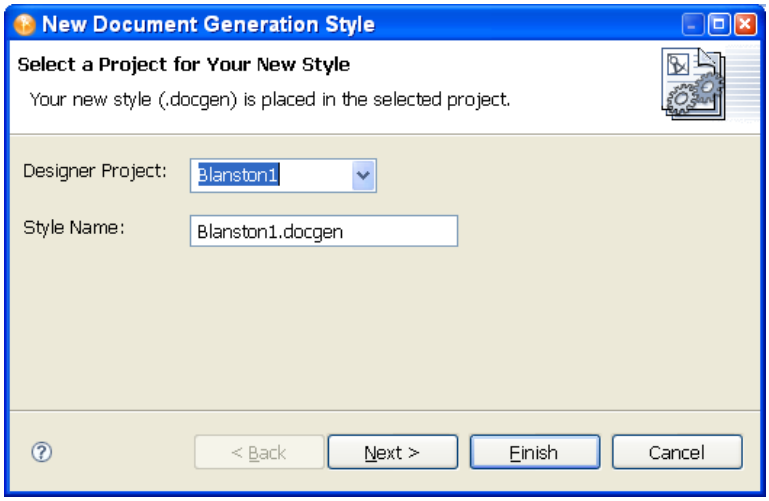
A document style allows you to define how you want your project information to look. In a matter of minutes, you can generate a document that contains all elements that you have placed in a project and define a document style to designate how the information looks, as well as what information you want in a document.

1. Select a project in the Project view, then right-click the *Toolbox* > *DocumentGenerator* > *Styles* icon.
2. Select *New* > *Document Generation Style (.docgen).*



The Document Generation Style (`.docgen`) is the default style that is provided with Designer. You use this as the template for your own `.docgen` style.

3 Specify a name for the Designer project, or use the default of the project's name. Then specify a name for the style, with a `.docgen` extension, or use the default name of the project you are presently in, then click *Finish*.



A `.docgen` file is created that you can use as the basis for designing your own style template. The `.docgen` template is placed in the Style Editor view for your modification (see Section 12.2, "Editing a Document Style for Your Needs," on page 271). You can use the defined elements in your new style template again and again.

4 Use this basic document style template to generate documentation for a project, or customize it for your needs.

- Section 12.3, "Generating a Document," on page 272
- Section 12.2, "Editing a Document Style for Your Needs," on page 271
- Section 12.4, "Using Your Style Template for Other Projects," on page 275
- Section 12.5, "Customizing Styles to Include or Exclude Information," on page 278
- Section 12.6, "Advanced Editing of a Document Style," on page 281

12.2 Editing a Document Style for Your Needs

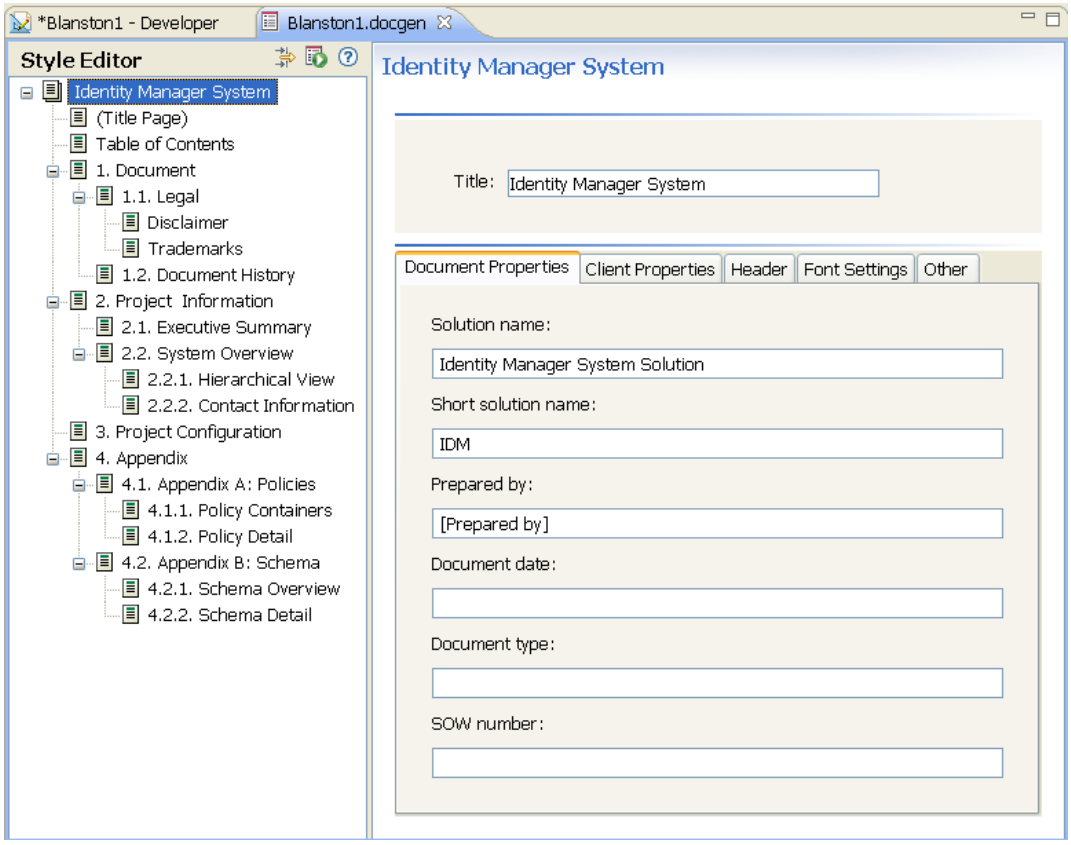
Now that you have created a .docgen style template, you can edit the file to meet your documentation needs. The .docgen file appears under the *Toolbox* > *DocumentGenerator* > *Styles* directory in the Project view. Double-click the .docgen file to bring up the template in the Style Editor.

You can add or modify information in the style template, which in turn affects the documents that you generate. The information that you can customize through the Style Editor appears in the Style Editor view. For more detailed editing, see Section 12.6, “Advanced Editing of a Document Style,” on page 281.

- Section 12.2.1, “Editing a Style Template,” on page 271
- Section 12.2.2, “Editing Sections of a Style,” on page 272

12.2.1 Editing a Style Template

1. Select a project in the Project view, then expand the *Toolbox* > *DocumentGenerator* > *Styles* icon.
2. Double-click the .docgen file. The file appears in the Style Editor view.



3. Use the Style editor to edit sections of a style or to modify the style according to your needs.

The Style editor is divided into two parts, beginning with the Identity Manager and working through the Appendixes. When you click an item under the *Style Editor* section, the right portion of the view changes to display the information associated with the heading. For example, clicking *Disclaimer* under the *Document* > *Legal* heading allows you to edit the disclaimer content.

4 Save your changes. Your changes are saved when you close the Style Editor, or when you click the *Save* icon .

12.2.2 Editing Sections of a Style

1 Select an item (for example, Executive Summary) in the Style Editor view.

2 Enter data or make changes in the left pane.

3 Select other items as appropriate and make changes. The information in the left pane varies, depending on items that you select.

The main areas that you need to pay attention to are the information found under Identity Manager System (Title Page and Table of Contents), Document, Legal, Disclaimer, Trademarks, and Executive Summary.

4 Save your changes. Your changes are saved when you close the Style Editor, or when you click *Save* .

5 Use this document style to generate documentation for a project, or continue to customize it for specific documentation needs.

- Section 12.3, “Generating a Document,” on page 272
- Section 12.4, “Using Your Style Template for Other Projects,” on page 275
- Section 12.5, “Customizing Styles to Include or Exclude Information,” on page 278
- Section 12.6, “Advanced Editing of a Document Style,” on page 281

12.3 Generating a Document

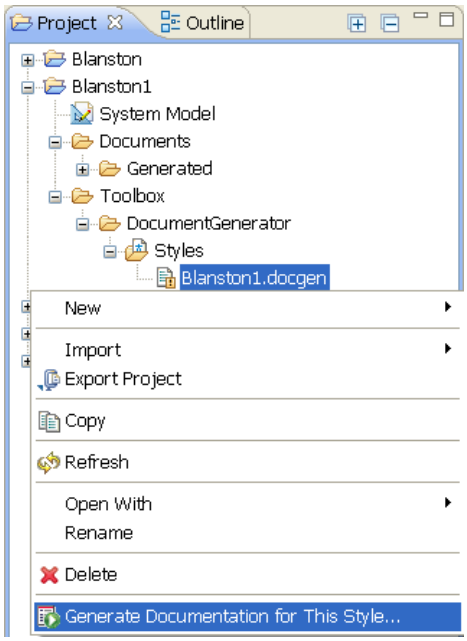
1 (Conditional) If you haven’t yet created a Designer Project, create one.

1a Select *File* > *New* > *Identity Manager Project*.

1b Provide a project name, then click *Finish*.

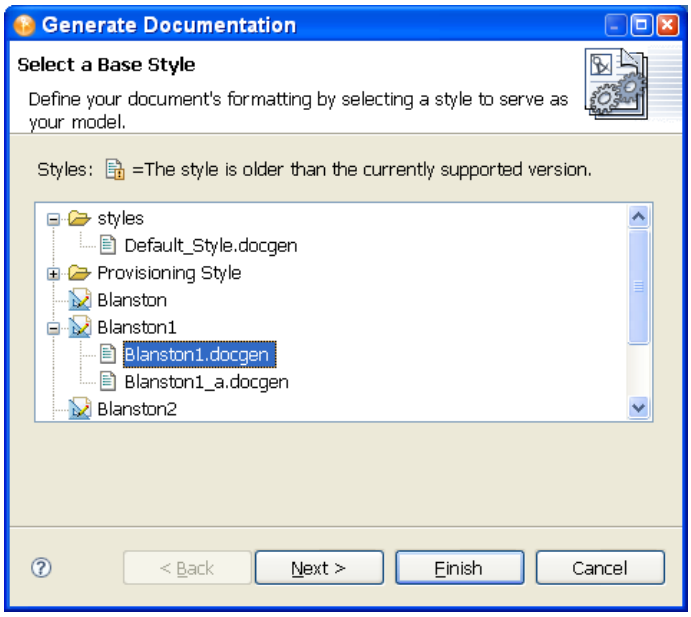
2 (Conditional) If you haven’t yet created a document style that you want to use as a template for your documentation, create one. See “Creating a Document Style” on page 269 and “Editing a Document Style for Your Needs” on page 271.

3 In the Project view, select and expand a project, then right-click the .docgen icon under *Toolbox* > *DocumentGenerator* > *Styles* and select *Generate Documentation for This Style*.

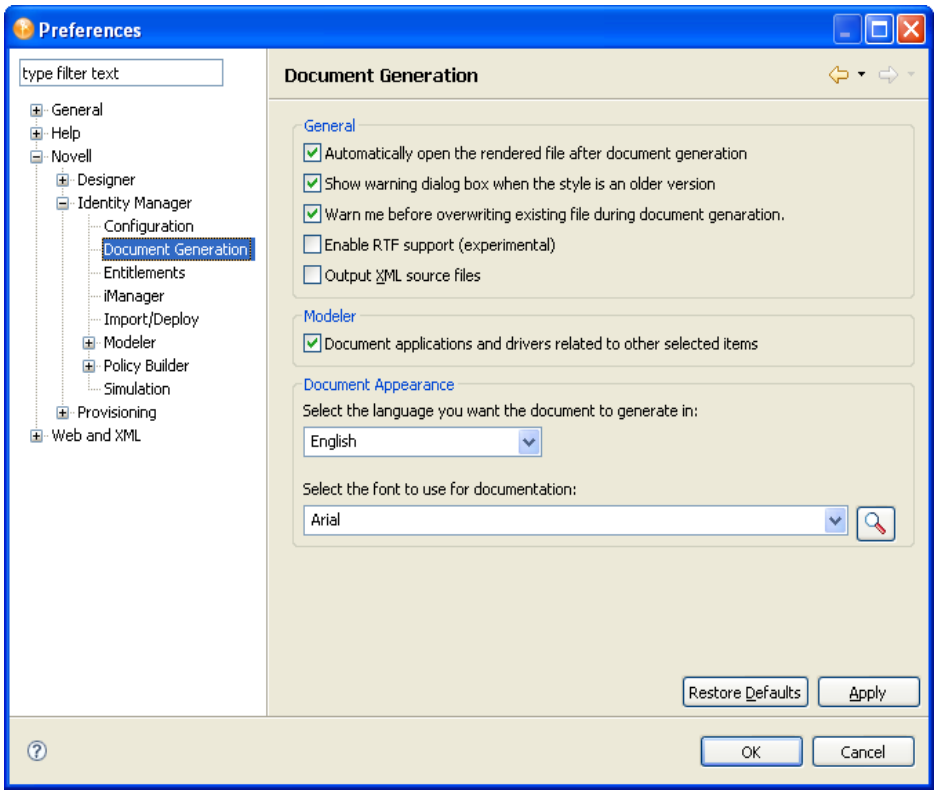


You can also expand the *Toolbox* > *DocumentGenerator* > *Styles* folder and click the `.docgen` file to open the file in the Style Editor, then click the *Document Generation* icon to the right of the Style Editor heading.

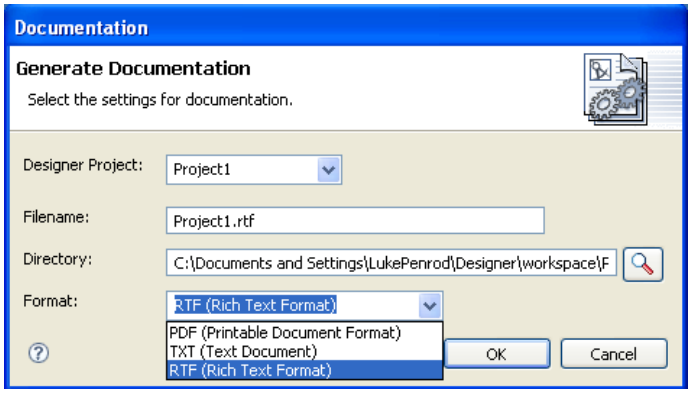
4 (Conditional) If the Project folder you selected does not contain a `.docgen` file, you are asked to select a Base Style. Select a `.docgen` style, then click *Next*.



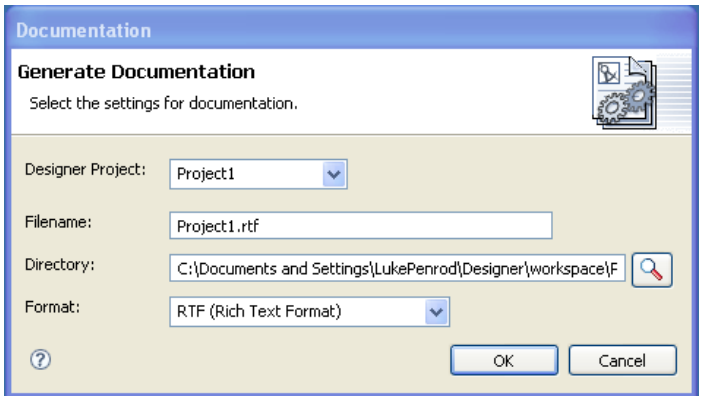
5 (Conditional) Designer includes the ability to generate documents to RTF (Rich Text Format). If you want to enable this functionality, click *Window* > *Preferences* to bring up the Preferences window. Then, under *Novell* > *Identity Manager*, select *Document Generation*.



By selecting *Enable RTF support (experimental)*, you can select the RTF format when creating documents.



6 On the Generate Documentation page, fill in the needed information.



- Select the name you want to call the file, or keep the default name. If you are generating the whole document, the default name often suffices. If you are generating a document for a section, such as an Executive Summary, name the file to reflect the section you are documenting.
- Select the directory where you want to store the document. If you use the default output location that appears in the Directory field, your generated document is visible under the `Documents\Generated` folder of the Project View.
- Select the format for the file. If you have enabled rich text formatting, you can select *PDF (Printable Document Format*), *TXT (Text Document)*, or *RTF (Rich Text Format)*.

7 Click *Finish* to generate the document. The document appears in the current `Project > Documents > Generated` folder unless otherwise specified.

PDF files must be viewed through a PDF viewer, such as Adobe Acrobat*. If Adobe Acrobat is installed on your workstation, Designer launches the document in Acrobat. RTF files can be viewed in any word processor that can handle Rich Text Formatted files, such as Wordpad in Windows.

The Filter editor provides an option to add notes to class and attributes, and these notes are added to the documentation. Password synchronization on drivers is also documented, showing how the administrator has set up password synchronization for the Publisher and Subscriber channels. You can also document contact information on the administrator for Identity Vault and application objects, as well as reciprocal mapping information.

12.4 Using Your Style Template for Other Projects

To generate documentation for any project, you can use the default style provided with Designer or you can use your own customized styles.

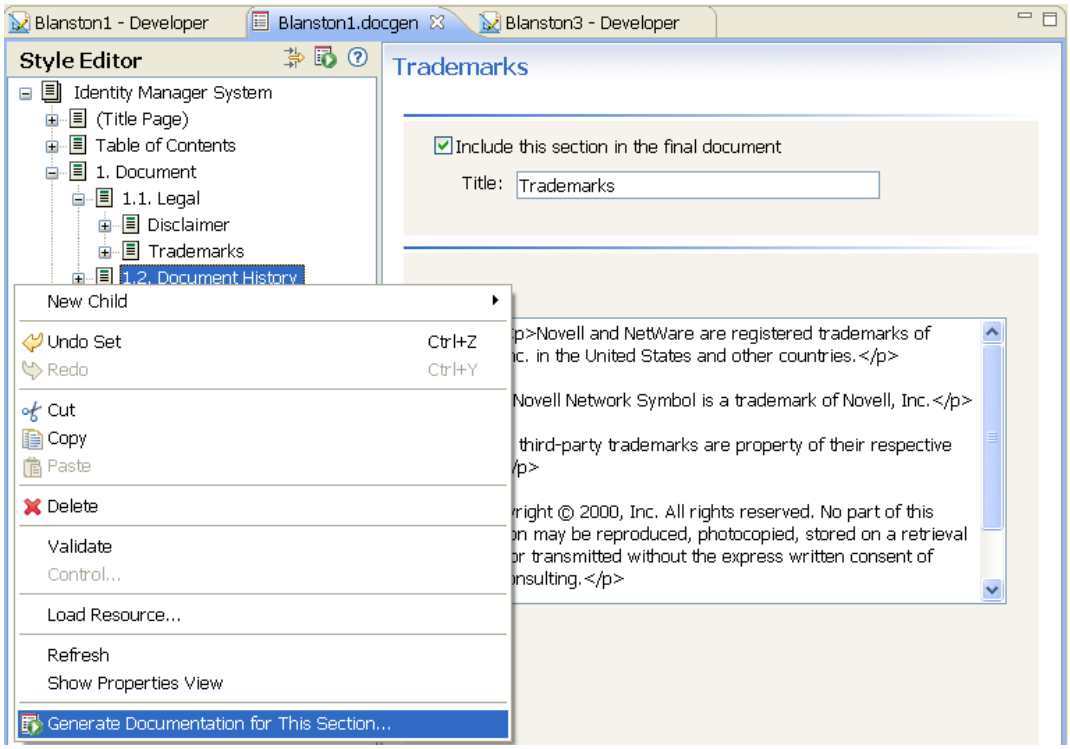
- Section 12.4.1, “Document a Section of the Style,” on page 275
- Section 12.4.2, “Document Multiple Sections of the Style,” on page 278

12.4.1 Document a Section of the Style

Instead of generating a document for the entire project, you can generate a document for a selected section of a project.

1 With the project’s `.docgen` file open in the Style Editor pane, right-click a section of the style.

2 Select *Generate Documentation for This Section*.



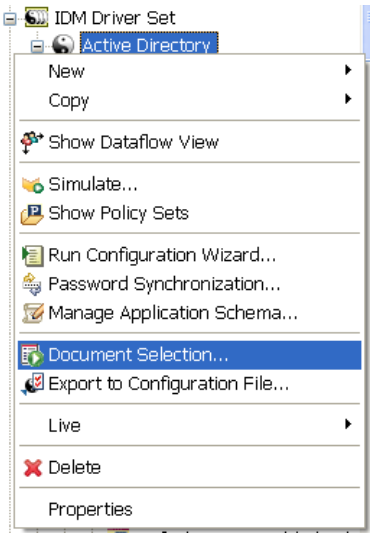
3 In the Generate Documentation window, type a different project name in the *Filename* field (for example, DocHistoryofMerger), then click *OK*.

Specify which portion or portions of the project you want to include in the generated document.

You can document domains, Identity Vaults, driver sets, drivers, and applications using the Modeler view or the Outline view (use the Ctrl key to select multiple items). Document generation also ties in with schema notes, classes and attributes. You can find out more about this in Chapter 6, “Managing the Schema,” on page 137.

For example:

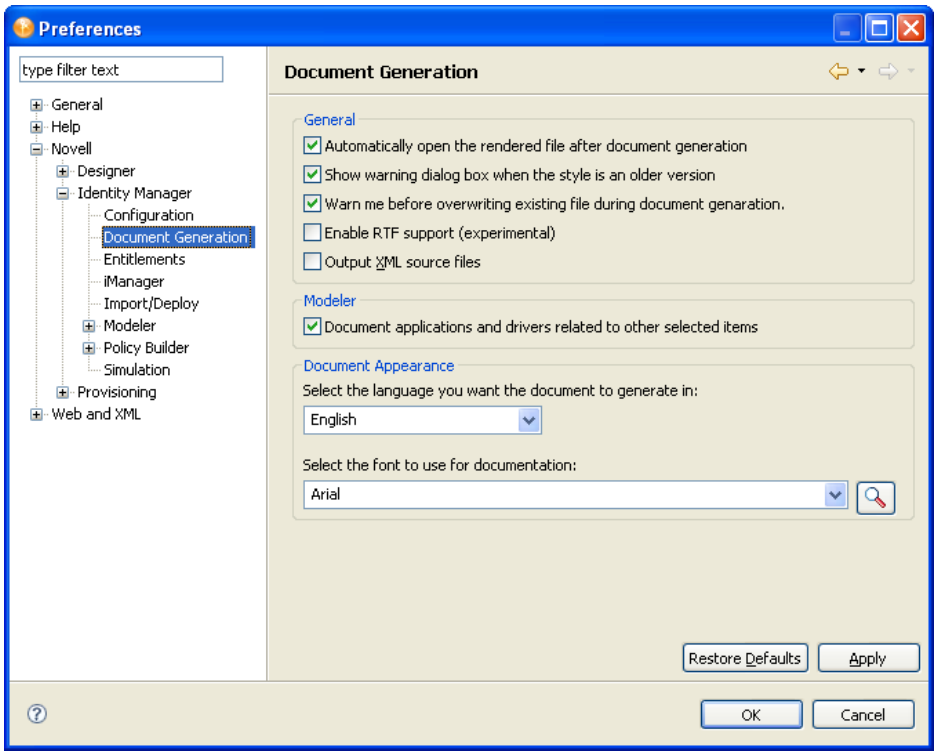
1 To document a specific driver in a project, right-click the driver in the Modeler or Outline view and select *Document Selection*.



2. Select the .docgen style sheet for the document and click *Next*.
3. Give the document a filename, such as the driver's name, select the document's format, and click *Finish* to generate the driver document.

By default, documenting an application includes the connected driver (and its related objects). Likewise, documenting a specific driver includes its connected application. However, you can change this behavior in the Documentation Generation's Preferences page.

1. Click *Window* > *Preferences* to bring up the Preferences window.
2. Under *Designer for IDM*, select *Document Generation*.



Under the Modeler heading, the *Document applications and drivers related to other selected items* option is selected by default, which means that directly related items are included in the documentation. For example, by default, documenting a driver set includes the direct children (the applications) as well as some information of the direct parents (the Identity Vault and domain) to give context to the driver set. Deselecting this option excludes direct children of the selected item.

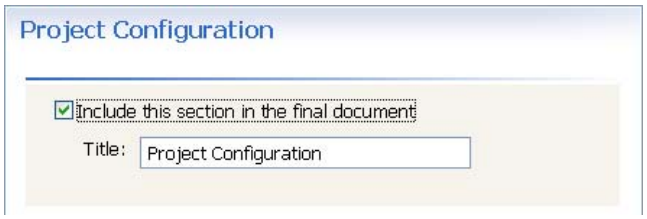
3 Select or deselect the options you want, then click *OK*.

12.4.2 Document Multiple Sections of the Style

If necessary, you can generate only selected sections so that peers can help you with information in the selected sections.

1 If you have not already done so, double-click the `.docgen` file to bring up the template in the Style Editor.

2 Select or deselect section headings. Each section and child section has a check box entitled *Include this section in the final document*. By default the box is checked, as shown below.



To limit the sections you want to document, deselect the check boxes in the sections you don't want to generate.

3 Click the Generate Document icon to the right of the Style Editor heading and generate your document.

Give the document a unique name to reflect the type of information it includes.

12.5 Customizing Styles to Include or Exclude Information

Novell does not recommend that you document all Identity Vault schemas unless you need to.

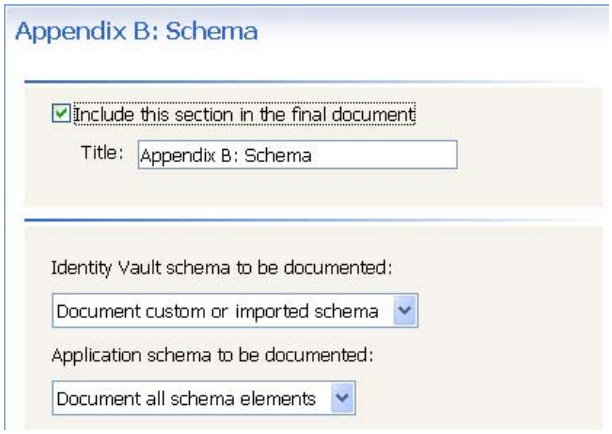
- Section 12.5.1, "Identity Vault Schema and Application Schema," on page 278
- Section 12.5.2, "Using Project Configuration to Limit Information," on page 279

12.5.1 Identity Vault Schema and Application Schema

The defaults for Identity Vault schema have been changed to include custom schemas and any modified changes to the Identity Vault base schema. For application schema, Designer includes all schemas by default. However, these can be turned off.

Select the *Appendix B: Schema* heading the in the Style Editor view. This brings up the Appendix B section template in the right side of the Style Editor view.

Figure 12-1 *The Appendix B: Schema Section Template*



The Appendix B: Schema section has three selections:

- **Include this section.** The *Include this section in the final documentation* check box allows you to include or not include Appendix B information in the documentation. By default, the box is selected to include this information. Deselect the check box if you do not want to include application or Identity Vault schemas in the document.
- **Document Custom and Imported Identity Vault Schema.** By default, the *Identity Vault Schema to be documented* selected documents any schema that you import from the Identity Vault or that you create. The choices are *Document custom or imported schema*, or *None*.
- **Document all Application Schema.** By default, the *Application schema to be documented* selection includes all of the application schema. The choices are *Document all schema elements*, or *None*.

12.5.2 Using Project Configuration to Limit Information

The Project Configuration heading allows you to include or deselect information on:

- Identity Vault
- Driver Sets
- Drivers
- General

The following table shows what type of information can be included or excluded in these areas.

Table 12-1 *Project Configuration*

Identity Vault	Information to Select or Deselect
Selected	Include host information
Selected	Include username information
Selected	Include deployment context information
Selected	Include driver set names

Identity Vault	Information to Select or Deselect
Selected	Include policy library on Identity Vaults
Deselected	Include e-mail notification templates
Deselected	Include XML source while documenting policies under the policy library
Deselected	Include XML source while documenting credential provisioning objects under the policy library
Deselected	Include XML source while documenting mapping table objects under the policy library
Driver Set	Information to select or deselect
Selected	Include server information associated with the driver set
Selected	Include driver set Global Configuration Value (GCV)
Selected	Include the policy library on driver sets
Selected	Include job objects on driver sets
Deselected	Include the XML source for policies under the policy library
Deselected	Include the XML source for credential provisioning objects under the policy library
Deselected	Include the XML source for mapping table objects under the policy library
Deselected	Include the XML source for job objects
Driver	Information to select or deselect
Selected	Include the driver filter policy
Selected	Include policy set
Selected	Include server-specific information for this driver
Selected	Include Remote Loader configuration
Selected	Include entitlements
Selected	Include credential provisioning
Selected	Include mapping table
Selected	Include ECMA script resource object
Selected	Include job objects
Deselected	Include the XML source when documenting entitlement objects
Deselected	Include the XML source when documenting credential provisioning objects
Deselected	Include the XML source when documenting mapping table objects
Deselected	Include the XML source when documenting job objects
General	Information to select or deselect
Deselected	Include passwords
Selected	Page break after this section

WARNING: Credential provisioning for the XML source might contain passwords that will be displayed in clear text. If this option is selected, passwords are displayed in clear text and documentation includes all passwords in the project.

Designer can only display passwords if they have been set in the Designer. If you import a project from the Identity Vault, Designer cannot show the passwords in the documentation.

12.6 Advanced Editing of a Document Style

In addition to selecting and deselecting the content of a document style, you can also change the layout and usability of your document style. You do this by editing the attributes that are associated with certain sections. You can also create additional sections for your documents as you see fit.

- Section 12.6.1, “What’s In the Advanced Editing Mode,” on page 281
- Section 12.6.2, “A Walk-through Tutorial,” on page 288
- Section 12.6.3, “Selecting a Language for Generated Documents,” on page 294
- Section 12.6.4, “Double-Byte Font Support,” on page 295

12.6.1 What’s In the Advanced Editing Mode

The Advanced Editing icon lets you toggle between simple editing and advanced editing modes. By using the advanced editing mode, you can define information and a structure that is different from the default template already attached to a predefined style. In this example, the Title Page template is shown in its XSL format, which you must maintain.

Figure 12-2 *Viewing a Predefined Template*

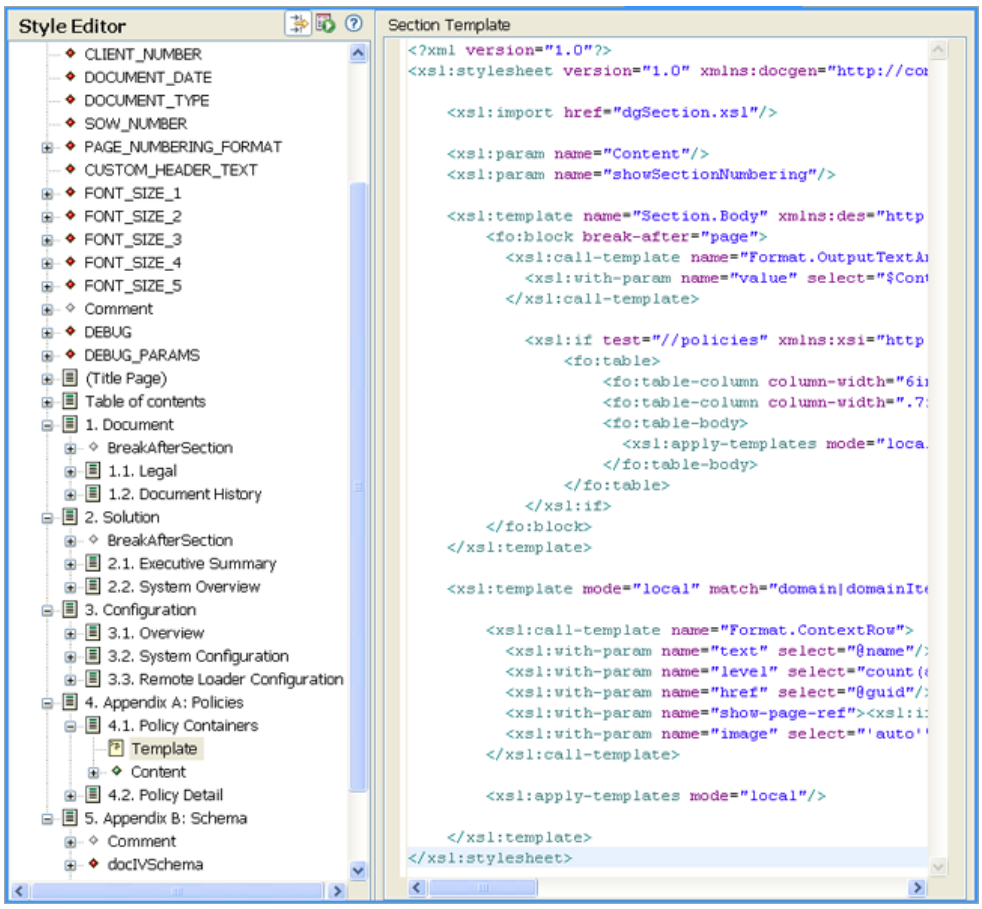


Table 12-2 *Style Editor Legend*

Icon	Name	Description
	Green Page	A green page means it's a titled section. The title appears when you generate the document.
	Grey Page	A grey page means it's not a titled selection. These pages are also in parentheses; for example, (Title Page).
	White Page	A white page means this section is disabled and is not included when you generate the document.
	Template	A yellow template page gives specific format and styles that are included when you generate a document.
	Global Attribute	A global (red) attribute means it is passed down to every section below it (all children sections).
	Local Attribute	A local (green) attribute means it is only used by the section in which it appears.
	Grey Attribute	A grey attribute is used for comments.

Icon	Name	Description
	Control Icon	A Control type defines the functionality that you can give to Attributes. Each Control type has a different functionality.

Advanced editing mode allows you to add the following:

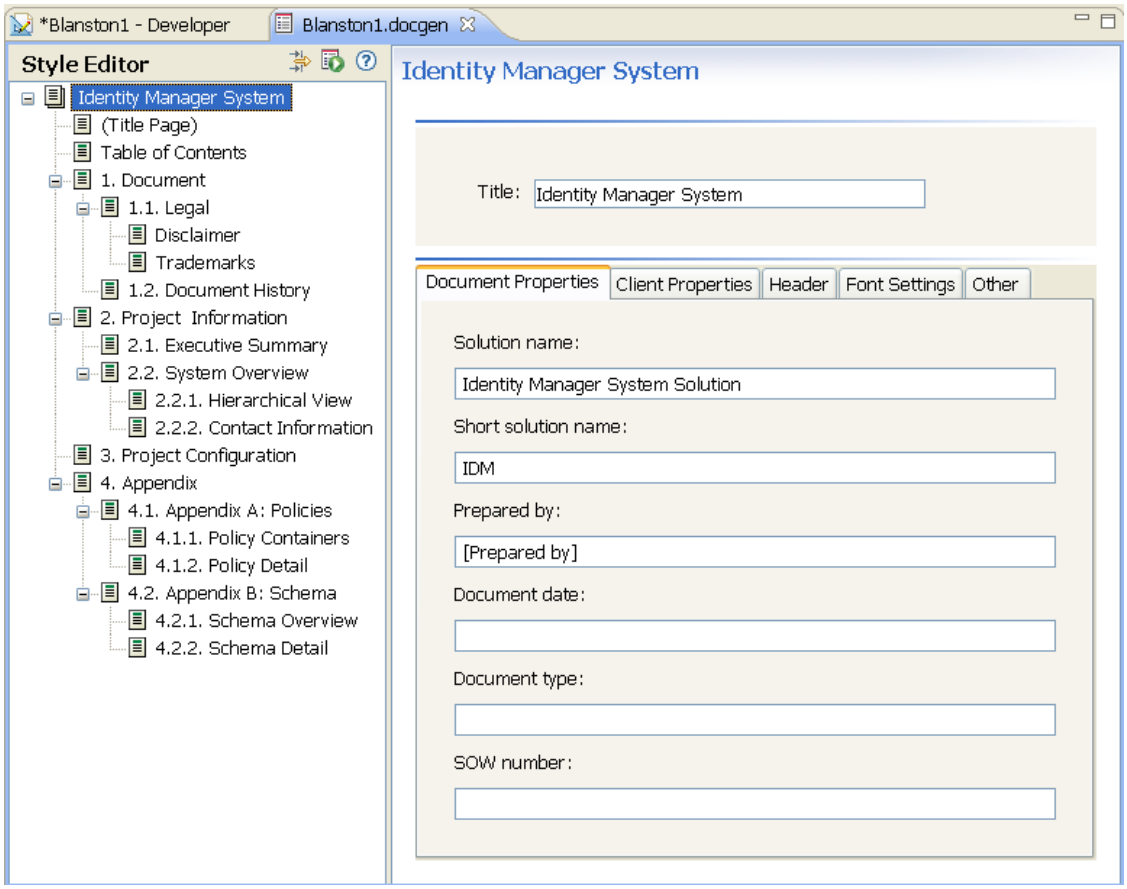
- “Sections” on page 283
- “Viewing or Editing Properties of a Section” on page 285
- “Templates” on page 286
- “Attributes” on page 286
- “Controls and Parameters” on page 287

You can have multiple sections in a document, but only one template per section. The template defines the section’s layout; however, you can use the default template for newly created sections. You can also have multiple attributes defining how the section looks, as well as multiple controls. You use parameters (such as names and values) to specify options for a Control type. A Parameter is a general name for a child item of a Control. The name of the Control denotes the type of control and what you can perform by using that type.

Sections

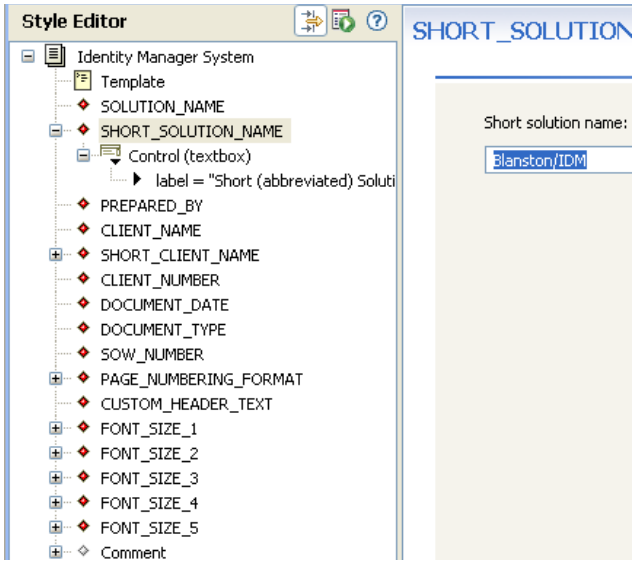
Sections are blocks of the documentation composed of attributes, parameters, templates, and controls through XSL programming. Section content includes a Title, Body, and children or subordinate information. The following example shows the Section Properties page of the Identity Manager System as seen through the simple edit mode.

Figure 12-3 *Section Properties Page*



The Identity Manager System section contains a section title (this can be changed), along with a number of tabs (attributes): *Document Properties*, *Client Properties*, *Header*, *Font Settings*, and *Other*. Each of these tabs contain fields that are editable; for instance, you can give the section title a different name. When you click the *Advanced Editing* icon, you see that the Identity Manager System section contains one template that includes several attributes, controls, and parameters underneath the heading.

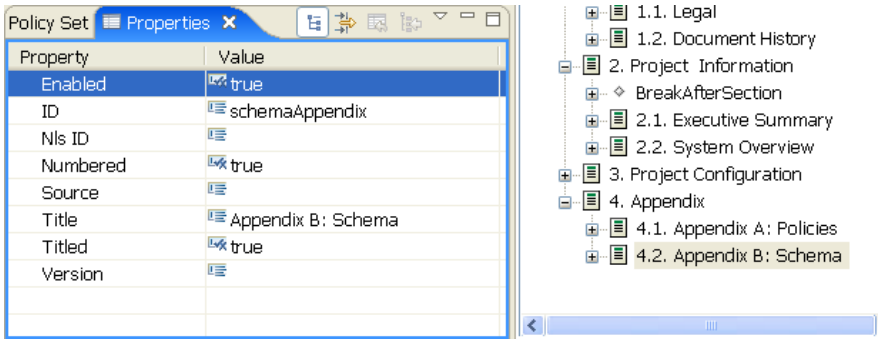
Figure 12-4 *Advanced Editing Mode*



Viewing or Editing Properties of a Section

If you click a section within the Style Editor and look in the Properties view (by default the lower left corner of Designer), you see the values associated with the selection. (If you do not see the Properties view by default, right-click a section and select *Show Properties View*.)

Figure 12-5 *The Properties View of the Appendix B: Schema*



These values are edited in the Properties view. The Values for the section heading are listed in the following table:

Table 12-3 *Values of a Section*

Property Name	Description
Enabled	(True/False) Indicates whether this section is enabled. You can change this setting by using the section Style Editor, which is the editor shown to the right of the hierarchical view.
ID	Used for reference. Most the time, ID is left empty. However, you can specify an ID for convenience in finding this section during the transformation process.

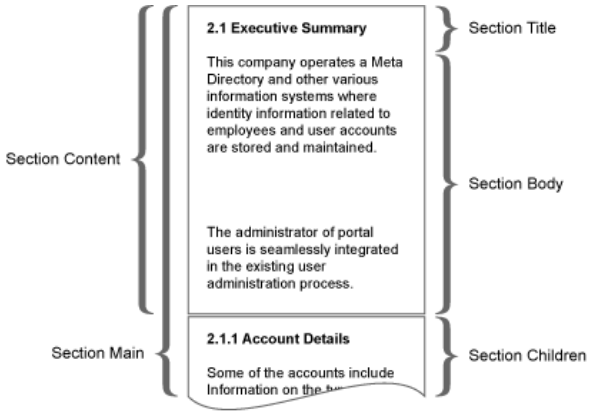
Property Name	Description
NLS ID	Used for reference. Most the time, NLS ID is left empty.
Numbered	(True/False) Indicates whether this section should be included in the numbering and placed in the Table of Contents.
Source	Data source used to transform the template. For example, designer, style, and none.
Title	The value to be displayed as the title. You can change this setting by using the section Style editor, which is the editor shown to the right of the hierarchical view.
Titled	(True/False) Indicates whether the title value should be shown in the generated document. Otherwise, it is used only in the GUI for context.
Version	The version of the section.

NOTE: Values change, depending on what you select under the Style Editor view. For example, an attribute shows different properties than a section or a template.

Templates

A template is the XML source that defines the overall layout of a generated page. For instance, the Title Page contains a template, as well as a number of headings. The following figure illustrates the parts of the Executive Summary template. For more information about templates, see "A Walk-through Tutorial" on page 288.

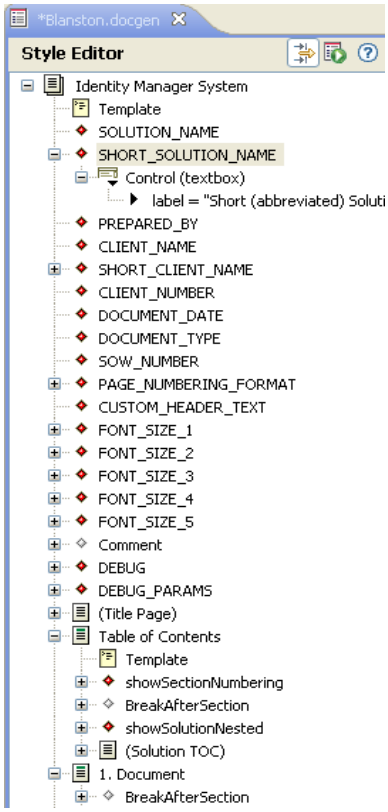
Figure 12-6 *Parts of the Executive Summary Template*



Attributes

Attributes are the child elements of a section. For example, clicking the Advanced Editor mode while selecting the section title Identity Manager System reveals the following attributes in red (global), green (local), and grey (comment):

Figure 12-7 *Attributes*



Controls and Parameters

You can add parameters to control the appearance of a style. For example, in the Advanced Editing mode, the structure of the Short (abbreviated) Solution Name entry is a global attribute that contains a control and a label, and the control type known as textbox allows anyone to type a name that appears in the generated document. Use the Properties view to edit controls.

Designer's supported parameters or values for controls include the following:

Table 12-4 *Supported Parameters*

Control	Parameter or Value	Description
Table	columns="3" header="date" width="30" label="show this" addrows removerows	Number of columns to show in the control. Column header text. Column width for each column. Explains what you see in this control. Displays a button to perform this function. Displays a button to perform this function.
File	extensions=".jpg;*.gif" label="show this"	Supported extensions separated by a semicolon (*.jpg;*.jpeg;*.gif). (One file only.) The label explains what you can do with this control.

Control	Parameter or Value	Description
Select (Identity Manager System/ font Settings)	option="font 1" option="font 2" label="show this"	Parameters allow font point selection, such as option = "20pt" and option = "24pt" The label indicates what you can do with this control.
Checkbox	label="show this"	The label explains what you can do with this control. It includes a check box.
Textbox or Textarea	label="show this"	The label explains what you can do with the text box or text area control. You edit these controls through the Properties view.
Comment	label="show this"	Allows you to add comments to help users. You edit this control through the Properties view.

12.6.2 A Walk-through Tutorial

Now that you better understand what components are necessary in order to add advanced functionality to your template, use this section to create a new section, add an attribute, and view the source.

- "Creating a New Section in a Style" on page 288
- "Adding an Attribute to a Style" on page 288
- "Enabling Documents to Recognize Your Additions" on page 289
- "Viewing the Source" on page 294

Creating a New Section in a Style

To insert an additional section into the Style Editor:

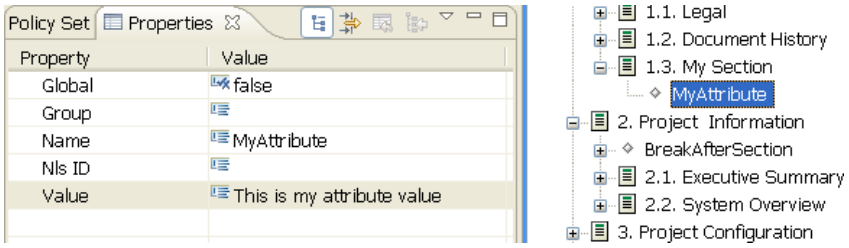
1. Create or open an existing `.docgen` file in the Style Editor.
2. Click the *Enable Advanced Editing* icon.

   The tree view of the document outline expands to include additional objects, (such as attributes, template, and parameters).
3. Right-click the parent section where you want to add your new section, then select *New Child* > *Section*. Specify a new section name; for this example, call it "My Section."
4. (Optional) Reorder the section by dragging the section object to a different location in the navigation tree. You can also copy and paste within this style or other styles.
5. Click the *Save* icon, then continue with "Adding an Attribute to a Style" on page 288.

Adding an Attribute to a Style

1. Right-click a section under the Style Editor view. Select *New Child* > *Attribute*.
2. Specify the attribute name in the Attribute Name window. For example, MyAttribute with no spaces. Click *OK*.

3 Specify a value under the Value property in the Properties view. For example, *This is my attribute value.*



The Properties view shows the following values for attributes (attributes are defined through controls and parameters):

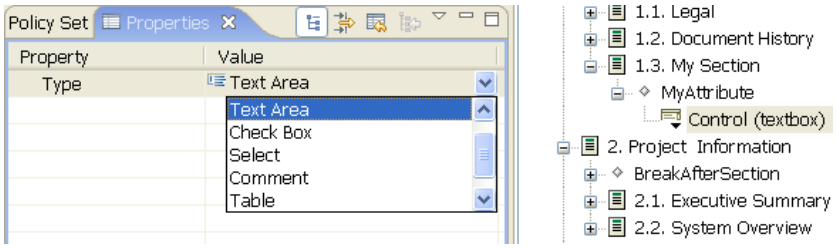
- **Global:** passed down to subsections.
- **Group:** used to group attributes together. These appear as part of a tab in Style Editor's simple mode.
- **Name:** The attribute's name.
- **NLS ID:** The attribute's NLS identification.
- **Value:** The attribute's value.

You can also show your attribute with another control type. The following example first creates a control, then changes the control type from check box (the default) to something else, such as a text area.

1 Right-click *MyAttribute* and select *New Child* > *Control*.

2 With your cursor on the control you just created, change the control type value to *Text Area* in the Properties view.

3 Click your section to see the changes take place.



Enabling Documents to Recognize Your Additions

After you have added attributes, your generated document doesn't include information from these attributes until you do one of the following:

- Make sure your section is a leaf section (does not contain child sections).
- Create a template that uses the attribute explicitly. This is usually the preferred method because you can display the attribute values exactly the way you want.

Method 1: Set the Section Source to “Style”

If you create a section without a template (and the template is a leaf section containing no section children), the default template generates the attribute values with the document. You do not need to do anything. You can generate a document for just that section by right-clicking the section head and selecting *Generate Documentation for This Section*. Or you can click the *Generate Document* icon at the top to generate the whole document.

NOTE: Text boxes, text areas, and tables are the only attributes that are generated through the default template (check boxes, selects, and comments are not generated).

Method 2: Add a Template for a Custom Layout

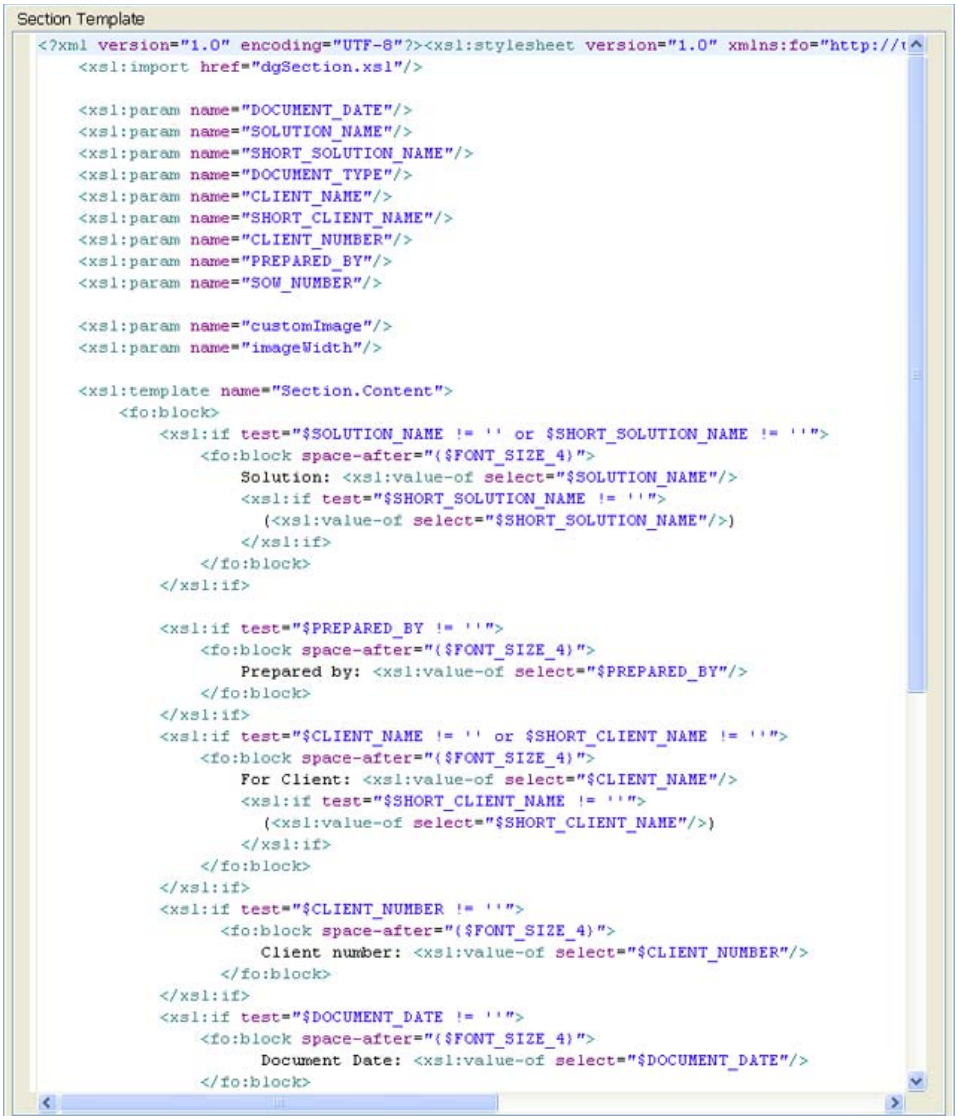
Complete the following tasks:

- “Creating a Template” on page 290
- “Creating Another Section and Template” on page 292

Creating a Template

1. Right-click your newly created section.

2 Select *New Child* > *Template*.



The template has some default content that consists mostly of comments, which helps you get started on your first template. This is shown in the next task; for now, replace the comments in this template with the following XSL commands:

```
<xsl:param name="MyAttribute"/>


<xsl:template name="Section.Body">
  MyAttribute:
  <xsl:call-template name="Format.OutputTextArea">
     <xsl:with-param name="value" select="$MyAttribute"/>
  </xsl:call-template>
</xsl:template>
```

There is a Format.OutputTextArea call in the XSL that is a helper function included with the Document Generator Core Support Templates. Because HTML code is allowed in text areas, this ensures that it's interpreted and escaped properly. If you want to see the core XSL library calls for documentation generation, see "Document Generator Core Support Templates" on page 531.

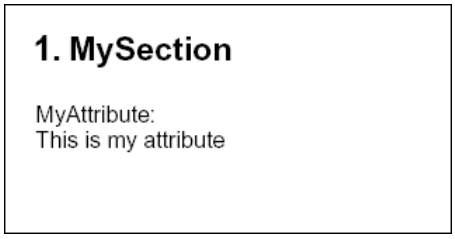
Your template should look like this:

Figure 12-8 *Example Template*

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1
  <xsl:import href="dgSection.xsl"/>

  <xsl:param name="MyAttribute"/>

  <xsl:template name="Section.Body">
      MyAttribute:
      <xsl:call-template name="Format.OutputTextArea">
        <xsl:with-param name="value" select="$MyAttribute"/>
      </xsl:call-template>
  </xsl:template>

</xsl:stylesheet>
```

Generating a document for this section should give you something like this:

Figure 12-9 *Sample Section*



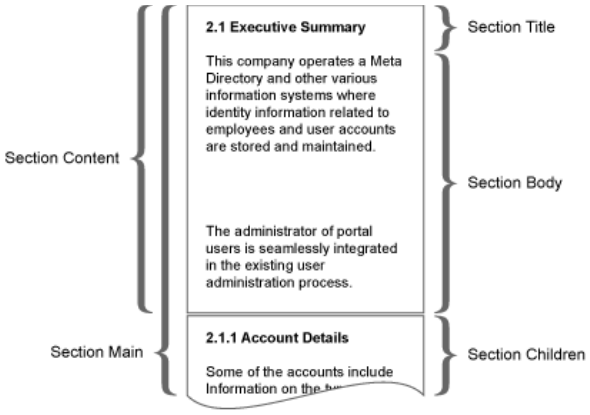
Creating Another Section and Template

1. Right-click your newly created section and select *New Child* > *Section*. In the Section Name window, name the new section Table of Contents and click *OK*.
2. Right-click this new section and select *New Child* > *Template*. Carefully read through the comments in this template. These details are important.

   When you create a custom section, you are inserting some information into the document. As the comments mention, developers usually override one of the following template functions:

   - Section Body (most common)
   - Section Content
   - Section Title

   The following image illustrates which section is being defined. As a developer of the style, you write this template to overwrite the area that is of interest.



For this example, you should overwrite the Section.Body because you don't want to change the default behavior of the Title, nor do you want to change the way other sections are related to this one. (You can use the hierarchical view to control this with the default template if necessary).

3 To overwrite the Section Body, simply uncomment the sample function that is shown in the default template. This is shown below:

Section Template

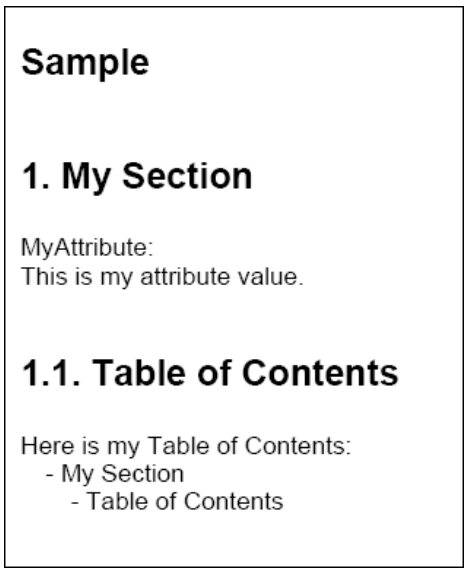
```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/
  <xsl:import href="dgSection.xsl"/>

<!-- /**
      *  Table of Contents example (remove commenting below
      *
      *  This template assumes you have your section source
      *  to 'style' (the default value), and demonstrates h
      *  render data taken from an XML source.
      *
      */ -->
  <xsl:template name="Section.Body">
    <fo:block>Here is my Table of Contents:</fo:block>
    <xsl:apply-templates/>
  </xsl:template>

  <xsl:template match="children">
    <fo:block margin-left="{count(ancestor::*)}em"> - <xsl:
    <xsl:apply-templates/>
  </xsl:template>

</xsl:stylesheet>
```

If you render your document at this point, you get no content in your Table of Contents (other than the surrounding text). This is because this template assumes that the style source has been specified for this section. To specify the source:

1 Click the Table of Contents section.

2 From the Properties view, set the source to *style*.

3 Right-click *MySection* and select *Generate Documentation for This Section*.



Viewing the Source

When your generate your document, you'll notice there is an *Output XML Source Files* option. Click the box next to this option to turn it on. You'll see `.xml` source files appear where you are generating the document. These source files are the XML data that is used in your template when you set the source (for example, to "style"). Designer 1.1 and above include the following sources:

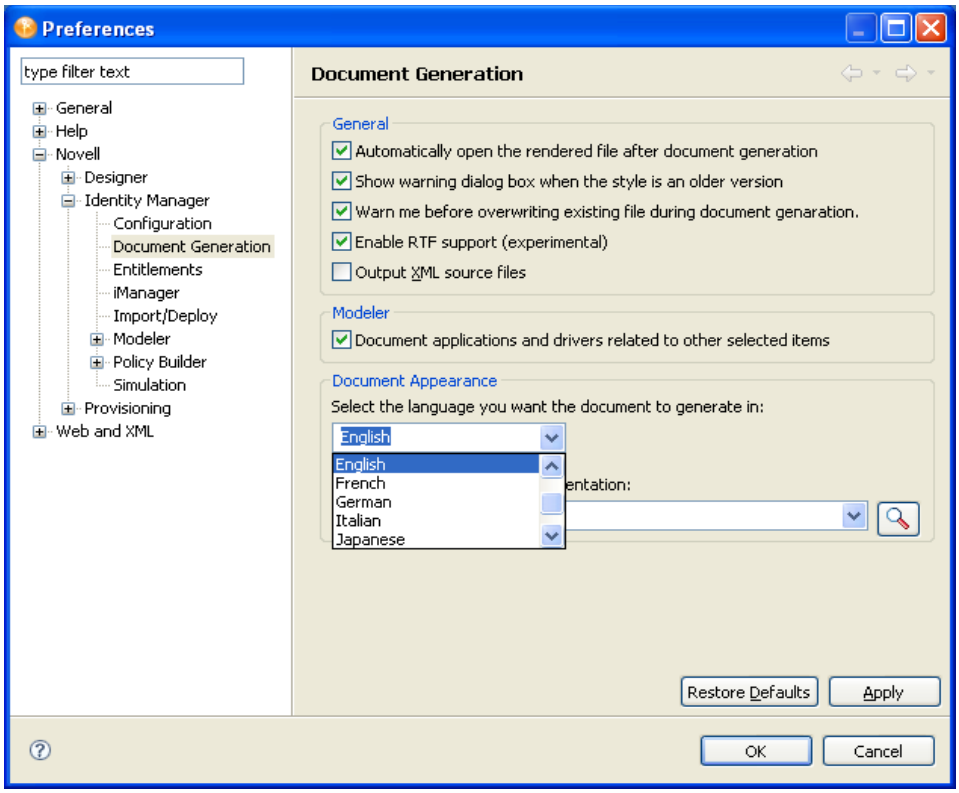
Table 12-5 *Sources*

Source Key	Description
none	An empty source, used when no source is specified or when "none" is specified.
style	The XML source of the style, used to build things like the Table of Contents.
designer	A source that has been defined by an extension point for the Designer model. This contains all information about the configuration of your Designer project.

12.6.3 Selecting a Language for Generated Documents

You can also select the language you want to print the document in.

1 Click *Window* > *Preference* > *Designer for IDM* > *Document Generation*. Under the *Document Language* heading, select the language you want to use for document generation.



Current languages include:

- Chinese Simplified
- Chinese Traditional
- Dutch
- English (default)
- French
- German
- Italian
- Japanese
- Portuguese Brazil
- Spanish

2 After you select a language, click *Apply*.

3 Click *OK* to close the Preference page.

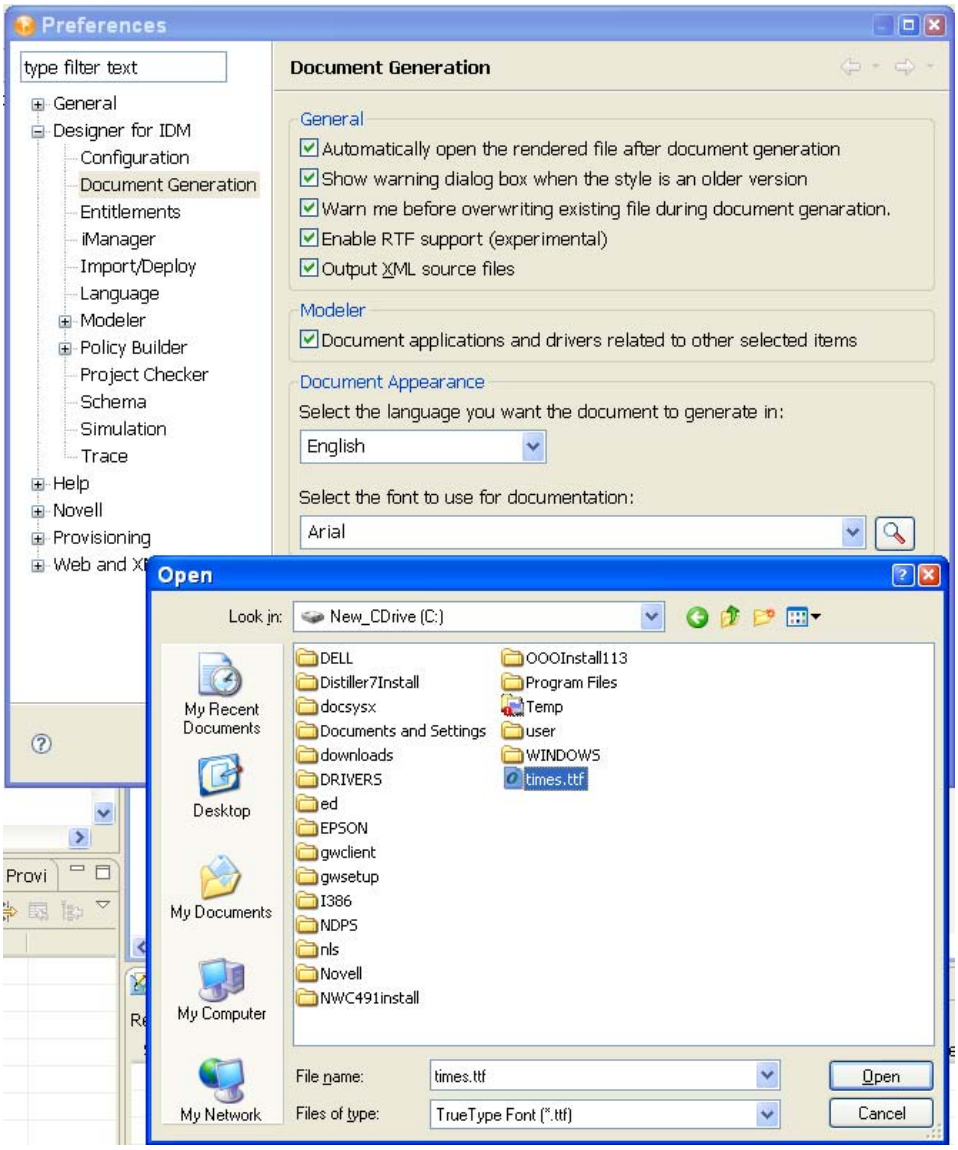
12.6.4 Double-Byte Font Support

Designer now has double-byte font support for the Document Generation feature. If you select a language that uses double-byte characters, such as Chinese Simplified, Chinese Traditional, and Japanese, as a starting point Designer automatically installs the font `Proportional Mincho`. A good font that covers both proportional spacing and double-byte support is `Arial Unicode MS`.

For English and other languages, the default font is Arial.

To add a font for your specific language:

1 Click *Window* > *Preference* then expand *Novell* > *Identity Manager* and select *Document Generation*. Under *Document Appearance*, select the font you want to use.



To change the font on a Windows workstation, you must first copy the font file from the `C:\Windows\Fonts` directory to another directory. You can then use the *Browse* button to select the font.

To change the font on a Linux workstation, browse to the `usr/share/fonts/truetype` directory, or to another directory containing the fonts you want.

2 Click the *Browse* button to bring up the Open window, change to the directory where you placed the font, then click *Open*.

You can also type the directory and font file name into *Font Settings* field, or use the drop-down menu to select a font that you have previously selected.

3 Click *Apply*, then click *OK*.

Using the above steps globally changes the font in the generated document, and also adds double-byte font support for your selections.

Using Entitlements

13

Identity Manager allows you to synchronize data between connected systems. Entitlements allow you to set up criteria for a person or group that, once met, initiate an event to grant or revoke access to business resources within the connected system.

You can think of an entitlement as a permission slip. For example, if you want a new employee to be given an Active Directory account when he is added to your Human Resource system, he must have a permission slip, or entitlement, for the Active Directory account. If the user doesn't have the permission slip, he doesn't receive the account. This gives you one more level of control and automation for granting and revoking resources.

Use Designer to create entitlements and deploy them into existing Identity Manager drivers. Designer allows you to create entitlements through the Entitlement Wizard, which gives you a graphical interface where you can create the entitlement step by step. Because of this graphical interface, we recommend using Designer for creating and editing entitlements.

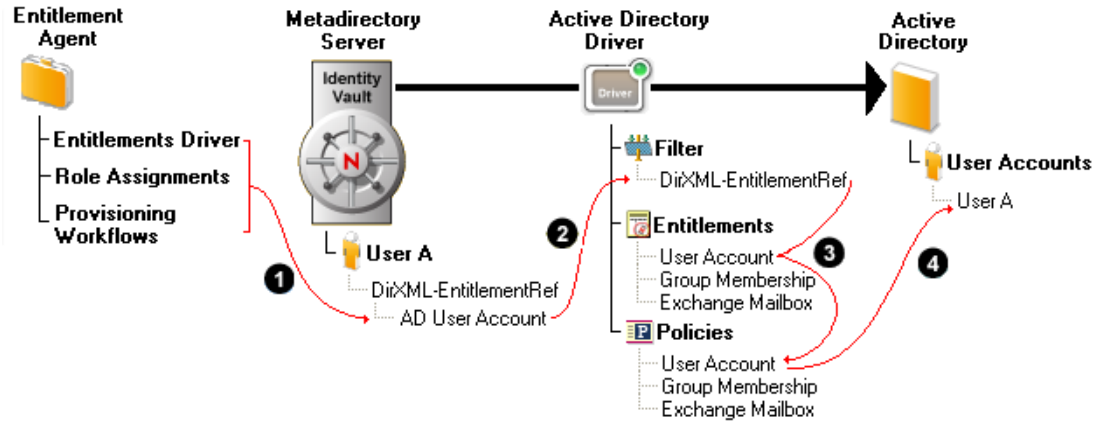
There are four aspects to making entitlements work effectively: design, creation, editing, and management.

- Section 13.1, "How Entitlements Work," on page 299
- Section 13.2, "Designing Entitlements," on page 301
- Section 13.3, "Creating Entitlements through the Entitlement Wizard," on page 304
- Section 13.4, "Editing and Viewing Entitlements," on page 316
- Section 13.5, "Managing Entitlements," on page 321
- Section 13.6, "Creating Entitlement Policies," on page 321

13.1 How Entitlements Work

The following diagram shows the basic entitlement process.

Figure 13-1 *Basic Overview of Entitlements*



1. An entitlement agent grants an entitlement to a user. There are three ways that entitlements are granted to a user:

   - **Role-Based Entitlements:** The Entitlements Service driver grants the entitlement based on criteria that places the user in a particular role (or group). This criteria can be based on any event that occurs in the Identity Vault. For example, adding a new employee in an HR system causes a User object to be created in the Identity Vault. Creation of the new User object is the criterion that causes the Entitlements Service driver to grant the Active Directory User Account entitlement to the user.

     To create role-based entitlements in Designer, see Section 13.3, “Creating Entitlements through the Entitlement Wizard,” on page 304.

   - **User Application Role Based Provisioning:** The user receives a role assignment through the User Application. The User Application’s Role Service driver grants the user any entitlements associated with the new role. For example, a user is assigned an Accountant role that requires access to the Accounting group in Active Directory. The Role Service driver grants the Active Directory Group Membership entitlement to the user.

     To create entitlements for role based provisioning, use the Role editor. See “Specifying Entitlements” under Using the Role Editor (http://www.novell.com/documentation/idmrbpm361/dgpro/data/roles_roleseditor.html#roles_entitlements).

   - **User Application Workflow-Based Provisioning** A provisioning workflow grants the entitlement to the user. For example, a new employee is added to the HR system, which causes a User object to be created in the Identity Vault. Creation of the new User object initiates a workflow that grants the Active Directory User Account entitlement to the user.

     Creating entitlements to use with workflow-based provisioning is an involved process. To get you started, see Configuring Provisioning Request Definitions (http://www.novell.com/documentation/idmrbpm361/dgpro/data/prdefworkingchapter.html)

2. When an entitlement is added to or removed from a user’s DirXML-EntitlementRef attribute, any entitlement-enabled drivers begin to process the event. Only drivers that have the DirXML-EntitlementRef attribute added to their Subscriber channel filter monitor users for entitlement changes.

3. The driver processes the entitlement event against the Subscriber channel policies. If the entitlement event is for an entitlement that applies to the driver, the policies are processed. Otherwise, no processing occurs. In the above diagram, the Grant User Account policy is processed because 1) the Active Directory User Account entitlement was added to the user’s

DirXML-EntitlementRef attribute and 2) the User Account entitlement is defined on the Active Directory driver. Likewise, if the Active Directory User Account entitlement is later removed from the user's DirXML-EntitlementRef attribute, the Revoke User Account policy is processed.

4. The policies trigger the granting or revoking of access to the entitled resource. In the above diagram, the Grant User Account policy triggers the creation of a user account in Active Directory.

13.2 Designing Entitlements

You must know beforehand what you want to accomplish with entitlements. Entitlements work from the functionality you build into Identity Manager drivers through policies. These driver policies implement rules and process the events between the Identity Vault and the connected system. If the policies in the Identity Manager driver do not specify what you want to do, entitlements cannot work. For example, if you don't specify the action section of the Check User Modify for Group Membership rule in the Command policy, attempts to grant or revoke a group membership entitlement are ignored.

When you know what you want to accomplish with Identity Manager, you can correctly design granting and revoking capabilities for any connected system resources. The following four-step procedure can help you plan to create and use entitlements:

1. Know what you want to accomplish in your business situation. You can design and implement many business solutions through Identity Manager, but you need to know what you want to do before implementing something that isn't defined. Make a numbered list of what you want to do.
2. Define an entitlement that represents one item from your numbered list. You can create valueless and valued entitlements. Valued entitlements can get their values from an external query, they can be administrator-defined, or they can be free form. There are examples in Section 13.3, "Creating Entitlements through the Entitlement Wizard," on page 304.
3. Add policies to the Identity Manager Driver to implement the designed entitlement. To create a policy for an Identity Manager driver, you need to be conversant with XSLT or DirXML script to define the way the connected system handles and receives information, and the way Novell eDirectory stores information. Unless you are a good DirXML programmer, this is a job for consultants.
4. Set up a managing agent to grant or revoke the entitlement. If you want an automated process, use Role-Based Entitlements; if you want a manual process, use the User Application's workflow-based provisioning feature. See "Installation" in the *Identity Manager 3.6.1 Installation Guide*. As you plan your entitlements, use the following sections for more information.

- Section 13.2.1, "Terminology," on page 302
- Section 13.2.2, "Entitlement Prerequisites," on page 302
- Section 13.2.3, "Identity Manager Drivers with Preconfigurations that Support Entitlements," on page 302
- Section 13.2.4, "Enabling Entitlements On Identity Manager Drivers," on page 303

13.2.1 Terminology

Following are some terms that are used throughout this section.

Table 13-1 *Terminology*

Terms	Explanation
Entitlement	An Identity Vault object that represents a business resource in a connected system.
Entitlements Service driver	Grants and revokes entitlements. For Role-Based Entitlements, the agent is the Entitlements Service driver, which must be initiated for entitlements to work.
Grant or revoke	Granting or revoking an entitlement is controlled by Global Configuration Variables (GCVs) on an Identity Manager driver.
Entitlement consumer	Anything that uses entitlement-related information. Entitlement consumers include iManager, the User Application, and Identity Manager policies.

13.2.2 Entitlement Prerequisites

- ❑ eDirectory 8.7.3.3 or later with the latest Support Pack (SP8 or later)
- ❑ Identity Manager 2 or 3
- ❑ An Entitlements Service driver

  You must have an Entitlements Service driver in each driver set where you want to use entitlements. This requires a very simple, two-step setup for each driver set. To do this, see Creating Entitlements (http://www.novell.com/documentation/idm36/idm_entitlements/data/be4roxr.html) in the *Identity Manager 3.6 Entitlements Guide*.
- ❑ A driver configuration that supports entitlements

  Before you can use entitlements with a connected system, do one of the following:

  - Import the Identity Manager driver configuration for the driver and specify that the driver has entitlements enabled.
  - Enable your driver to support entitlements. To do this:
    1. Create entitlements using Designer.
    2. Add the DirXML-EntitlementRef attribute to your driver filter as described in Section 13.2.4, “Enabling Entitlements On Identity Manager Drivers,” on page 303.
    3. Write policies to implement the entitlements you create in Step 1 under Section 13.2, “Designing Entitlements,” on page 301.

13.2.3 Identity Manager Drivers with Preconfigurations that Support Entitlements

The following drivers include configuration files that already contain entitlements and the policies required to implement the entitlements. These entitlements support the most common scenarios: granting and revoking user accounts, groups, and email distribution lists.

- Active Directory: Grant and revoke accounts, group membership, Exchange Mailbox

- GroupWise: Grant and revoke accounts, grant and revoke members of distribution lists
- LDAP: Grant and revoke user accounts
- Linux and UNIX: Grant and revoke accounts
- Lotus Notes: Grant and revoke user accounts and group memberships
- RACF: Grant and revoke group accounts and group memberships

These are example entitlements and policies that you can use as is if they meet your needs. If not, you can modify them to meet your needs, or you can use them as examples as you implement additional entitlements.

13.2.4 Enabling Entitlements On Identity Manager Drivers

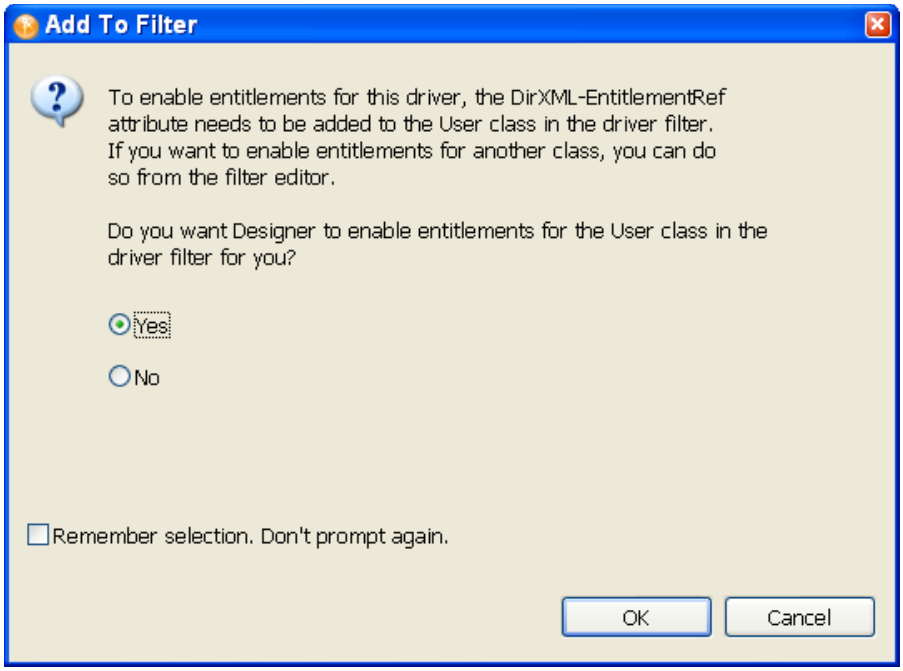
Before you can use entitlements, you must first ensure that your driver has entitlements enabled. You can do this through the Entitlements Wizard as you finish creating entitlements; this applies to both preconfigured and non-preconfigured drivers.

However, if you want to use the preconfigured driver's entitlements and the infrastructure that supports them, you must enable entitlements when you initially create a driver in Designer or iManager; the preconfigured policies and rules that support the preconfigured entitlements cannot be added later without re-creating the driver. If you import a driver that has entitlements enabled into Designer from an Identity Vault, the imported driver also has entitlements enabled. If you deploy a driver that has entitlements enabled into an Identity Vault, the deployed driver also has entitlements enabled.

You can see if your preconfigured drivers have entitlements enabled by clicking the Outline view, then clicking the Subscriber channel of your selected driver. If entitlements are enabled, you should see the preconfigured entitlements appear under the Subscriber Channel. If entitlements do not appear under the Subscriber Channel in the Outline view, entitlements were not enabled when the driver was initially installed.

However, you can still use entitlements on preconfigured Identity Manager drivers that do not have entitlement preconfigurations enabled. To do this, run the Entitlement Wizard. The last page in the Entitlement Wizard asks if you want to add the DirXML-EntitlementRef attribute to the driver filter, with *Yes* selected. Click *OK*. However, because the policies and rules are not in place on the driver, you won't be able to use their preconfigured entitlements without adding those supporting policies and rules yourself.

Figure 13-2 *Enabling Entitlements*



You can also use entitlements on Identity Manager drivers that do not contain entitlement preconfigurations. To enable your driver to support entitlements, add the DirXML-EntitlementRef attribute to your driver filter. Run the Entitlement Wizard as described above to add the DirXML-EntitlementRef attribute to the driver filter.

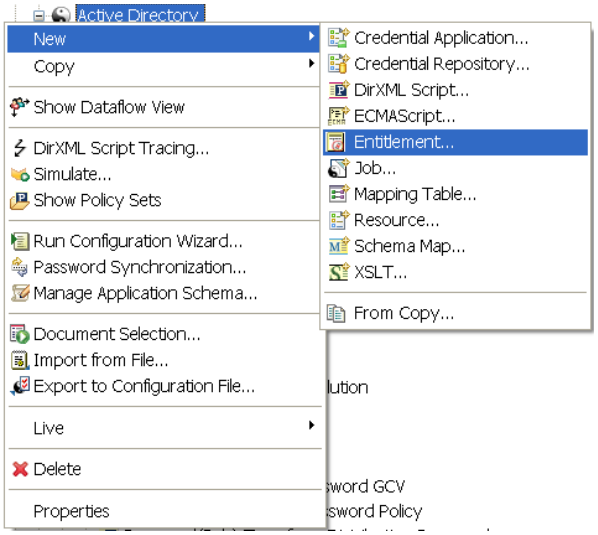
13.3 Creating Entitlements through the Entitlement Wizard

Designer comes with an Entitlement Wizard. This wizard steps you through the creation of entitlements by asking a series of questions about how the entitlement will be used in the enterprise. Use one of the following methods to access the Entitlement Wizard:

From the Outline View

To access the Entitlement Wizard from the *Outline view*, right-click a Driver object, then click *New > Entitlement*.

Figure 13-3 *Adding an Entitlement through the Outline View*



From the Modeler View

To access the Entitlement Wizard from the *Modeler view*, right-click the driver icon, then click *New > Entitlement*.

There are two types of entitlements that you can create: valueless and valued. The type you use depends on whether you need to pass additional information to the policies.

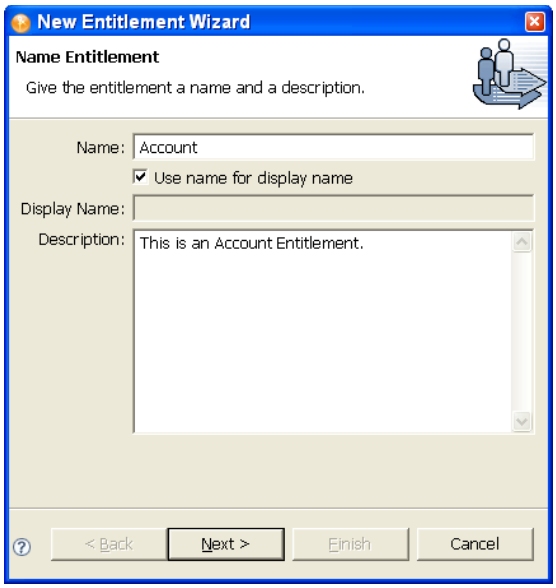
- Section 13.3.1, “Valueless Entitlements,” on page 305
- Section 13.3.2, “Valued Entitlement that Queries an External Application,” on page 307
- Section 13.3.3, “Administrator-Defined Entitlements with Lists,” on page 312
- Section 13.3.4, “Administrator-Defined Entitlements without Lists,” on page 314

13.3.1 Valueless Entitlements

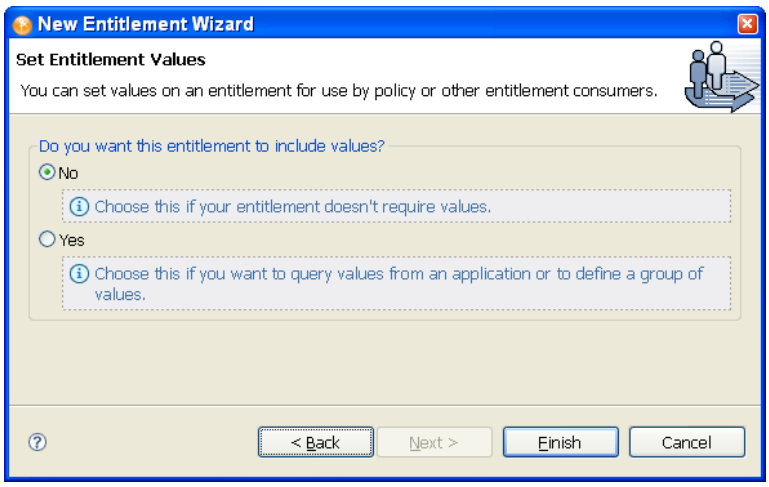
A valueless entitlement has no values to go with it. An example is the Account Entitlement for Active Directory, which is used to turn on account capabilities. You use valueless entitlements if you don’t need to pass any extra information to driver policies.

To create a valueless entitlement:

1. Right-click the driver icon in the *Outline view* or in the *Modeler view*, then click *New > Entitlement*.



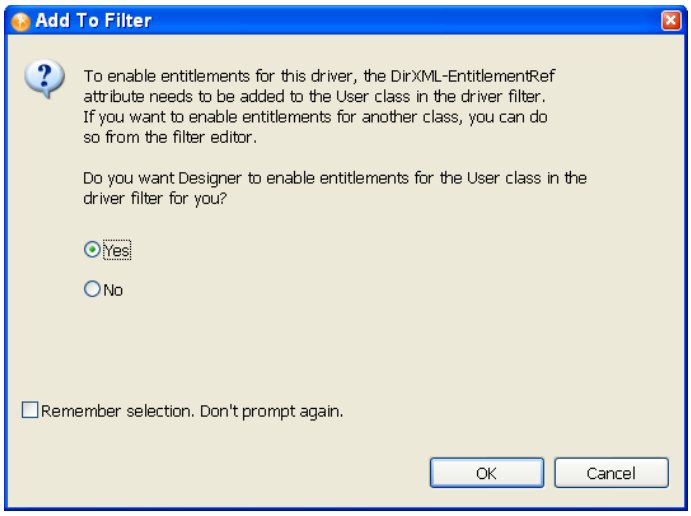
2 Type the name and description information. For this example, the entitlement is named Account, with a description of “This is an Account Entitlement.” Click *Next*.



3 Because this first example is valueless, select *No* to *Do you want this entitlement to include values?*

4 Click *Finish*.

5 In the Add To Filter dialog box, answer *Yes* if you want the driver to listen for this entitlement. This enables entitlements for the driver.



The DirXML-EntitlementRef attribute allows the driver filter to listen for entitlement activities. This is necessary in order to use the entitlements you are creating.

If you don't want to see the Add To Filter window on every entitlement you are creating for any driver in Designer, select *Remember Selection - Don't Prompt Again*, then click *OK*. However, after the attribute is added to this driver filter, you won't see the Add To Filter window again.

If you have a file conflict, you are asked to save the editor's changes before continuing. Once the editor is saved, the entitlement displays in the Modeler view.

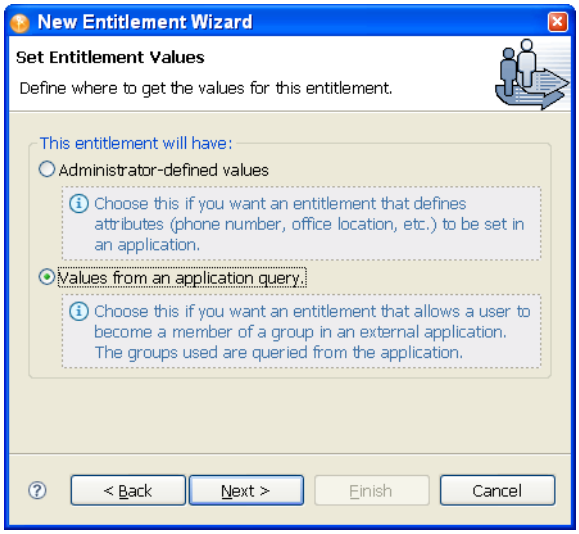
13.3.2 Valued Entitlement that Queries an External Application

Values are a way of passing data that you might need to use in policies. Valued entitlements can get their values from an external query; they can be administrator-defined, or they can be free-form.

1. Right-click the driver icon in the *Outline view* or in the *Modeler view*, then click *New* > *Entitlement*.
2. Give the entitlement a name. This example uses Application Query, with the *Use Name for Display Name* option selected. In the Description box, type `Looks for the Class name of Groupx`, then click *Next*.

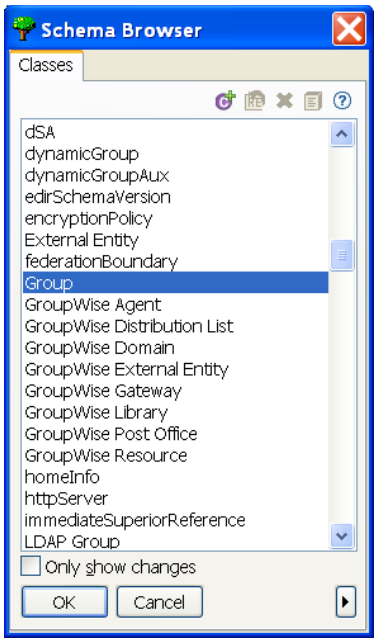
3 On the Set Entitlements Values page, select *Yes* so you can query values from an application or define a group of values, then click *Next*.

4 The next Set Entitlements Values page allows you to define where you get the values for this entitlement. Valued entitlements can get their values from an external query, or they can be administrator-defined. For this example, select the *Values from an application query* option, then click *Next*.



The Define Application Query window combines two steps: defining the query and mapping the query results.

5 To fill in the Class Query, click the *Schema Browser* button on the right side of the Class entry.



6 The Schema Browser shows you the Classes in the eDirectory namespace that are available. If you know the name of the Class type you want to query, click to select a selection in the *Classes* tab, then start typing the Class name. The browser jumps to the alphabetical order of what you type. Select the Class name, then click *OK*.

7 Type the base distinguished name (DN) and the scope. For this example, select the Class Group, at the Base Distinguished Name of Blanston, with the Scope of subtree (choices are subtree, entry, and subordinates).

This example maps the query results from the connected system to certain values that entitlement consumers can use. At present, the consumers are iManager managing Role-Based Entitlement policies and the User Application managing workflow-based provisioning entitlements. The *Value From Query* information pre-populates the consumer's user interface with the following:

- **Display Name:** Defines the attribute that displays in the list of values. The example selects Source Distinguished Name for the display name. Click the drop-down button on the *Display Name shown to entitlement consumers* list to see a list of Attributes associated with the class you selected through the Schema Browser. The list includes both the Attributes and the Inherited Attributes for the selected class.
- **Description:** Defines the attribute that displays as a description for that value. For the description, select *Description* from the Value drop-down list to map the query results from the connected system to the entitlement.
- **Value:** Defines the attribute or token that is the actual value. The *Value* entry is not seen in the entitlement consumer, but it is the value that is assigned when the entitlement is granted or revoked. In this case, choose *Association*.

If you do not use the *Schema Browser* button when selecting the class, you see only two selections in the *Value From Query* lists: *Association* and *Source Distinguished Name*. If these attributes suit your needs, use them. You can also type the attribute name into the text field. However, if you want to select the attributes from the lists, use the *Schema Browser* button when selecting a class for the query. You see the attributes and inherited attributes for the selected class.

8 When the values are filled out, select *Next*.

9 In the Assign Multiple Values window, select *Yes* if you want the entitlements to be granted more than once and with different values. If you select *No*, the entitlement can only be granted once. For this example, click *Yes*, then click *Next*.

It makes sense to assign group entitlements with multiple values, but it would not make sense to assign an account entitlement more than once.

10 You are asked if this entitlement is intended to be used by Role-Based Entitlement policies through iManager. If you want this entitlement to be granted or revoked automatically, select *Yes* to the Role-Based Entitlements question, click *Next*, then continue with Step 11.

or

If you want the granting or revoking of this entitlement to be a manual process (approved by someone), select *No* to use the User Application, then skip to Step 12 on page 311.

We recommend that you have only one agent control an entitlement. If multiple agents are in control, you have the following consequences:

- Whoever comes last controls the entitlement results
- Results are unpredictable
- Using both agents to control an entitlement is not supported by Novell

11 (Conditional) If you select *Yes* to the Role-Based Entitlements question, you are asked if you want to use the Role-Based Entitlements priority to resolve any conflicts that might happen when this entitlement is assigned more than once with different values. You can resolve the conflict by either using Role-Based Entitlements priority, or by merging the values.

Merging the values merges the entitlements of all involved Role-Based Entitlement policies, so if one policy revokes an entitlement but another policy grants an entitlement, the entitlement is eventually granted. Solving conflicts by priority works if you need to ensure that only one policy is applied to this entitlement at any time. This example uses priority.

12 Click *Finish*.

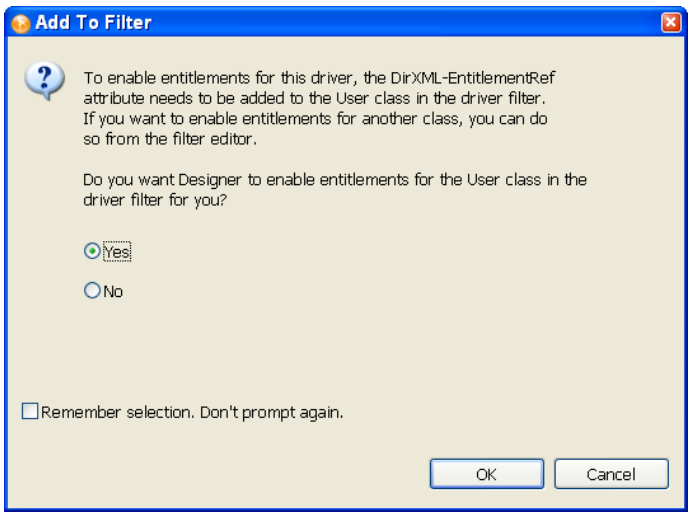
For this example, the query values look for the Source Distinguished Name attribute of the Class name of Group, starting from the Base DN (Blanston) and checking through the subtree from that beginning point. The values that come back from the query are similar to the following:

```
<instance class-name="Group" src-dn="o=Blanston,cn=group1">
   <association>o=Blanston,cn=group1</association>
   <attr attr-name="Description"> the description for group1</attr>
</instance>
<instance class-name="Group" src-dn="o=Blanston,cn=group2">
   <association>o=Blanston,cn=group2</association>
   <attr attr-name="Description"> the description for group2</attr>
</instance>
<instance class-name="Group" src-dn="o=Blanston,cn=group3">
   <association>o=Blanston, cn=group3</association>
   <attr attr-name="Description"> the description for group3</attr>
</instance>
<!-- ... ->
```

The information received from the query fills in the various fields. For instance, the <display-name> field receives o=Blanston,cn=group1. The <description> field receives the description for Group1, and the <ent-value> field receives o=Blanston,cn=group1. Because more than one group exists and meets the query criteria, this information is also collected and shown as other instances of the query.

The association format value is unique for every external system, so the format and syntax are different for each external system queried.

13 In the Add To Filter window, click *Yes* if you want the driver to listen for this entitlement. This enables entitlements for the driver.



The DirXML-EntitlementRef attribute allows the driver filter to listen for entitlement activities. This is necessary in order to use the entitlements you are creating.

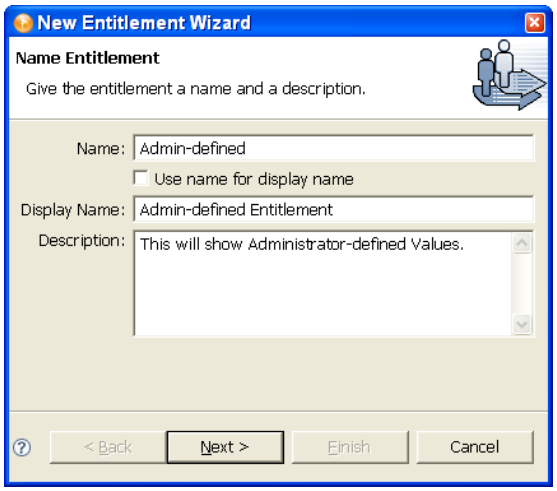
If you don't want to see the Add To Filter window on entitlements you are creating for any driver in Designer, select *Remember Selection - Don't Prompt Again*, then click *OK*. However, after the attribute is added to this driver filter, you won't see the Add To Filter window again.

If you have a file conflict, you are asked to save the editor's changes before continuing. Once the editor is saved, the entitlement displays in the Modeler view.

13.3.3 Administrator-Defined Entitlements with Lists

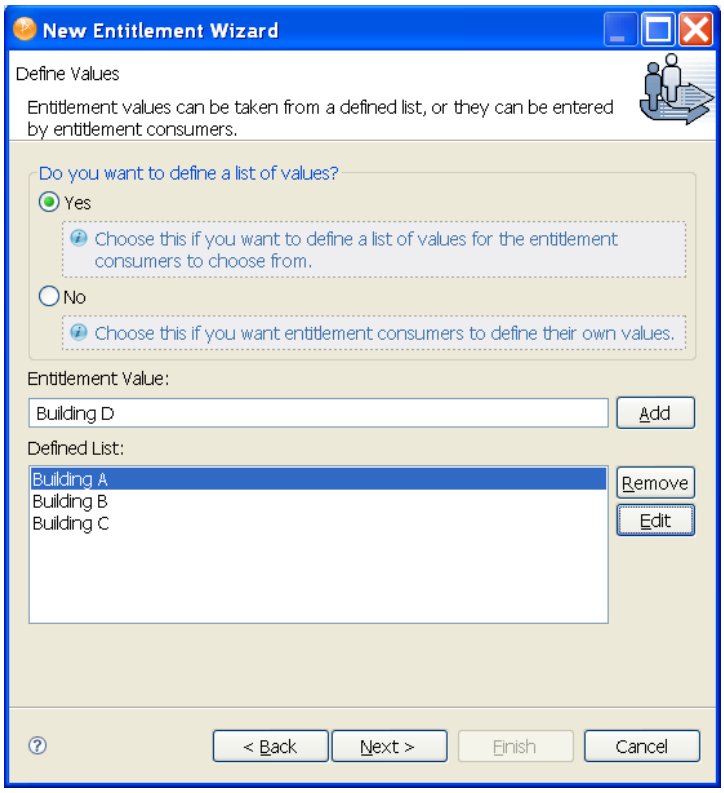
The example in the following procedure is an administrator-defined entitlement that allows you to select a listed entry. This type of entitlement is best used through Workflow entitlements rather than Role-Based Entitlements.

1. Right-click the driver icon in the Outline view or the Modeler view, then click *New* > *Entitlement*.



   In this example, the entitlement name is Admin-defined, but the defined display name is Admin-defined Entitlement. You need to define a display name only if you want the display name to be different from the name you called the entitlement; otherwise, you can just use the entitlement name as the display name. In this example, the *Description* field is defined as *This will show Administrator-defined Values*.

2. Click *Next*.
3. In the Set Entitlement Values window, select *Yes* to the question "Do you want this entitlement to include values?" Click *Next*.
4. In the next Set Entitlement Values window, select *Administrator Defined Values*, then click *Next*.
5. In the Define Values window, type the values you want to add to the Entitlement Value entry, click *Add* to add the value to the Defined List pane, then click *Next*.



In this example, the values are corporate buildings: Building A through Building D. Through an entitlement client, such as an iManager Role-Based Entitlement task or through the user application, users or defined-task managers can specify the building information, which is then included in an external application, such as Novell eDirectory.

Use the *Remove* button to remove a value, or use the *Edit* button to edit a value.

6 In the Assign Multiple Values window, select *Yes* if you want the entitlements to be granted more than once and with different values. If you select *No*, the entitlement can only be granted once. For the example, click *No*, then click *Next*.

It makes sense to assign group entitlements with multiple values, but it does not make sense to assign building letters more than once.

7 You are asked if this entitlement is intended to be used by Role-Based Entitlement policies through iManager. If you want this entitlement to be granted or revoked automatically, select *Yes* to the Role-Based Entitlements question, click *Next*, then continue with Step 8 on page 314.

or

If you want the granting or revoking of this entitlement to be a manual process (approved by someone), select *No* to use the User Application, then skip to Step 9 on page 314.

We recommend that you have only one agent control an entitlement. If multiple agents are in control, you have the following consequences:

- Whoever comes last controls the entitlement results
- Results are unpredictable
- Using both agents to control an entitlement is not supported by Novell

8 (Conditional) If you select *Yes* to the Role-Based Entitlements question, you are asked if you want to use the Role-Based Entitlements priority to resolve any conflicts that might happen when this entitlement is assigned by different Role-Based Entitlement Policies with different values. You can resolve the conflict by either using the Role-Based Entitlements priority, or by merging the values. This example merges the values.

   Merging the values merges the entitlements of all involved Role-Based Entitlement policies, so if one policy revokes an entitlement but another policy grants an entitlement, the entitlement is eventually granted. Solving conflicts by priority works if you need to ensure that only one policy is applied to this entitlement at any time.

9 Click *Finish*.

10 If you see the Add To Filter window, answer *Yes* if you want the driver to listen for this entitlement. This enables entitlements for the driver. The DirXML-EntitlementRef attribute allows the driver filter to listen for entitlement activities, which is necessary in order to use the entitlements you are creating.

   Or

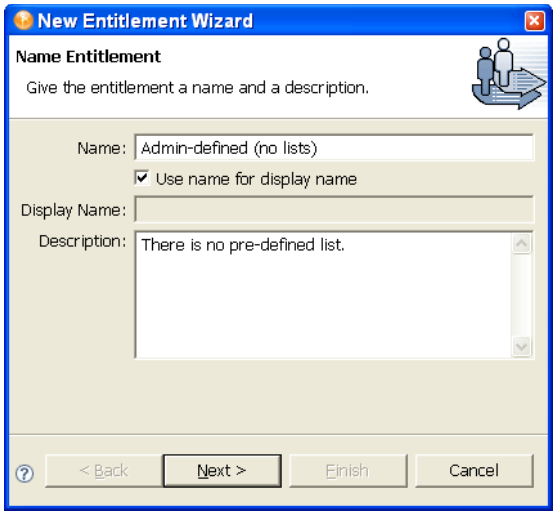
   If you don't want to see the Add To Filter window on entitlements you are creating for any driver in Designer, select *Remember Selection - Don't Prompt Again*, then click *OK*. However, after the attribute is added to this driver filter, you won't see the Add To Filter window again.

Before you can edit this entitlement, you are asked to save the editor's changes before continuing. Once the editor is saved, the entitlement displays in the Modeler view.

13.3.4 Administrator-Defined Entitlements without Lists

The example in the following procedure is an administrator-defined entitlement that forces the administrator to type a value. You can use this kind of entitlement if you do not have all of the information at the initial setup, so you cannot create a task list.

1 Right-click the driver icon in the Outline view or the Modeler view, then click *New* > *Entitlement*.



   In this example, the entitlement name is *Admin-defined (no lists)*, and it uses the entitlement name as the displayed name because the *Use Name For Display Name* option is selected.

2. Click *Next*.
3. Select *Yes* on the Set Entitlement Values page, then click *Next*.
4. Select *Administrator Defined Values* on the second Set Entitlement Values page, then click *Next*.
5. Select *No* to the question “Do you want to define a list of values?” on the Define Values page, then click *Next*.

   Selecting this option allows the administrator or users to type in a value. Using this option can be risky however since wrong or mispelled information can cause the vaule to be incorrect and the action in the entitlement to fail.
6. Select *No* to the question “Allow this entitlement to be assigned multiple times with different values?” on the Assign Multiple Values page, then click *Next*.
7. You are asked if this entitlement is intended to be used by Role-Based Entitlement policies through iManager. If you want this entitlement to be granted or revoked automatically, select *Yes* to the Role-Based Entitlements question, click *Next*, then continue with Step 8 on page 315.

   or

   If you want the granting or revoking of this entitlement to be a manual process (approved by someone), select *No* to use the User Application, then skip to Step 9 on page 315.

   We recommend that you have only one agent control an entitlement. If multiple agents are in control, you have the following consequences:

   - Whoever comes last controls the entitlement results
   - Results are unpredictable
   - Using both agents to control an entitlement is not supported by Novell
8. (Conditional) If you select *Yes* to the Role-Based Entitlements question, you are asked if you want to use the Role-Based Entitlements priority to resolve any conflicts that might happen when this entitlement is assigned by different Role-Based Entitlement Policies with different values. You can resolve the conflict by either using the Role-Based Entitlements priority, or by merging the values. This example uses priority.

   Merging the values merges the entitlements of all involved Role-Based Entitlement policies, so if one policy revokes an entitlement but another policy grants an entitlement, the entitlement is eventually granted. Solving conflicts by priority works if you need to ensure that only one policy is applied to this entitlement at any time.
9. Click *Finish*.
10. If you see the Add To Filter window, answer *Yes* if you want the driver to listen for this entitlement. This enables entitlements for the driver. The DirXML-EntitlementRef attribute allows the driver filter to listen for entitlement activities, which is necessary in order to use the entitlements you are creating.

    or

    If you don’t want to see the Add To Filter window on entitlements you are creating for any driver in Designer, select *Remember Selection - Don’t Prompt Again*, then click *OK*. However, after the attribute is added to this driver filter, you won’t see the Add To Filter window again.

Before you can edit this entitlement, you are asked to save the editor’s changes before continuing. Once the editor is saved, the entitlement displays in the Modeler view.

13.4 Editing and Viewing Entitlements

After you have created entitlements, you might need to edit them. You can also use the Edit mode to see the entitlements in their XML source code.

- Section 13.4.1, “Entitlement XML Source and XML Tree Views,” on page 317
- Section 13.4.2, “Using the Novell Entitlement DTD,” on page 320

To edit an entitlement:

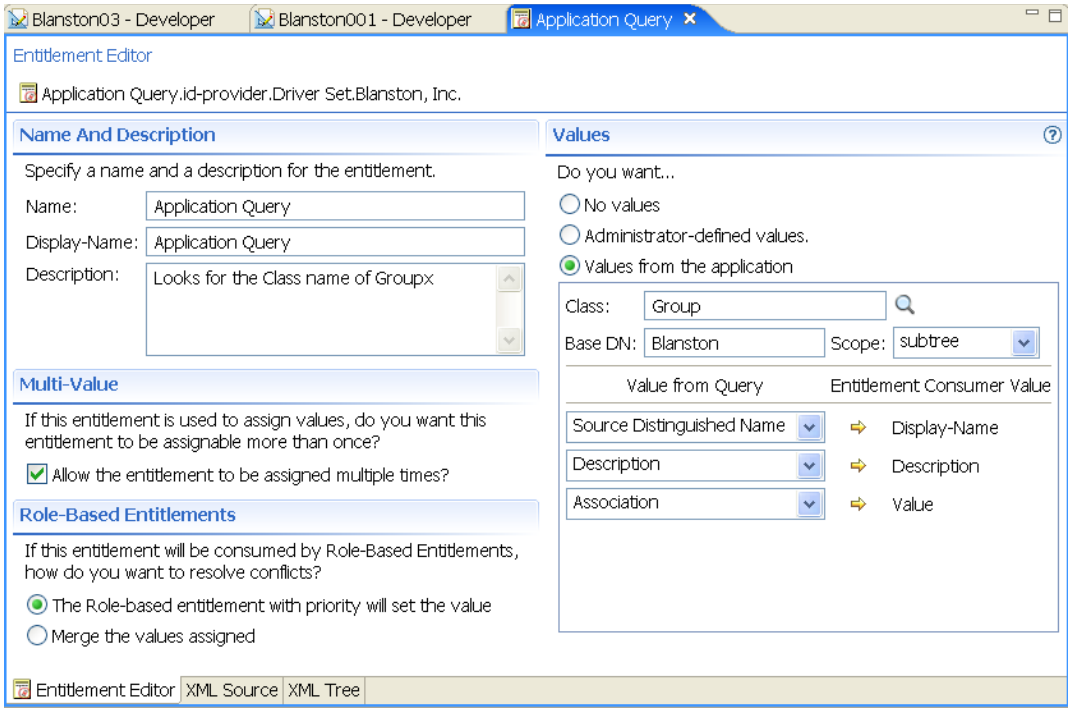
1. From the Outline view, right-click an entitlement that appears under the Subscriber channel of the selected driver, then click *Edit*.

   or

   Double-click the entitlement icon to bring up the entitlement in the Entitlement Editor.

   You can also right-click the driver icon in the Modeler view, then select *Edit Entitlements*.

2. If you have more than one entitlement for the selected driver, you see the Edit Entitlements windows listing the available entitlements. Select an entitlement, then click *OK*. The entitlement appears in the Entitlement Editor.



The Entitlement Editor view shows you all of the pages and choices that you see in the Entitlement Wizard, but the information is on one page.

- **Entitlement Editor:** Displays the full DN name for the entitlement. If there is a conflict with the entitlement name or some other error, you see a red icon to the left of the Entitlement Editor name, followed by an error message.
- **Name and Description:** Allows you to edit the name, the display name, and the description that you have given to this entitlement.

- **Multi-Value:** Allows you to select if you want an entitlement to be assigned multiple times.
- **Role-Based Entitlements:** Allows you to select conflict resolution for Role-Based Entitlements. If you do not select Role-Based Entitlements, *the Role-based entitlements with priority* button is the default.
- **Values:** Allows you to define how values are defined: no values, administrator defined values, or values from an application.

The information that appears in the Entitlement Editor depends on what you initially defined in the entitlement. If you choose to edit a valueless entitlement, the *Values* heading displays *No Values*. If you are editing a valued entitlement and you want to add values to a list, type the value in the *Value* field and click *Add*. If you want to remove a value, select the value in the *Values* list and click *Remove*.

If you don't want to select from a list, select *Administrator Defined Values* under the *Values* heading and leave the *Values* list blank. This gives you a blank text box in iManager or in the user application, and you can fill in the value there.

3 When you have made your changes to the entitlement, click the *Save* icon in the upper left corner of Designer, or click the *X* on the entitlement's tab to display a Save Resource window, allowing you to save changes (*Yes/No/Cancel*).

13.4.1 Entitlement XML Source and XML Tree Views

To view the entitlement in XML source code:

1 From the Outline view, right-click an entitlement that appears under the Subscriber channel of the selected driver, then click *Edit*.

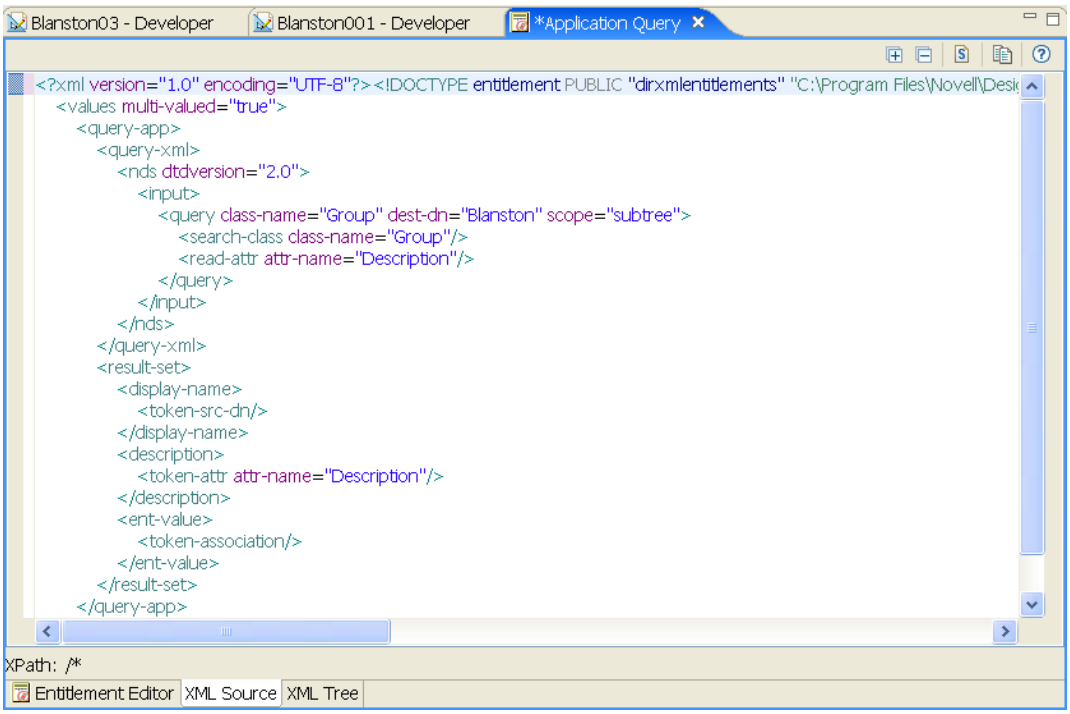
or

Double-click the entitlement icon to bring up the entitlement in the Entitlement Editor.

You can also right-click the driver icon in the Modeler view, then select *Edit Entitlements*.

2 To see the XML Source view, click *XML Source* at the bottom of the Entitlement Editor view.

The XML Source view shows the XML code in a formatted state.



The upper right corner of the XML Source view has the following selections:

Name	Description
Expand All	Allows you to see all items under the item that you have selected.
Collapse All	Allows you to collapse all items that you have selected.
Attach XML Catalog Entry, XML Schema, or DTD	Allows you to attach an XML Catalog entry, an XML schema file, or a DTD (Document Type Definition) file. For default Windows installation, the DTD for entitlements is found under `C:\Program Files\Novell\Designer\eclipse\plugins\com.novell.designer.idm.entitlements_1.1.0\DTD\dirxmlentitlements.dtd`.
Copy XML to Clipboard	Allows you to copy highlighted XML code to the clipboard. This action removes the DOCTYPE element.
Find/Replace (Ctrl+F)	Ctrl+F brings up the Find/Replace window, which allows you to query text, structure, and XPath searches in a forward or a backward direction. Other options include case sensitive, wrap search, whole word, incremental, and regular expressions search capabilities.
Help	Opens the Help view to the right of the XML Source view.

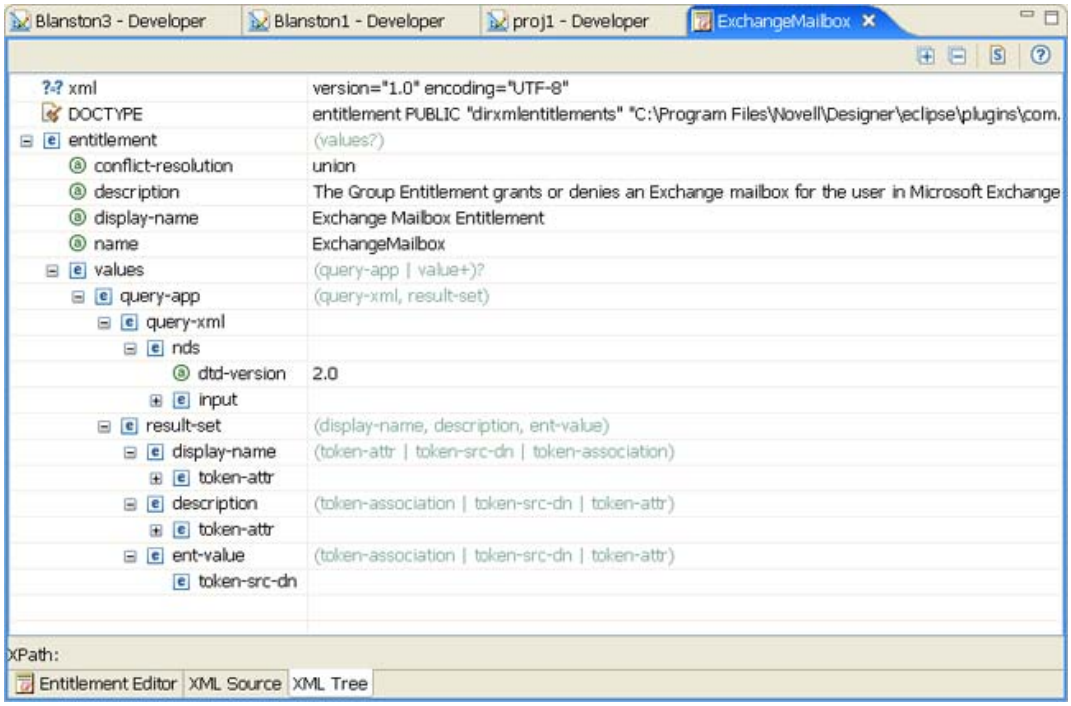
Right-clicking in the XML Source view brings up the following options:

- Undo Text Change (Ctrl+Z)
- Revert File
- Save
- Cut (Ctrl+X)

- Copy (Ctrl+C)
- Paste (Ctrl+V)
- Format the document or active elements
- Clear Validation Errors
- Validate
- Preferences

3 To see the XML Tree view, click *XML Tree* at the bottom of the Entitlement Editor view.

The XML Tree view is a tree control view of the XML source code. You can perform the same edits in this view as you can in the Entitlement Editor view or the XML Source view. To view the entitlement in XML Tree view, select *XML Tree* at the bottom of the Entitlement Editor view.



The upper right corner of the XML Tree view menu contains the following selections:

Name	Description
Expand All	Allows you to see all items under the item that you have selected.
Collapse All	Allows you to collapse all items that you have selected.
Attach XML Catalog Entry, XML Schema, or DTD	Allows you to attach an XML Catalog entry, an XML schema file, or a DTD (Document Type Definition) file. For default Windows installation, the DTD for entitlements is found under `C:\Program Files\Novell\Designer\eclipse\plugins\com.novell.designer.idm.entitlements_1.1.0\DTD\dirxmlentitlements.dtd`.

Name	Description
Find/Replace (Ctrl+F)	Brings up the Find/Replace window, which allows you to query text, structure, and XPath searches in a forward or a backward direction. Other options include case sensitive, wrap search, whole word, incremental, and regular expressions search capabilities.
Help	Opens the Help view to the right of the XML Tree view.

Right-clicking in the XML Tree view can bring up a number of different options. For example, right-clicking the highlighted value on the right side presents the following options:

- Undo
- Cut
- Copy
- Paste
- Delete
- Select All

Right-clicking an attribute on the left side in the XML Tree view presents the following options:

- Remove
- Edit the Selected Attribute
- Replace with a value

Depending on what you select on the left side in the XML Tree view, you see different options. For example, right-clicking an element presents the following options:

- Remove Element
- Add New Attribute
- Add to a Child Element a Comment, a Processing Instruction, a PCDATA, a CDATA Section, a new Element
- Add Before a Comment, a Processing Instruction, a PCDATA, a CDATA Section, a new Element
- Add After a Comment, a Processing Instruction, a PCDATA, a CDATA Section, a new Element

13.4.2 Using the Novell Entitlement DTD

Some entitlements come predefined on drivers that have entitlements enabled. (For a list of these drivers with predefined entitlements see Section 13.2.3, “Identity Manager Drivers with Preconfigurations that Support Entitlements,” on page 302.) You can use these entitlements or you can create your own entitlements in iManager or Designer. To help you create your own entitlements, you can use the Novell Entitlement DTD as an example to create entitlements. For an example of the Novell Entitlement DTD and an explanation of its functionality, see the Writing Entitlements in XML (http://www.novell.com/documentation/idm36/idm_entitlements/data/be4rqan.html) section of the *Identity Manager 3.6 Entitlements Guide*.

13.5 Managing Entitlements

After you create entitlements (or use entitlements that come preconfigured with certain Identity Manager drivers), you need to manage them. Entitlements are tied into the eDirectory event system and granting and revoking are initiated through two agents:

- iManager through Role-Based entitlement policies
- The User Application as workflow entitlements

Role-Based Entitlements allow you to automatically grant or revoke business resources if the criteria are met. In order for workflow entitlements to work with the User Application, manual approval is first required.

For instance, you can specify that if user has A, B, and C, then the user is made a member of Group H; but if the user has E and F qualifications, he or she is made a member of Group I. Through Role-Based Entitlements, this action is done automatically, as long as the conditions are met. In order for this entitlement to work with workflow entitlements, the User object must first acquire approval, which you need to set up through the User Application. However, if you do not add to the driver the policies and rules to interpret the event in the designated system, granting and revoking entitlements has no effect.

Use either Role-Based Entitlements or workflow entitlements; it is a not good idea to mix them to manage the same resource. We recommend that you have only one agent control an entitlement. If multiple agents are in control, you have the following consequences:

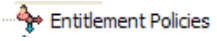
- Whoever comes last controls the entitlement results
- Results are unpredictable
- Using both agents to control an entitlement is not supported by Novell

13.6 Creating Entitlement Policies

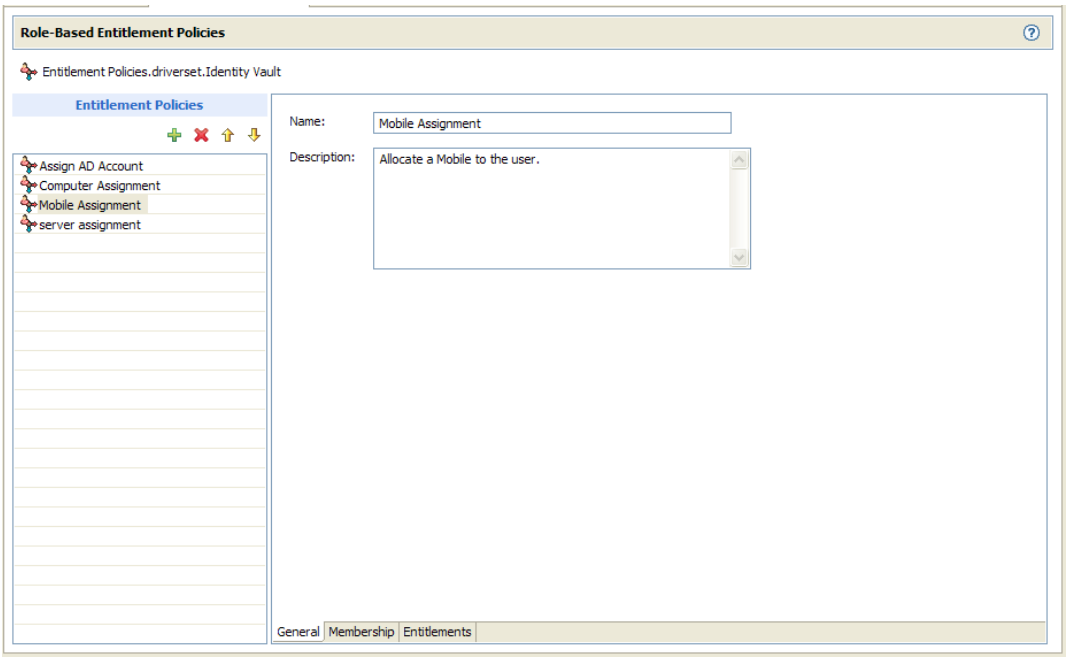
The Entitlements Service driver implements entitlements through the use of entitlement policies. An entitlement policy contains the following:

- **General:** The name and description of the entitlement policy.
- **Membership:** The list of users assigned to the policy. A user can be dynamically assigned to the policy when he or she meets the criteria for the policy, or the user can be statically (manually) assigned to the policy.
- **Entitlements:** The list of entitlements associated with the policy. Users assigned to the policy receive all of the entitlements associated with the policy. If the user is removed from the policy, he or she loses all entitlements associated with the policy.

To create an entitlement policy:

1. In Designer, right-click the *Driver Set* > *New* > *Role-based Entitlements Policies*.
2. When you double-click the 

, it displays the editor to configure policies.



3 On the left side, click on the '+' icon to add the entitlement policy. The editor displays three tabs- General, Membership, and Entitlements.

This list displays all entitlement policies that have been created for the driver set. If you are using Role-Based Entitlements for the first time, no policies are listed. The order of the policies in the list assumes importance during the conflict resolution of the entitlement, especially if the conflict-resolution is 'priority'.

General page contains the following fields:

Entitlement Policy Name: Provide a name that indicates the purpose of the entitlement. The name must be unique within the driver set and cannot include more than 64 characters.

Description: Provide any additional information you want to identity the policy.

4 Membership page allows you to configure static and dynamic membership.

4a Static membership lets you include users who don't meet the dynamic membership criteria or exclude users who meet the criteria but should not be members of the policy.

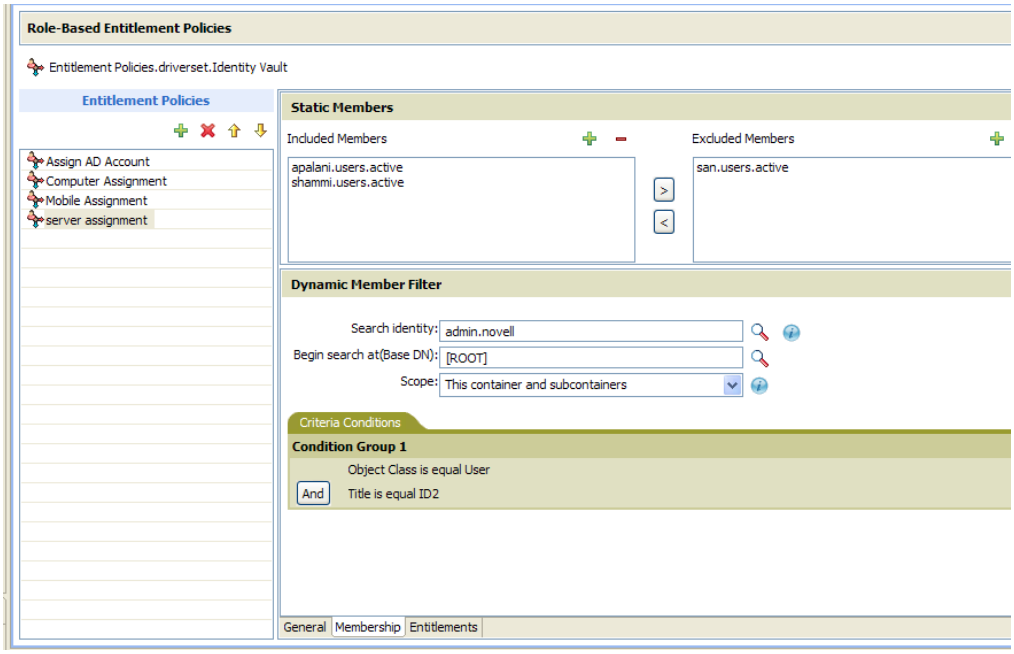
Under the *Static Members*, fill in the fields:

Include Members: Type the DN of a user you want to include, or click to browse for and select the user, then press Enter to add the user to the inclusion list. To remove a user from the inclusion list, select the user and and click icon. To edit a user name, double-click the user.

Exclude Members: Type the DN of a user you want to exclude, or click to browse for and select the user, then press Enter to add the user to the exclusion list. To remove a user from the exclusion list, select the user and click icon. To edit a user name, double-click the user.

4b Dynamic membership lets you define which users should be members of the entitlement policy by specifying criteria and specifying where in the tree to search for users that meet the criteria. If a user meets the criteria you specify, the policy's entitlements are automatically applied to the user. If the user's information changes and no longer meets the criteria, the entitlements are revoked without any manual intervention.

Under *Dynamic Membership*, fill in the fields:



Search Identity: Specify an object that has the rights that you want to be used when querying for Dynamic Membership. Specify the object with the appropriate rights.

For example, there might be parts of the tree that you have rights to that you don't want included in the query for the dynamic list of members.

You could use this field to specify the Driver Set object, making sure that the Driver Set has the appropriate rights. Or, you could create a User object specifically for use with Entitlement Policies, and assign it the rights you want the query to use.

Begin Search at (Base DN): Specify the base container where you want the user search to begin.

Scope of Search: Specify whether you want to search the base container and all of its subcontainers (*This container and its subcontainers*) or only the base container (*This container only*).

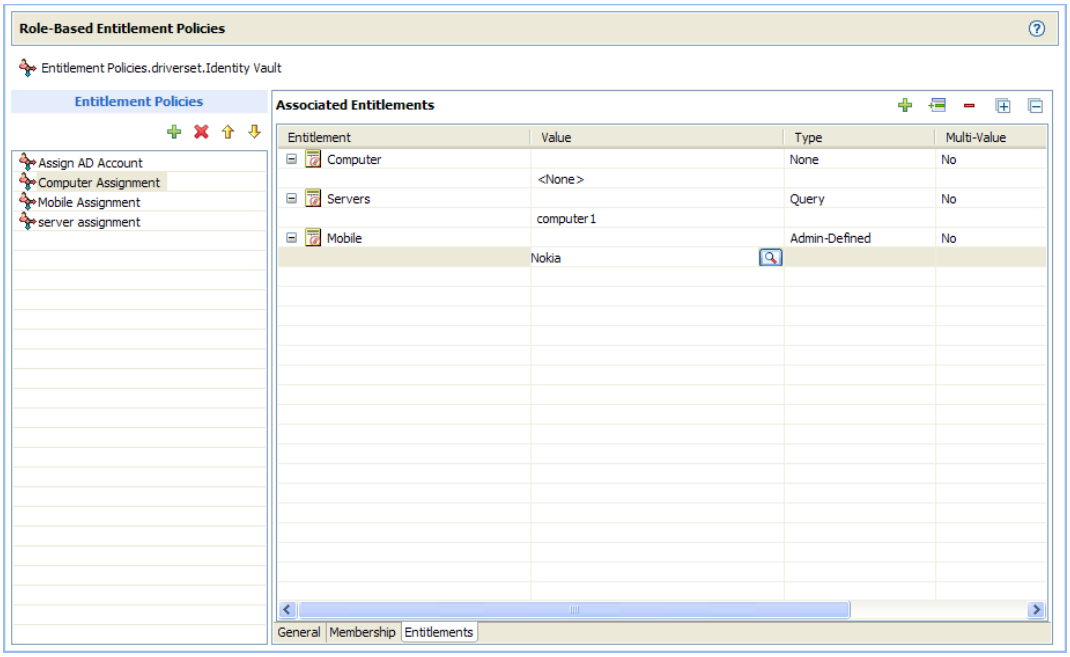
For the entitlement policy to evaluate users in the containers you specify, the users must be in a read/write or master replica on the Metadirectory server that is running the Entitlements Service driver.

Criteria: Specify the criteria that determine which users are members of the policy. The criteria are organized into criteria groups. Each group can contain one or more criterion. You click the icon to add criterion to a group. You can also click *Add New Group* to create additional groups.

By default, the criteria includes all User class objects (and objects of classes derived from the User class) within the search scope.

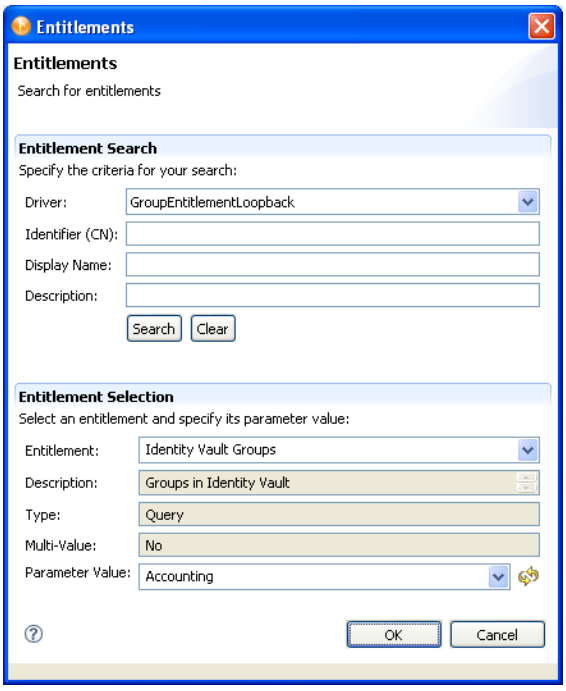
If you create a new object class derived from User, an existing entitlement policy does not recognize that class until you make a modification to the entitlement policy. This prevents users of a new class from being granted entitlements unintentionally. When any modification is made to the entitlement policy, the list of user-derived classes for that policy is updated.

5 In the Entitlements page, add the entitlements you want associated with the policy.



Associating Entitlements

5a In the *Entitlements* section, click icon to add an entitlement for this role. The Entitlement Search dialog box displays:



5b To complete the Entitlement search:

5b1 Choose the driver that contains the entitlement you want.

5b2 Specify the *CN*, *Display Name*, and *Description* on which you want to search.

You can enter a wildcard (such as S*, *S) or a regular expression (such as [A-Z][a-z]*), then click *Search.*

You can enter a value for all of the fields or none of the fields. If you do not supply a value in a particular field, the search returns all of the possible values for that field. If you enter values in one or more of the fields, the values are ANDed together to create the search filter. The search occurs locally. Entitlements contained by the selected driver that match the search criteria are displayed in the Entitlements Selection section.

5b3 The search is complete when the *Entitlement* field displays <*Select an Entitlement*>.

5c To complete the Entitlement selection:

5c1 Choose the entitlement from the Entitlement drop-down list.

The *Description*, *Type*, and *Multi-Value* fields are read-only. These values are obtained from the Entitlement definition.

5c2 Choose the parameter value.

Type	Parameter Value Options
None	No parameter value needed.
Admin-defined	Select from the available parameter values provided in the drop-down list. These values are retrieved from the local Entitlement definition.
Query	Select from the available parameter values provided in the drop-down list. Designer connects to your Identity Vault to retrieve a cached list of available parameter values. These values were obtained by a prior run of the query defined in the Entitlement section. If Designer is unsuccessful in retrieving these values, it displays a dialog box reporting the problem. You can either resolve these issues before attempting to create this entitlement reference again or simply enter your own value in the *Parameter Value* field.There are two buttons to help you retrieve your query parameter values: : *Refresh cached query result*. Click this button if you want to Designer to attempt to connect to the Identity Vault and retrieve the query parameter values. This is most useful if Designer was not able to connect to the Identity Vault on the first attempt. : *Run query in Identity Vault*. Click this button if you want Designer to run the query in the Identity Vault and return the refreshed results.

5c3 Click *OK* to save the definition. Designer displays the definition in the Entitlements table. Query parameter values are translated to the query's full CN when displayed in the table.

5d Configuring Entitlements

5d1 To update/modify the value of the entitlement, click the value cell of the associated entitlement and click icon.

5d2 Click the icon to add a new value.

6 You can assign rights to the objects through iManager. Refer to *iManager 2.7 documentation* (http://www.novell.com/documentation/imanager27/).

Each member of the policy becomes a trustee of the objects you add. There are several reasons why you might want to make the policy a trustee of an object:

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

NOTE: You must verify entitlements that users have on the connected systems, and decide whether to grant, change, or revoke their entitlements. You must re-evaluate the entitlements before creating or editing an Entitlement Policy, when a related membership attribute is changed for a user, or when a user is moved or renamed. iManager supports re-evaluation process. Refer to the *Identity Manager 3.6 Driver for Role-Based Entitlements: Implementation Guide* (http://www.novell.com/documentation/idm36drivers/entitlements/data/bktitle.html) for more information.

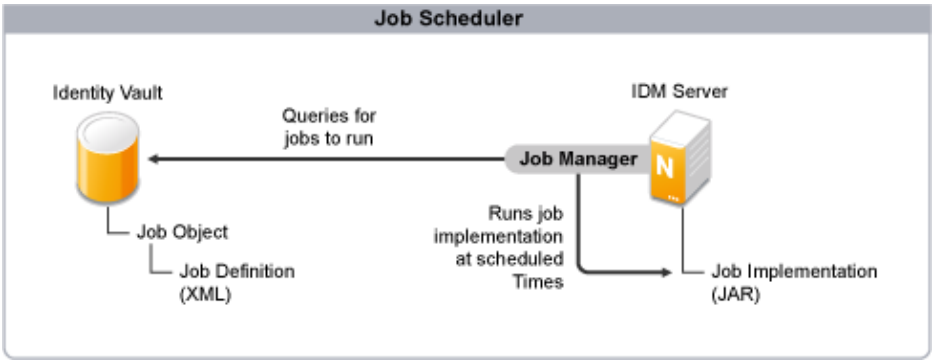
Scheduling Jobs

14

Designer has a job scheduling utility to schedule events, such as setting the system to disable an account on a specific day, or initiating a workflow to request an extension for a person's access to a corporate resource. You can do the following:

- Create a Job object from an installed job definition.
- Define when a job is to run, which servers the job is to run on, the scope of the job in terms of eDirectory objects, and the job reports for intermediate and final results.
- Set values for the job's parameters, its description, and display name.
- Enable or disable a job, manually start a job, stop a job that is running, and display a list of running jobs.

Figure 14-1 *High-level view of Job Scheduler process*



The Job Scheduler consists of the following principal components:

Job Manager: Identity Manager includes a Job Manager that is responsible for launching scheduled jobs. It runs in the background on each Identity Manager server and checks every minute to see if a job needs to run, based on the job definition. When it encounters one, it runs the appropriate Job Implementation.

Job Object: The Job Object you create in Designer contains all the information necessary to invoke the job. It includes the name, description, schedule, server list, and XML job definition.

Job Definition: The Job Definition is an XML description of all the parameters necessary to perform a specific job, including the Job Implementation used to actually perform the job on the target servers. The Job Definition is an XML attribute associated with the Job Object.

Job Implementation: The Job Implementation is typically a JAR file that contains the java classes that perform the job on the target Identity Manager server. Each server where you want a job to run must have a copy of the Job Implementation file. At the designated time, as specified in the Job Definition, Job Manager runs the Job Implementation to perform the job.

- Section 14.1, "Creating a Job," on page 328
- Section 14.2, "Editing a Job," on page 330

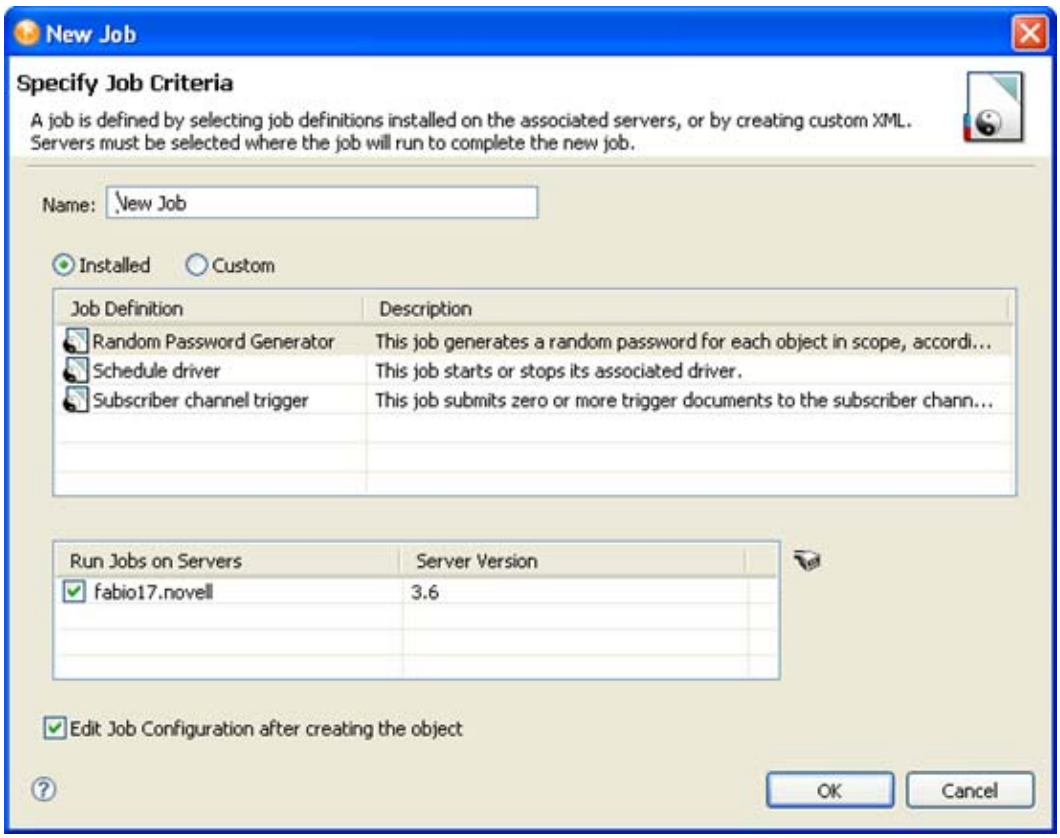
14.1 Creating a Job

1 In the Outline view, right-click a driver and select *New* > *Job*.

NOTE: You can also right-click a driver set and select *New* > *Job* to create a driver health job. For more information about driver health configuration and the driver health job, see Section 4.7.10, “Driver Health Configuration,” on page 105.

This opens the New Job page.

2 In the *Names* field, specify a descriptive name for the job, or use the default name provided.



3 Select *Installed* to create a job using an existing job definition, or select *Custom* to create a custom job definition for this job.

3a If you are creating a job from an existing job definition, select the job definition you want to use from the list of available jobs.

Choose a definition for the job. The New Job Wizard comes with three job definitions.

- **Random Password Generator:** Generates a random password for each object in the job’s scope. The password is generated by NMAS to match the Password Policy object that the job references. These Password Policy objects are not usually the same as those used for eDirectory user password policies.

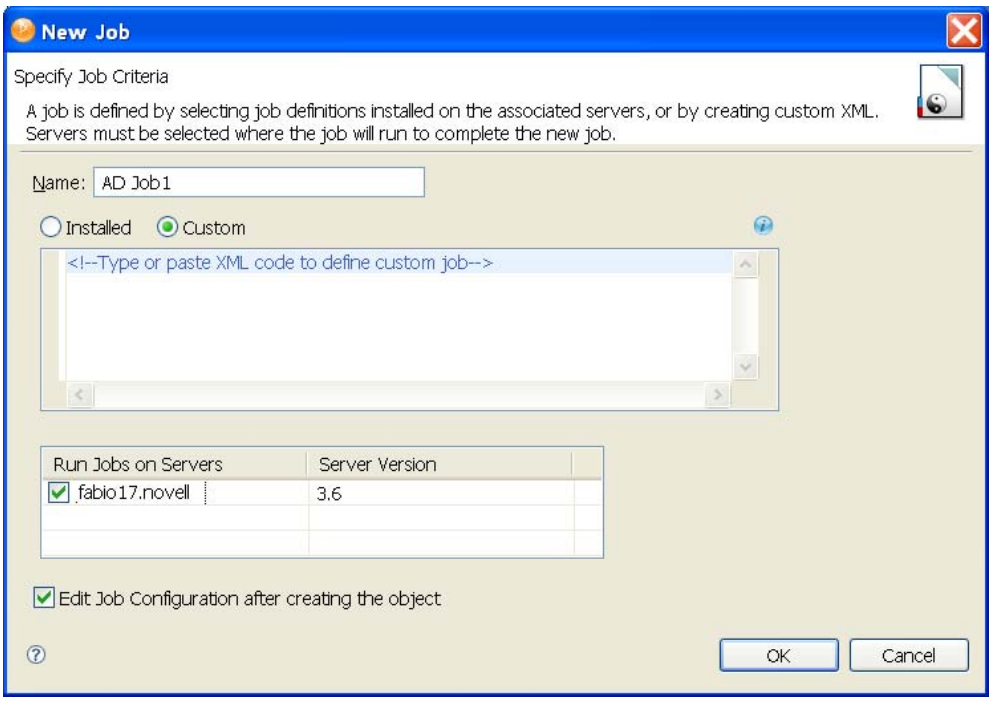
  The job submits the generated passwords one at a time to the driver’s Subscriber channel. The Subscriber channel policies must do something useful with the passwords.

- **Schedule Driver:** Starts or stops the associated driver. You can also toggle a driver to start the driver if it is stopped or to stop the driver if it is running.
- **Subscriber Channel Trigger:** Submits zero or more trigger documents to the Subscriber channel. The submission can either be a document per object if a scope is defined, or it can be a single trigger event if no scope is defined.

  Trigger event documents identify the job and the scope object. A trigger event can bypass the cache and "go to the head of the queue" if desired. You will probably use trigger jobs the most; they allow you to use driver policies that you can customize for your personal requirements.

NOTE: Click the *Update Job Definitions from Server* icon to display any custom job definitions on the selected server. Since Designer is an offline modeling tool, only the Identity Manager job definitions display by default.

3b If you are creating a custom job definition, paste the job definition XML into the code field.



The code field isn't designed for entering XML directly, although you can do so if desired. Identity Manager provides a Job Scheduler DTD that defines the XML structure for job definitions. For more information, see Jobs DTD (http://www.novell.com/documentation/idm36/policy_dtd/data/dtdjobschoverview.html) in the *Novell Identity Manager DTD Reference*.

NOTE: The Job Scheduler automatically validates the custom job XML against the DTD specified in the content, or against the default Job Scheduler DTD if none is specified. It marks any errors it finds so you can review them, and requires you to fix serious errors before allowing you to save the custom job.

---

4 In the *Run Jobs on Servers* field, select the servers where you want to run the job.

5 Select *Edit Job configuration after creating the object* if you want Designer to open the newly created job in the IDM Job Editor window after saving the job object.

6 Click *OK*.

The File Conflict window informs you that you must save the job object to continue. Click *Yes* to save the job and continue.

Continue with "Job Editor Selections on the General Settings Page" on page 331.

14.1.1 Copying a Job

There are two ways to create a new job based on an existing job:

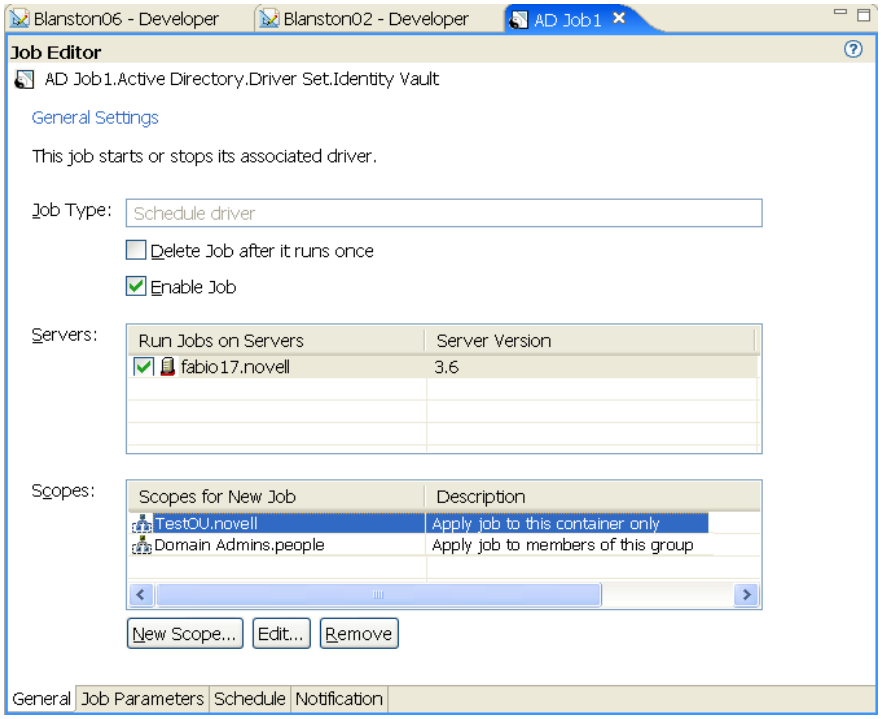
- In the Outline view, right-click an existing job object, then select *Copy*. This creates a duplicate job object in the same location as the original job object.
- Right-click a driver, then select *New* > *Copy From*. This is useful if you want to create a job in a different location from the original job object. For example, in a different driver.

In either case, once you create the new job object, you can then edit the job as needed to fit your needs. For more information, see Section 14.2, "Editing a Job," on page 330.

14.2 Editing a Job

After you create a job, you need to add the necessary information to make the job useful. To edit a job, double-click a newly created job in the Outline view to bring up the job in the IDM Job Editor view.

Figure 14-2 *The IDM Job Editor View*



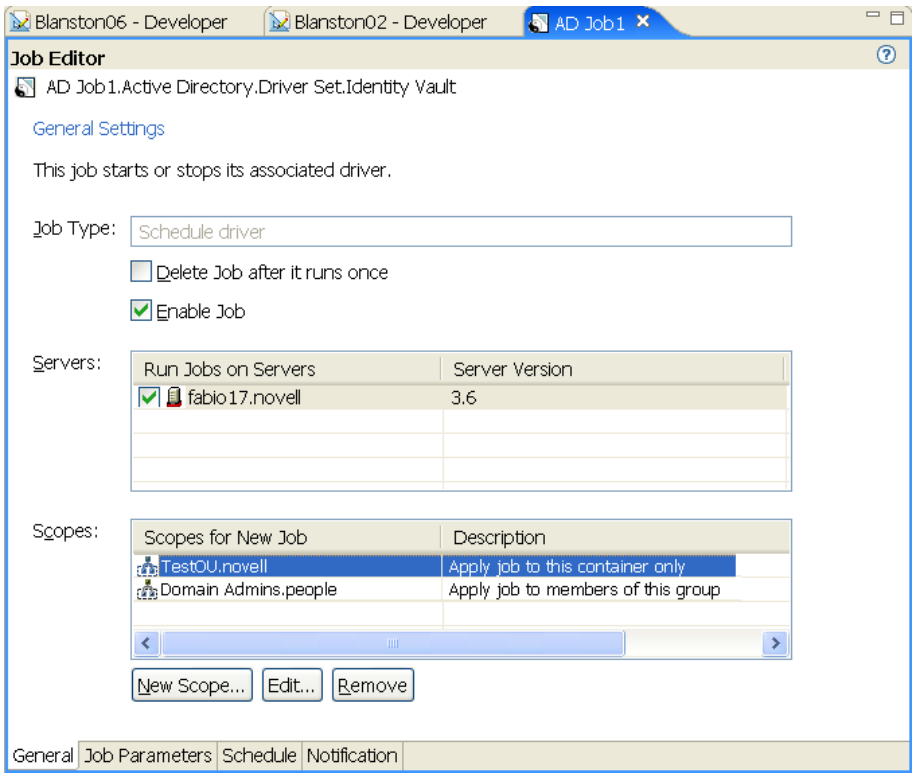
The IDM Job Editor has four tabs at the bottom of its view:

- Section 14.2.1, “Job Editor Selections on the General Settings Page,” on page 331
- Section 14.2.2, “Job Editor Selections on the Job Parameters Page,” on page 334
- Section 14.2.3, “Job Editor Selections on the Scheduler Page,” on page 336
- Section 14.2.4, “Job Editor Selections on the Notification Settings Page,” on page 338

14.2.1 Job Editor Selections on the General Settings Page

The title of the General Settings Page shows the Java class name of the job. This is followed by the job type, which shows the type of job you selected. Under the *Job Type* heading, you can enable or disable the job, or to delete the job after it runs.

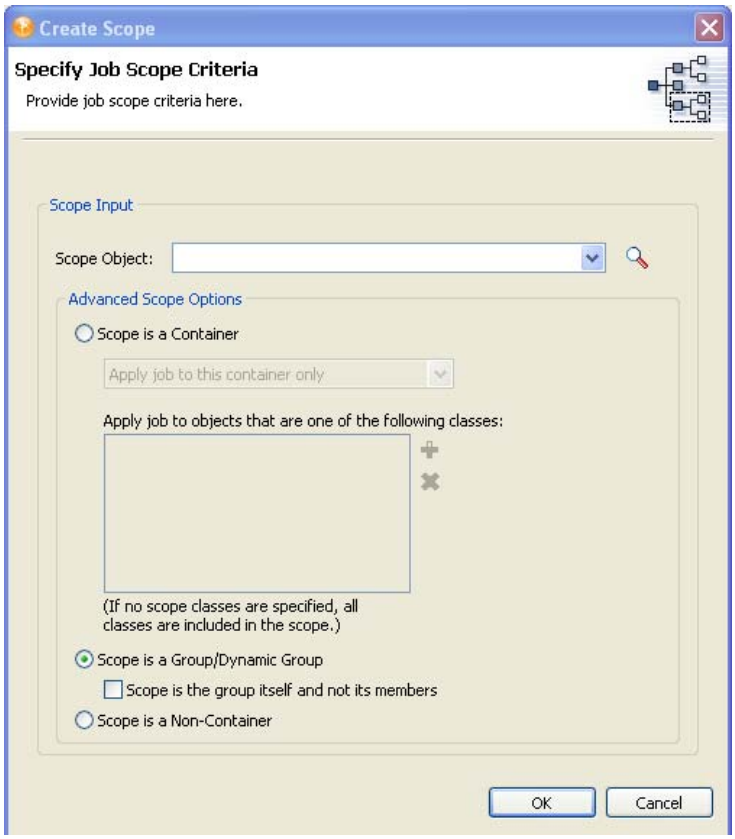
Figure 14-3 *Items Under the General Tab*



1. To delete the job after it runs, select *Delete job after running once*.
2. To disable the job from running, deselect *Enable job*.
3. In the *Servers* column, select the server or servers where this job should run.

   A filtered list of servers is available to help you assign this job. A custom job can be installed on one server but not on another. In this case, the server without this custom job is filtered out of the Server List.

   A job can be assigned to multiple servers as long as it has been installed on each server. Designer only allows this association if the jobs are properly installed and packaged so that the Metadirectory engine can see them.
4. To add a scope to the *Scopes* column, click *New Scope*.



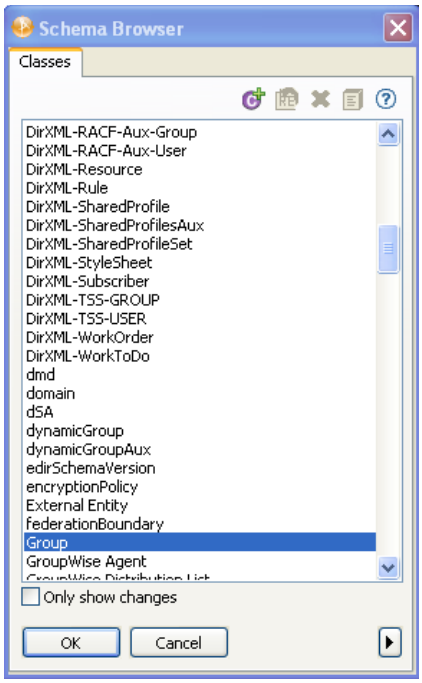
5 To select a scope object, type the Distinguished name of the object or use the *Browse* icon to browse to the object. Click *OK* to add the scope object.

   Scopes allow you to define the objects that this job applies to. An object in eDirectory can be a container, a dynamic group, a group, or a leaf object. If you select a group object, you can apply the job to the group's members, or only to the group. If you select a container object, you can apply the job to all descendants in that container, to all of the children in the container, or to the container only.

6 If the object is a container, select *Scope is a Container*. Then select how you want to apply the job:

   - Apply job to this container only
   - Apply job to children of this container
   - Apply job to all descendants of this container

7 (Optional) If you select *Apply job to children of this container* or *Apply job to all descendants of this container*, you can specify the classes and attributes you want to scope. Click the plus icon to bring up the Schema Browser window to select the classes you want to scope. Select the class schema, then click *OK*.



The classes are added to the Classes box. To remove a class, select it and click the minus icon.

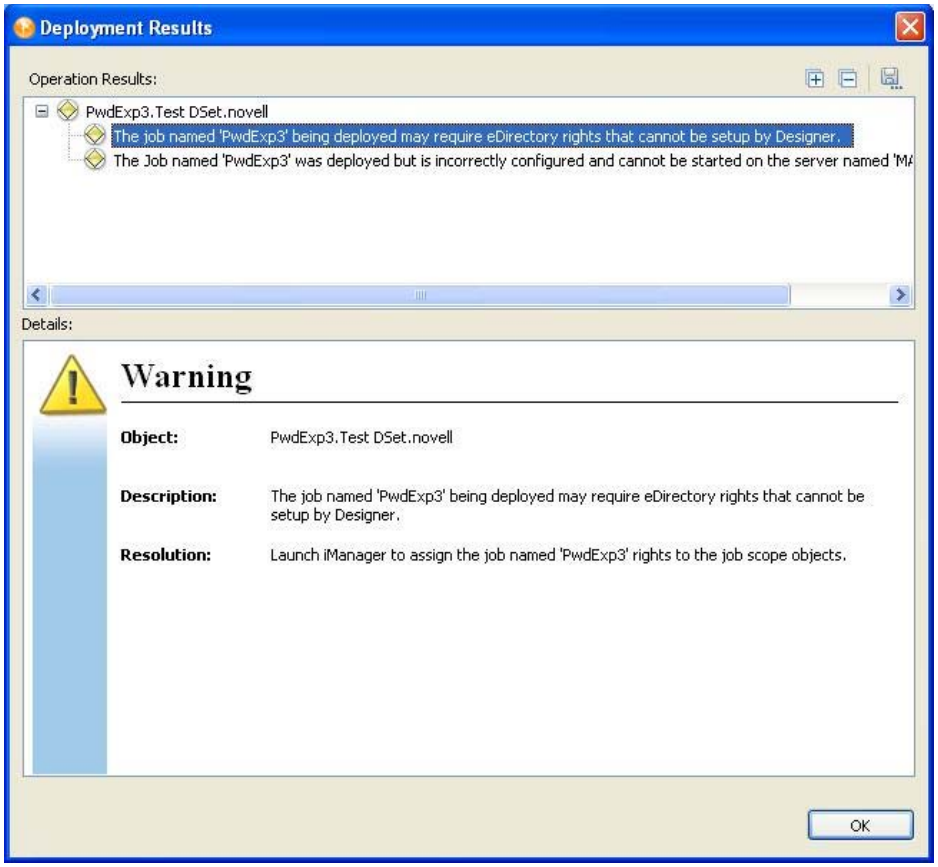
8. If the object is a group or a dynamic group, select *Scope is a Group/Dynamic Group*. You can then select the *Scope is the group itself and not its members* option if the scope is for the group.
9. If the object is a non-container, select *Scope is a Non-Container*.
10. After the scope criteria are selected, click *OK* to return to the General Settings page.
11. If you need to edit a scope, select the scope name, then click *Edit*.
12. To remove a scope, select the scope name, then click *Remove*.

Deploying a Job with Scope Objects

Jobs might need access to eDirectory data and certain Identity Manager actions, such as starting and stopping drivers. Such access is subject to eDirectory rights assignments and is controlled by the rights that are granted to the DirXMLJob object. Although Identity Manager actions are controlled by special attributes, normal eDirectory rights are needed for data reads and writes.

When you deploy a job object that has scope objects, there might be eDirectory rights assignments that Designer cannot properly set up. The rights needed to complete the task depend on the scope objects that are assigned to the job object.

Figure 14-4 *Warning Messages When Deploying a Job with Scope Objects*



If you see this warning when deploying job objects, use the iManager utility to assign eDirectory rights to the job object so it can properly access the job scope objects and complete its task.

14.2.2 Job Editor Selections on the Job Parameters Page

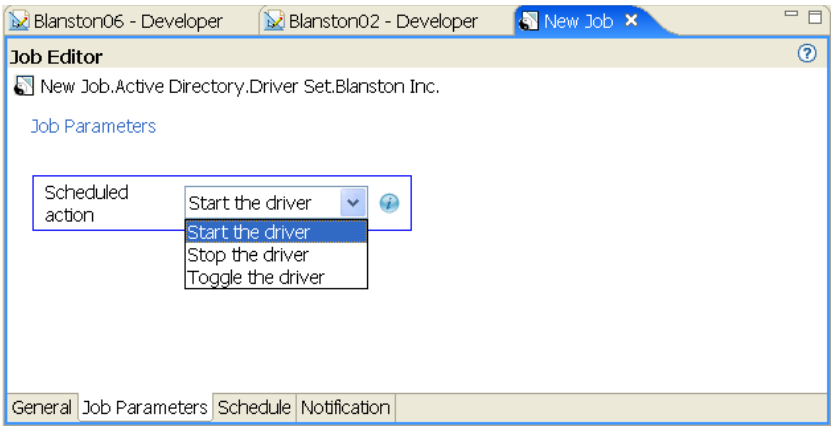
The Job Parameters page allows you to add additional parameters to the job and to view the parameters as they are presently set up. What you can do depends on the type of job you selected.

- “Parameters for the Schedule Driver Job” on page 335
- “Parameters for the Generate Random Passwords Job” on page 335
- “Parameters for the Subscriber Channel Trigger Job” on page 336

NOTE: The parameters for a custom job vary based on the job’s design. For more information about creating a custom job, see Section 14.1, “Creating a Job,” on page 328.

Parameters for the Schedule Driver Job

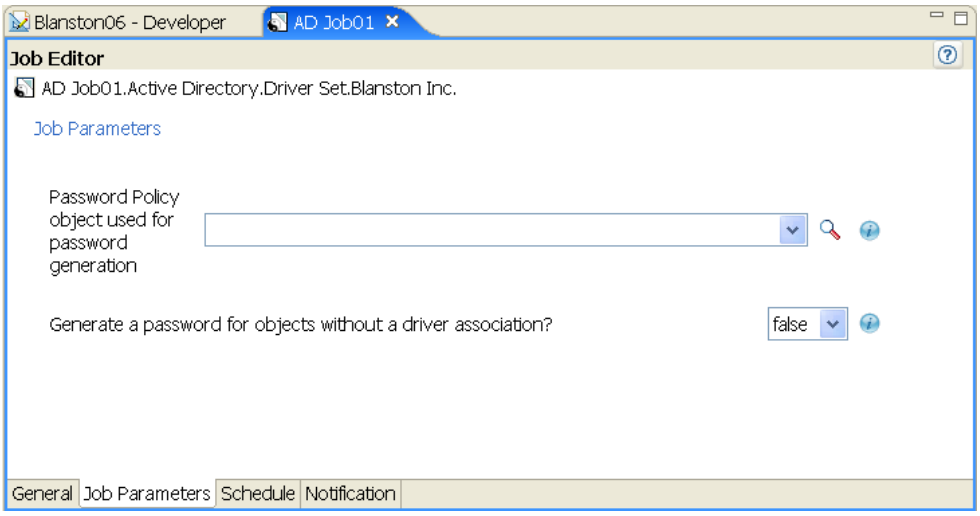
Figure 14-5 *The Job Parameter Page for a Schedule Driver Job*



1. If you want the job to start the driver, select *Start the driver*.
2. If you want the job to stop the driver, select *Stop the driver*.
3. If you want the job to switch from one to the other, select *Toggle the driver*.

Parameters for the Generate Random Passwords Job

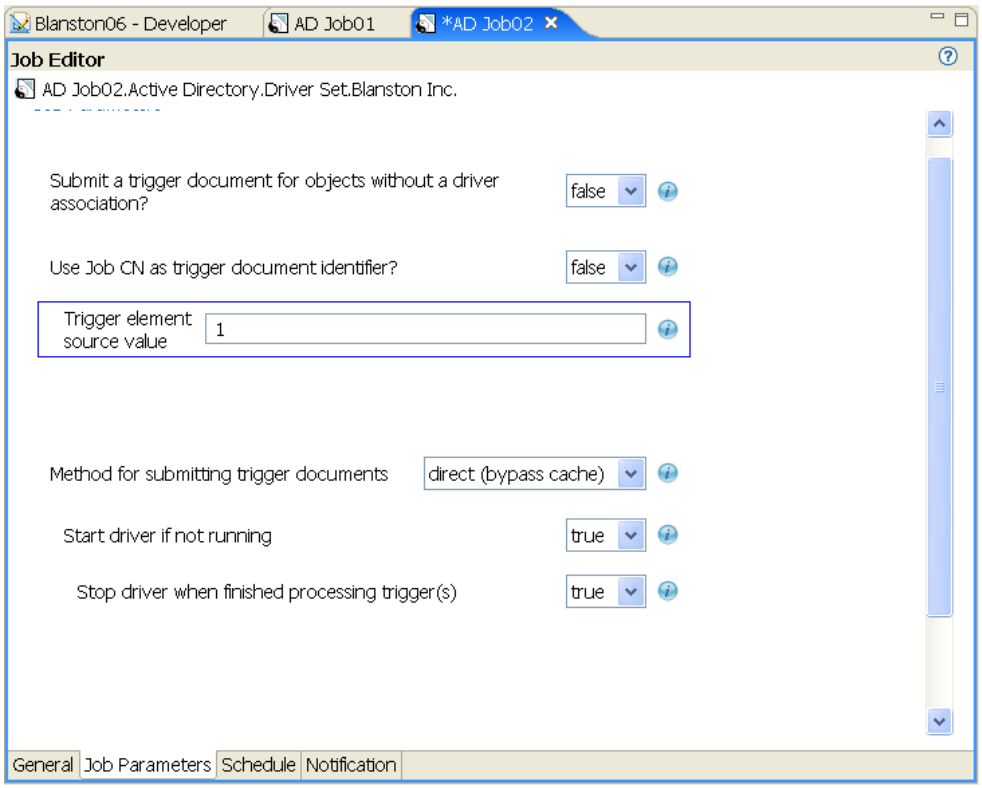
Figure 14-6 *The Job Parameter Page for the Generate Random Password Job*



1. Type the Password policy object's Distinguished name, or use the *Browse* icon to select the Password policy you want to use for password generation.
2. If you want to generate passwords for scoped objects without a driver association, select *True*. Otherwise, select *False*.

Parameters for the Subscriber Channel Trigger Job

Figure 14-7 *The Job Parameter Page for the Subscriber Channel Trigger Job*



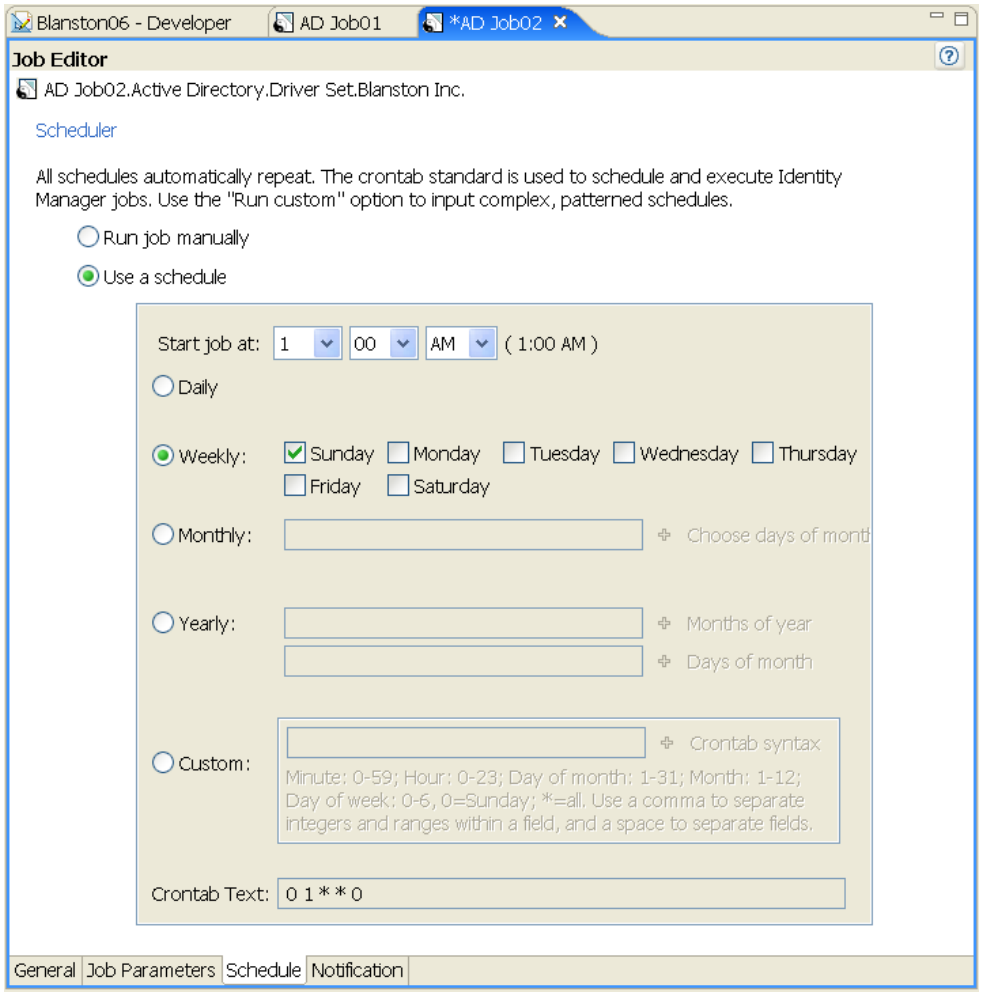
1. If you want to submit a trigger document for scoped objects that do not have a driver association, select *True*. Otherwise, keep the default of *False*.
2. If you want to use the job's Common Name (CN) as a document identifier trigger, keep the default of *True*. Otherwise, select *False*.
3. (Optional) If you select *False*, enter the string that the job can use as the value for the trigger element's Source attribute.
4. Select a method for submitting the trigger documents. If you want to queue the job the trigger is from, keep the default of *Queue (use cache)*. Otherwise, select *Direct (bypass cache)*.
5. (Optional) If you select *Direct (bypass cache)*, you are presented with the *Start driver if not running* option. If you want to start the driver if it is not running, keep the default of *True*. Otherwise, select *False*.
6. (Optional) If you select *True* on the *Start driver if not running* option, you are presented with the *Stop driver when finished processing triggers* option with the default of *True*. Use the default to stop the driver once it finishes processing the trigger job, or select *False* to keep the driver running.

A customized job definition has its own parameter set.

14.2.3 Job Editor Selections on the Scheduler Page

The Scheduler page allows you to set up when you want to run the job.

Figure 14-8 *The Job Options for the Scheduler Page*



1. Select the *Use schedule* option to set the date and time, and whether to run the job daily, weekly, monthly, yearly.

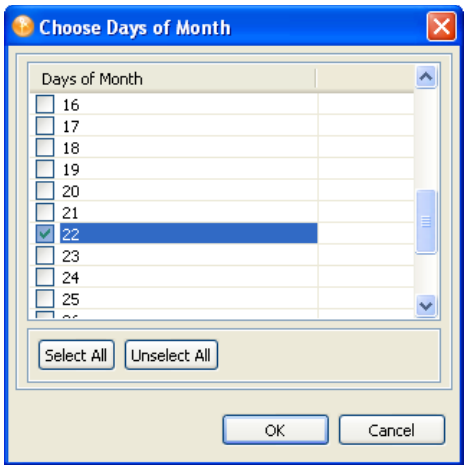
   or

   Select the *Run job manually* option to run the job when you choose to.

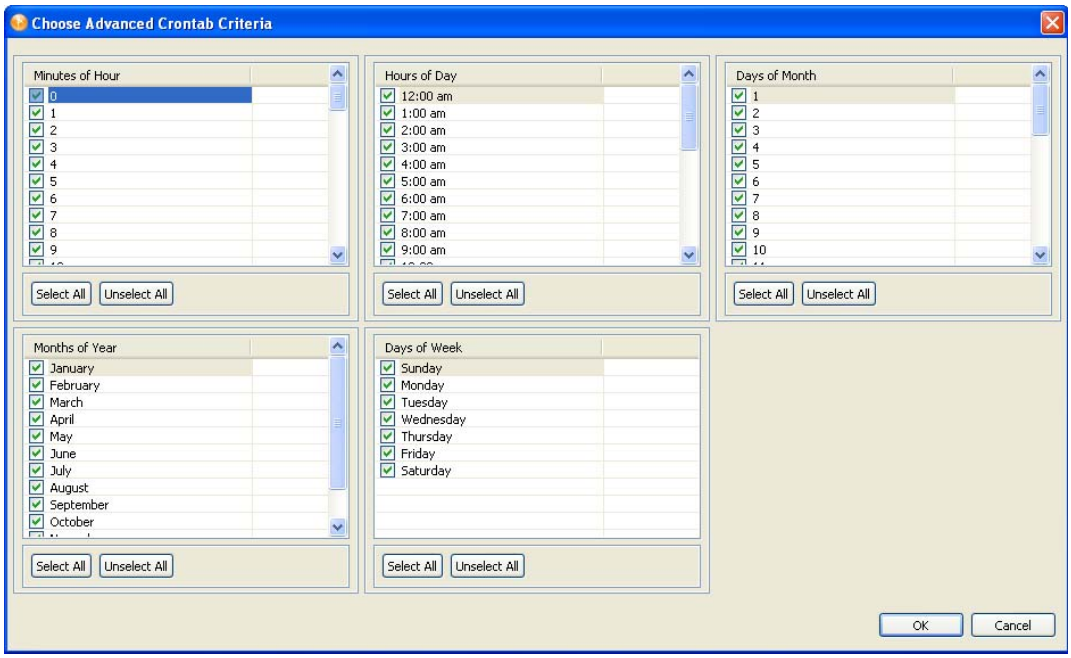
2. With *Use schedule* selected, set the time when you want the job to start running. Use the drop-down menus to select the hours, minutes, and AM or PM. The default is 1:00 AM.

3. If you want to run the job repeatedly, use the *Daily*, *Weekly*, *Monthly*, *Yearly*, or *Custom* fields to select when you want it to run.

   For example, if you want the job to run weekly, select *Weekly*, then the day you want it to run on. If you want the job to run once a month, select *Monthly*, then click the Plus icon to select the day of the month.



4 (Optional) Select *Custom*, to choose minutes, hours, days, months and days of the week from the Choose Advanced Crontab Criteria page.



5 The Choose Advanced Crontab Criteria page default has everything selected. Click *Unselect All*, choose the time and days you want to run the job, then click *OK* to return to the Scheduler page.

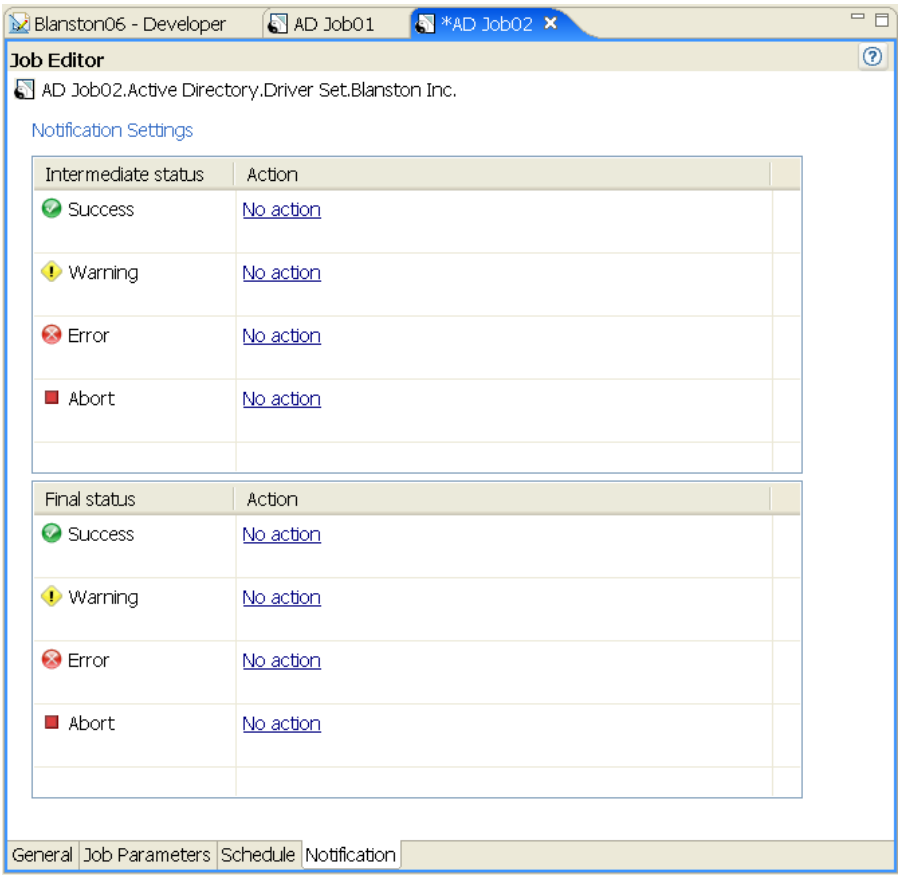
The information displayed in the *Crontab Text* field displays any settings you make on the Scheduler page. For example, if you click *Monthly* and select two days, those two days are displayed in the *Crontab Text* field.

14.2.4 Job Editor Selections on the Notification Settings Page

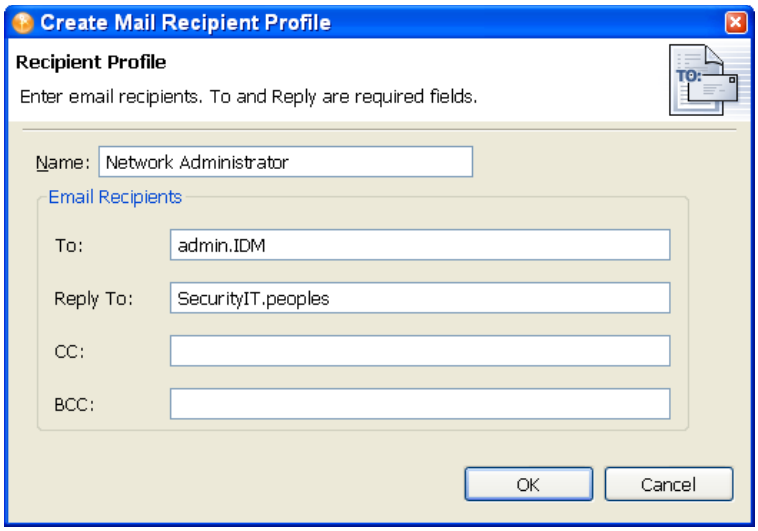
The Notification Settings page allows you to define what you want to do with the job results. It is divided into two parts, Intermediate and Final, with the Success, Warning, Error, and Aborted results for each part.

The Notification Settings page allows you to set how you want to be notified for each result. Actions include sending an audit result or sending an e-mail when the result completes.

Figure 14-9 *The Job Options Under the Notification Tab*



1. If you select *Send email for this event*, Designer allows you to search in the *Default Notification Collection* directory for an appropriate template to use in the *Notification Template* field. Click the *Model Browser* icon to select an appropriate template.
2. Under Notification Recipients, select who you want to send the results to by typing the user's or group's fully distinguished name. You can use the Plus icon to create a mail profile or click the *Model Browser* icon to choose a mail profile.



The To and Reply fields are required for a profile.

3 When you have filled in the information, click *OK*.

4 If you want the results to go to Novell Audit, select *Use Novell Audit for this event*.

5 Use Step 1 on page 339 through Step 4 on page 340 for each of the options:

- Intermediate Success
- Intermediate Warning
- Intermediate Error
- Intermediate Abort
- Final Success
- Final Warning
- Final Error
- Final Abort

If you do not select an option, no action is taken for the result.

Deploying and Exporting

15

The Deploy feature in Designer places a project, a set of drivers, a single driver, channels, and policies into a deployed Identity Manager system in an eDirectory tree. This can be a production tree or a test tree.

Use the Export feature to make backups of all of your projects and the drivers you want to implement. This way, if something happens to the driver in production, you have a backup.

Use the Deploy feature after you have thoroughly tested the policies that make up your drivers. To test policies, use the Policy Simulator (right-click a policy to see the simulation results of the policy that is being tested) or use the Project Checker to ensure that the project is valid. Then use Deploy to test the policy in a test environment; after you test the driver, you can deploy the driver into production.

You can also use the Import feature to import an existing eDirectory driver, a channel, or a policy; after it is imported, you can modify the object or objects, run the Policy Simulator to ensure that the object is working correctly, then deploy the object back into a test tree for further analysis. For more information about policies, see *Understanding Policies for Identity Manager 3.6* (http://www.novell.com/documentation/idm36/policy/data/bookinfo.html).

To help you decide on changes to make before deploying, you can use the Compare feature to see differences between the objects you are deploying and those that already reside in an eDirectory tree. See Section 15.7, “Using the Compare Feature When Deploying,” on page 348.

- Section 15.1, “Preparing to Deploy,” on page 341
- Section 15.2, “Deploying a Project to an Identity Vault,” on page 342
- Section 15.3, “Deploying a Driver Set to an Identity Vault,” on page 342
- Section 15.4, “Deploying a Driver to an Identity Vault,” on page 344
- Section 15.5, “Deploying a Channel to an Identity Vault,” on page 346
- Section 15.6, “Deploying a Policy to an Identity Vault,” on page 347
- Section 15.7, “Using the Compare Feature When Deploying,” on page 348
- Section 15.8, “Troubleshooting Deployed Objects,” on page 354
- Section 15.9, “Exporting a Project,” on page 354
- Section 15.10, “Exporting To a File,” on page 357

15.1 Preparing to Deploy

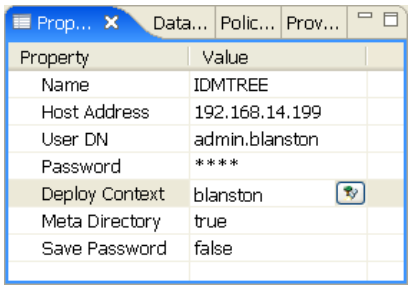
Before deploying a project, run Project Checker and fix any errors that appear.

1. Click *Window* > *Show View* > *Project Checker*, then click the *Run the Project Checker* icon.

After you have corrected any problems to the project, make a backup copy of the project before deploying.

Before you deploy objects into an Identity Vault, you need to designate the Deployment DN (distinguished name), or the place in the tree where you plan on deploying the Identity Manager project or objects. To do this:

1 In Designer, select the Identity Vault that contains the object or objects you want to deploy, then look in the Properties view below the Project/Outline view. (You can also open the Identity Vault's or driver's Properties window.)



2 In the Properties view, fill in the Identity Vault's name, host address, user DN, password, and Deployment DN information if it is not already present.

3 Click the *Browse* button to find the Deploy Context distinguished name on an existing tree if the other information is accurate and Designer can attach to the tree. You need this information to deploy anything, even a policy.

You can also use the driver set's Deploy Context entry if you want to deploy a driver set to a different context than the one designated in the Identity Vault's Properties view. The driver set's Deploy Context entry overrules the Identity Vault's Deploy Context entry.

IMPORTANT: You must have enough rights to access the eDirectory tree that is associated with the Identity Vault to which you want to deploy.

15.2 Deploying a Project to an Identity Vault

To deploy a project to an eDirectory tree that is running Identity Manager, you use the same procedure that you use for deploying a driver set, a driver, channels, or policies. The procedure is described in Section 15.3, "Deploying a Driver Set to an Identity Vault," on page 342.

To deploy an Identity Manager-based project or an object in a project, you must have access to the eDirectory tree that is associated with the Identity Vault you are designing. You also need to know the deployment DN (distinguished name) context, or the place in the tree where you plan to deploy the Identity Manager driver set or driver objects.

15.3 Deploying a Driver Set to an Identity Vault

Suppose you finish a new driver set that you want to deploy into a test tree, or suppose you have imported a driver set, made modifications, and now you want to deploy the driver set back into its working tree. Use the following procedure to deploy an Identity Manager Driver Set object (and all contained Identity Manager drivers) into an existing Identity Manager system in an eDirectory tree:

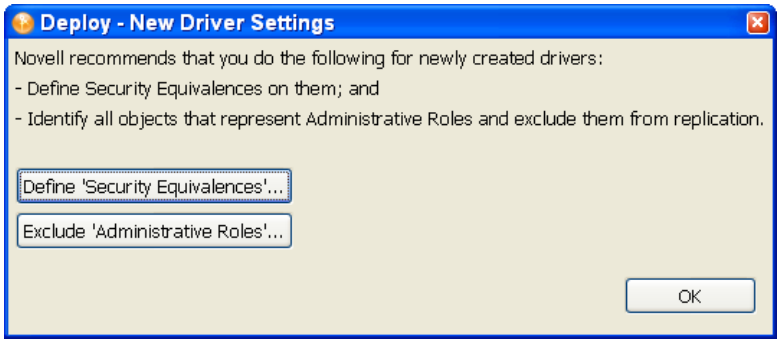
1 Right-click the Driver Set icon in the Modeler view, then click *Live* > *Deploy*.

You can also deploy the Driver Set from the Outline view by right-clicking the Driver Set object, then selecting *Live* > *Deploy*.

The Identity Vault Credentials window displays if Designer can't authenticate to the eDirectory tree specified in the Identity Vault, or if you do not have the Deployment DN designated in the *Properties* tab of the Identity Vault where you are deploying.

2. Use the Compare feature to see differences between the objects you are deploying and those that already reside in an eDirectory tree. See Section 15.7, "Using the Compare Feature When Deploying," on page 348.
3. In the Deployment Summary window, click *Deploy*.
4. Click *OK* to close the Information window.
5. (Conditional.) If you see other informational messages, decide what action to take.

   You might also see a message in the Deployment Results window stating that the deployment was unsuccessful. Click the error messages in the Operation Results portion of the window to see the error descriptions and possible reasons in the Details portion.
6. (Conditional) If this is a new deployment, the Deploy - New Driver Settings window displays. Define security equivalences on the driver set and identify all objects that represent Administrative roles and exclude them from being replicated.



In both instances, Novell recommends that you select the Admin object, and any other objects that qualify in your network environment.

15.3.1 eDir-to-eDir Deployments and SSL/TLS

By default, always deploy both sides of an eDirectory-to-eDirectory connection when you have SSL and TLS enabled. If SSL/TLS are enabled, Designer creates the certificates in the eDirectory tree when you deploy the drivers. SSL and TLS are not enabled or configured by default.

To check your present SSL settings, select *Window* > *Preferences* > *Designer for IDM* > *Configuration* > *eDir-to-eDir SSL/TLS*. After configuration, the Deploy feature adheres to the SSL preference settings under *Certificate Overwrite Policy*.

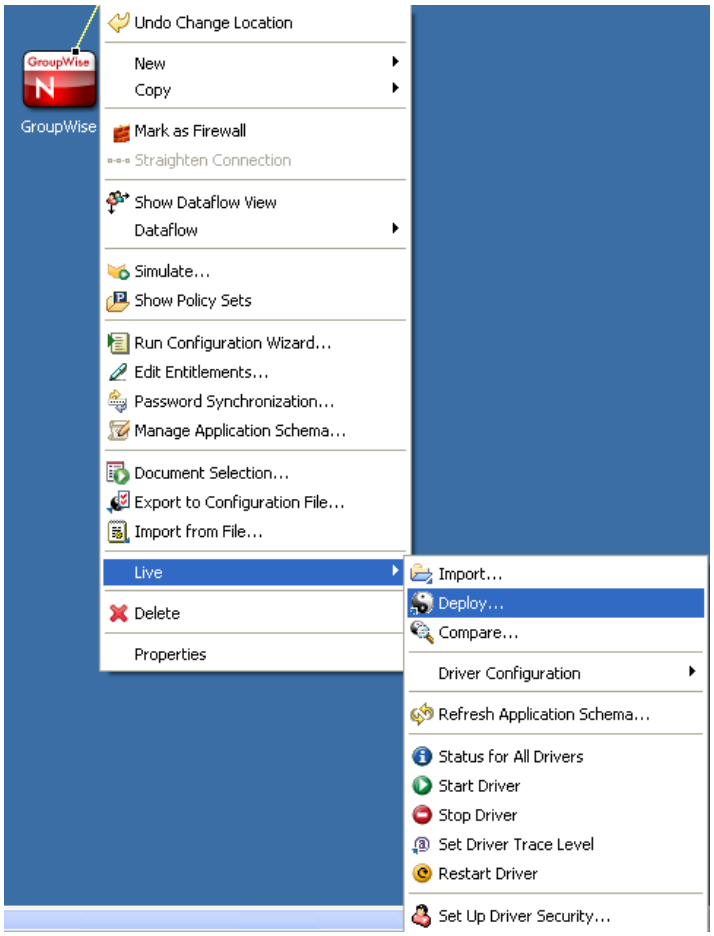
15.4 Deploying a Driver to an Identity Vault

Suppose you finished a new driver object that you want to deploy into a test tree, or suppose you have imported a driver object, made significant modifications, and now you want to deploy that driver object back into its working tree. Use the following procedure to deploy an Identity Manager Driver object (and all contained channels and policies) into a driver set:

1. Select an Identity Vault in the Modeler view.
2. Right-click a driver object connected to a Driver Set icon in the Identity Vault.

   The driver object is represented by a circle icon.
3. Click *Live* > *Deploy*.

   or

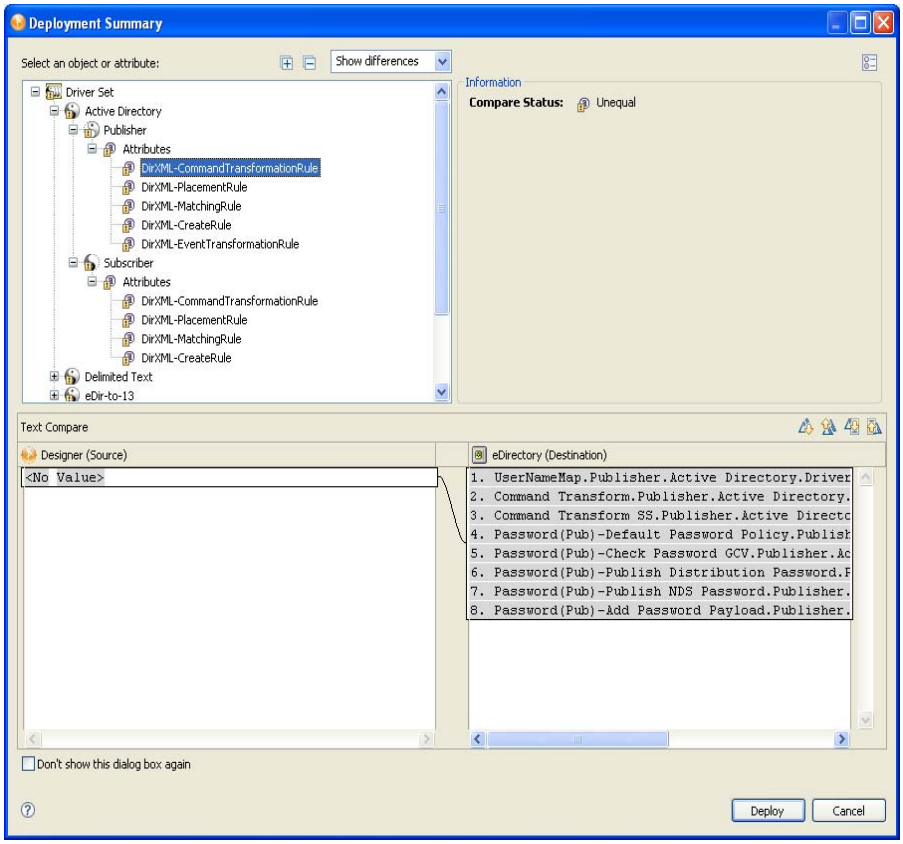


   You can also select the driver object from the Outline view. Click the *Outline* tab, right-click the driver object you want to deploy, then click *Live* > *Deploy*.

   An error displays if Designer can't authenticate to the eDirectory tree specified in the Identity Vault, or if you do not have the Deployment DN designated in the *Properties* tab of the Identity Vault you are deploying to.

The Deployment Summary window shows you the differences between the objects you are deploying and those that already reside in an eDirectory tree. It is the same as the Compare feature. For more information on how to use the Compare window, see Section 15.7, “Using the Compare Feature When Deploying,” on page 348.

NOTE: When you deploy or reconcile a driver, the Identity Manager version of the Identity Vault server is updated to match the live system. Updating the Identity Manager version allows Designer to correctly set the engine controls for the driver so that invalid engine controls are not deployed to the Identity Vault.



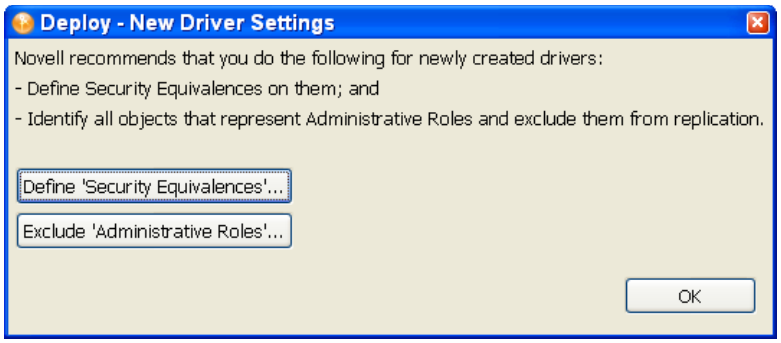
4 Click *Deploy* to begin the process.

5 Click *OK* to close the Deployment Results window.

6 (Conditional.) If you see other informational messages, decide what action to take.

You might also see a message in the Deployment Results window stating that the deployment was unsuccessful. Click the error messages in the Operation Results portion of the window to see the error descriptions and possible reasons in the Details portion.

7 (Conditional) If this is a new deployment, the Deploy - New Driver Settings window displays. Define security equivalences on the driver set and identify all objects that represent Administrative roles and exclude them from being replicated.



In both instances, Novell recommends that you select the Admin object, and any other objects that qualify in your network settings.

You can modify security equivalences and excluded roles after the driver is deployed. To do so, right-click the driver object and select *Live* > *Set Up Driver Security*, or right-click the Application object and select *Driver* > *Set Up Driver Security*.

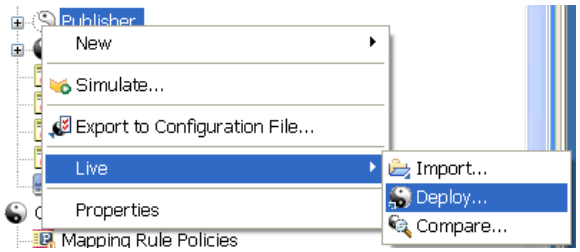
15.5 Deploying a Channel to an Identity Vault

A channel is a grouping of rules and policies, and Designer allows you to deploy a channel object into a driver if necessary. The Subscriber and Publisher channels describe the direction in which the information flows. The Subscriber channel takes the event from Identity Vault (eDirectory) and sends that event to the connected system (application, database, CSV file, etc). The Publisher channel takes the event from the application, database, CSV file, etc., and sends that event to the Identity Vault. The Subscriber and Publisher channels act independently; actions in one are not affected by what happens in the other.

Channel objects must be a part of a newly created driver, or they must be a part of an existing driver that now needs to be modified. Driver objects are created through the Designer or iManager utilities. Because channel objects are a part of a driver object, you deploy a channel object into an existing driver object. If you simply deploy the channel object, Designer creates a skeleton driver as a placeholder for the channel object.

To deploy an Identity Manager channel (a Subscriber channel or a Publisher channel) object and all contained policies into a driver in an Identity Vault:

1. In the *Outline* tab, select the channel object under the driver object. The driver object is represented by a circle icon; the Publisher icon shows a black dot on the icon and the Subscriber icon shows a white dot .
2. Right-click the channel object you want to deploy, then click *Live* > *Deploy*.



An error displays if Designer can't authenticate to the eDirectory tree specified in the Identity Vault, or if you do not have the Deployment DN designated in the *Properties* tab of the Identity Vault you are deploying to.

The Deployment Summary window shows you the differences between the objects you are deploying and those that already reside in an eDirectory tree. It is the same window format as the Compare feature. For more information on how to use the Compare window, see Section 15.7, "Using the Compare Feature When Deploying," on page 348.

An error displays if Designer can't authenticate to the eDirectory tree specified in the Identity Vault, or if you do not have the Deployment DN designated in the *Properties* tab of the Identity Vault to which you are deploying.

3 In the Deployment Summary window, click *Deploy*.

4 After the channel deploys, click *OK* to close the Deployment Results window.

15.6 Deploying a Policy to an Identity Vault

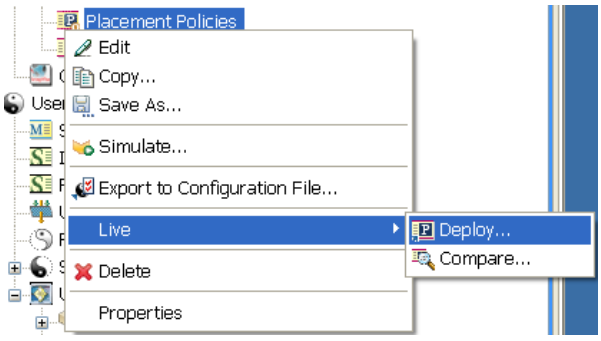
A policy is a collection of rules and arguments that allow you to configure an application so it can send and receive events between itself and an Identity Vault (eDirectory). You use policies to manipulate the data you receive from an Identity Vault or from the application. Each driver performs different tasks and policies tell the driver how to manipulate the data to perform those tasks. For more information about policies, see Understanding Policies for Identity Manager 3.6 (http://www.novell.com/documentation/idm36/policy/data/bookinfo.html).

To deploy an Identity Manager-based Policy object (for example, a rule or a style sheet) into a driver or channel (Subscriber or Publisher):

1 Click the *Outline* tab and select a policy under a driver object or a channel object.

Policies can be of the type DirXML® Script, Schema Mapping, or XSLT style sheet, and each type has its own icon.

2 Right-click a policy object, then select *Live* > *Deploy*.



An error displays if Designer can't authenticate to the eDirectory tree specified in the Identity Vault, or if you do not have the Deployment DN designated in the *Properties* tab of the Identity Vault you are deploying to.

The Deployment Summary window shows you the differences between the objects you are deploying and those that already reside in an eDirectory tree. It is the same window format as the Compare feature. For more information on how to use the Compare window, see Section 15.7, "Using the Compare Feature When Deploying," on page 348.

3 In the Deployment Summary window, click *Deploy*.

4 After the policy deploys, click *OK* to close the Deployment Results window.

15.7 Using the Compare Feature When Deploying

Designer's Compare feature allows you to see differences between driver sets, drivers, channels, and policies that are stored in projects and those that are running in deployed systems. Previous versions of Designer only provided conflict resolution when importing a driver. While importing, you could select which policies of the driver you wanted to update, but you could not view any differences between existing and new values.

Designer now provides conflict resolution on an object-by-object basis and allows you to view the differences between existing and new values when importing and deploying driver sets, drivers, channels and policies. For example, before deploying a driver object in Designer to a driver object that already exists in the Identity Vault, you can run Compare. Compare shows whether the driver objects are equal (no action is necessary) or unequal. If they are unequal, you can choose not to reconcile the driver objects, choose to update the driver object in Designer, or choose to update the driver object in eDirectory.

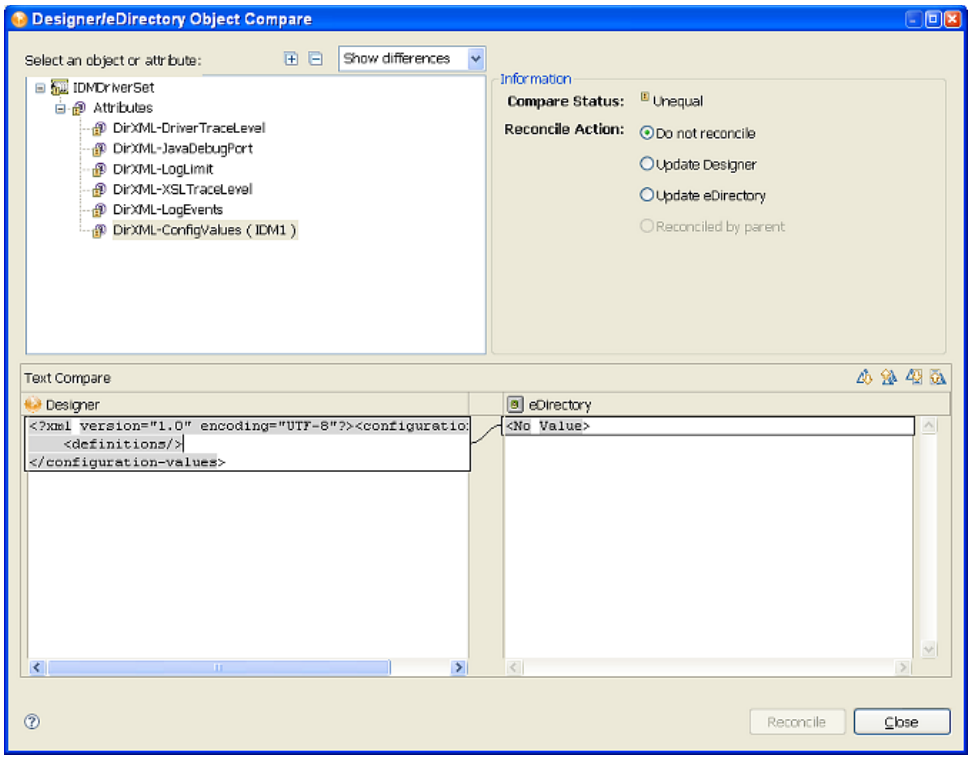
You can run the Compare feature at any time. If you choose to reconcile the differences between drivers objects in Designer and eDirectory while in Compare, you won't need to run Import or Deploy.

- Section 15.7.1, "Using Compare when Deploying a Driver Object," on page 348
- Section 15.7.2, "Using Compare Before Deploying a Channel Object," on page 351
- Section 15.7.3, "Using Compare Before Deploying a Policy," on page 352
- Section 15.7.4, "Matching Attributes with Designer Properties," on page 352
- Section 15.7.5, "Comparing Driver Set and Driver Attributes," on page 352
- Section 15.7.6, "Renaming and Deleting Deployed Objects," on page 353

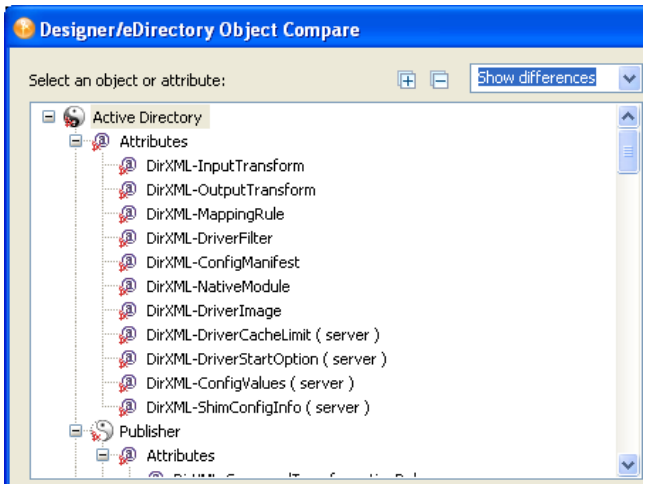
15.7.1 Using Compare when Deploying a Driver Object

Suppose you want to determine if you have deployed all of the changes you have made to a driver object in Designer to the same driver in the Identity Vault.

1 Right-click the driver object in either the Modeler view or in the Outline view. Select *Live* > *Compare* to bring up the Designer/eDirectory Object Compare window.



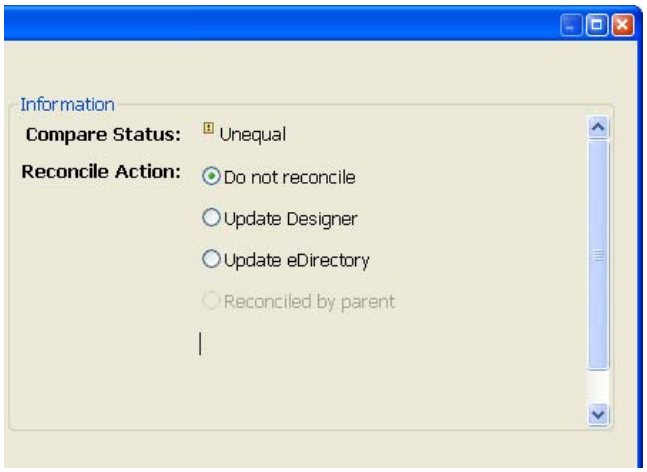
2 In the *Select an object or attribute* portion of the window, you see the listed objects and attributes. Select the attributes and child objects to see the actual differences displayed in the *Text Compare* portion of the window.



The plus icon at the right side of the *Select an object or attribute* allows you to expand all elements in the parent object, and the minus icon collapses all of the elements. The ? icon displays the Summary/Compare dialog box help. Server-specific attributes, which are attributes that have a value for each server that is associated with a driver set, are displayed in the *Attributes* list with the server name in parentheses to the right of the attribute name.

3 By default, the Compare window only displays values that are different between eDirectory and Designer. To view all of the object values, select *Show all* from the pull-down menu. Your choices are *Show differences*, *Show deletes*, and *Show all*.

4 Check to see the status of the values that are shown. Values that are equal are shown as *Equal* on the *Compare Status* line in the Information portion of the Compare window.



The overlay image displayed in the *Compare Status* entry identifies objects or attributes that need reconciliation. The following table describes what you see in the *Compare Status* line and the overlays that you can see:

Compare Status	Description
Equal	The selected attribute's value or all attributes of the selected object are the same in eDirectory and Designer.
Unequal	The value of the selected attribute, or one or more attributes of the selected object, are different in eDirectory and Designer.
Not Deployed	The selected object or the object containing the selected attribute is not deployed to eDirectory.
Not Imported	The selected object or object containing the selected attribute does not exist in Designer.
Renamed	Designer tracks objects that are deployed, then renamed in the Designer project. The Designer and eDirectory DNs are displayed in the value fields.
Unknown	The selected object or object containing the selected attribute cannot be compared, such as a password.
Deleted	Designer also tracks objects that are deployed, then deleted from the Designer project. To delete the object from eDirectory during deployment, select *Delete the Identity Vault object*.

You can also see an *Attribute Note* if you select an attribute.

5 Under the Information portion of the Compare window, select how you want to reconcile the differences between the Source and Destination.

If *Compare Status* shows *Unequal*, you have three choices:

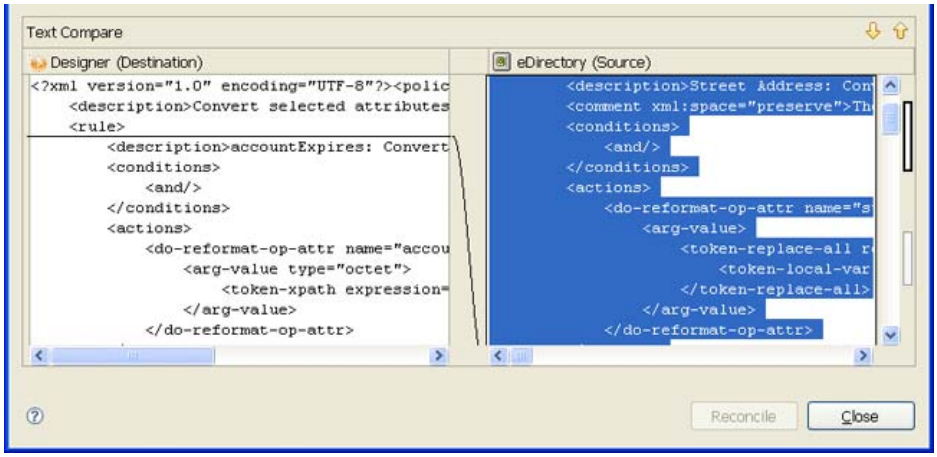
- To do nothing, keep the default value of *Do Not Reconcile*.

- To update the driver in Designer so that it contains the same information as the driver in eDirectory, select *Update Designer*.
- To update the driver in eDirectory to reflect the changes you have just made to the driver in Designer, select *Update eDirectory*.

The green check box in the bottom corner of the icons shows all of the child objects that are being reconciled with the parent object. If you select the parent object to perform the update, then all of the child objects under the parent reflect that choice and you see the *Reconciled By Parent* button selected. If you do not choose a parent object, you can reconcile each child object individually. You can also see a small Designer icon and an eDirectory icon, showing how objects are being reconciled.

6 Check to see the *Text Compare* values.

The *Text Compare* values displayed in the bottom portion of the Designer/eDirectory Object Compare window vary, depending on the object being compared. For instance, Compare shows changes to policies or XML data. The Text Compare dialog box uses the Eclipse* Compare editor to compare attributes that contain XML data, such as policy data, driver filters, or configuration data. The differences in the code are highlighted in blue.



7 After you view the differences, click *Reconcile* to perform the reconciliation actions for each object in the tree, or click *Close* to close the Designer/eDirectory Object Compare window.

15.7.2 Using Compare Before Deploying a Channel Object

Suppose you want to deploy a channel object from the Identity Vault and the same channel already exists in Designer. You can compare the two channels to see similarities and differences.

1 Right-click the channel object in the Outline view.

2 Select *Live* > *Compare* to bring up the Designer/eDirectory Object Compare window.

All Compare windows behave the same as described under the Section 15.7.1, “Using Compare when Deploying a Driver Object,” on page 348.

15.7.3 Using Compare Before Deploying a Policy

Suppose you want to deploy a policy object from the Identity Vault and the same policy already exists in Designer. You can compare the two policies to see similarities and differences

1. Right-click the policy object in the Outline view.
2. Select *Live* > *Compare* to bring up the Designer/eDirectory Object Compare window.

   All Compare windows behave the same as described under the Section 15.7.1, “Using Compare when Deploying a Driver Object,” on page 348.

15.7.4 Matching Attributes with Designer Properties

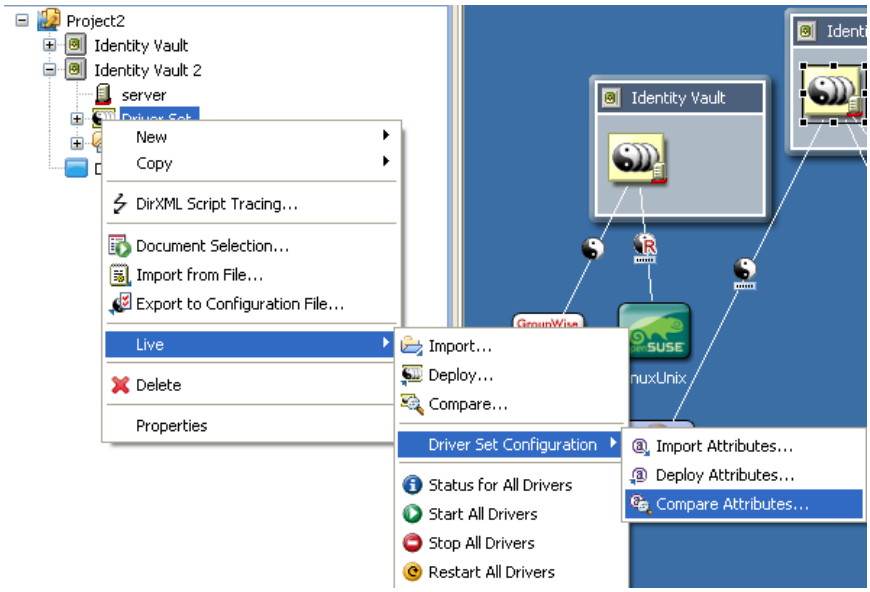
The attributes of the object are displayed in the single select attribute list. Selecting an attribute displays its value below the attribute list with the Designer value on the left and the eDirectory value on the right. The name displayed in the list is the eDirectory attribute name.

Three tables map the eDirectory attribute to the Designer property page or control, where you can change or set the attribute (you can’t make changes inside the Compare window). Table 11-1 on page 264 shows driver set eDirectory attributes, Table 11-2 on page 264 shows driver eDirectory attributes, and Table 11-3 on page 265 shows channel eDirectory attributes.

15.7.5 Comparing Driver Set and Driver Attributes

Use the Compare feature to compare the attributes of a driver set or a driver without comparing all of the child objects.

1. Right-click the driver set or driver, then select *Live* > *Driver Set Configuration* > *Compare Attributes*.

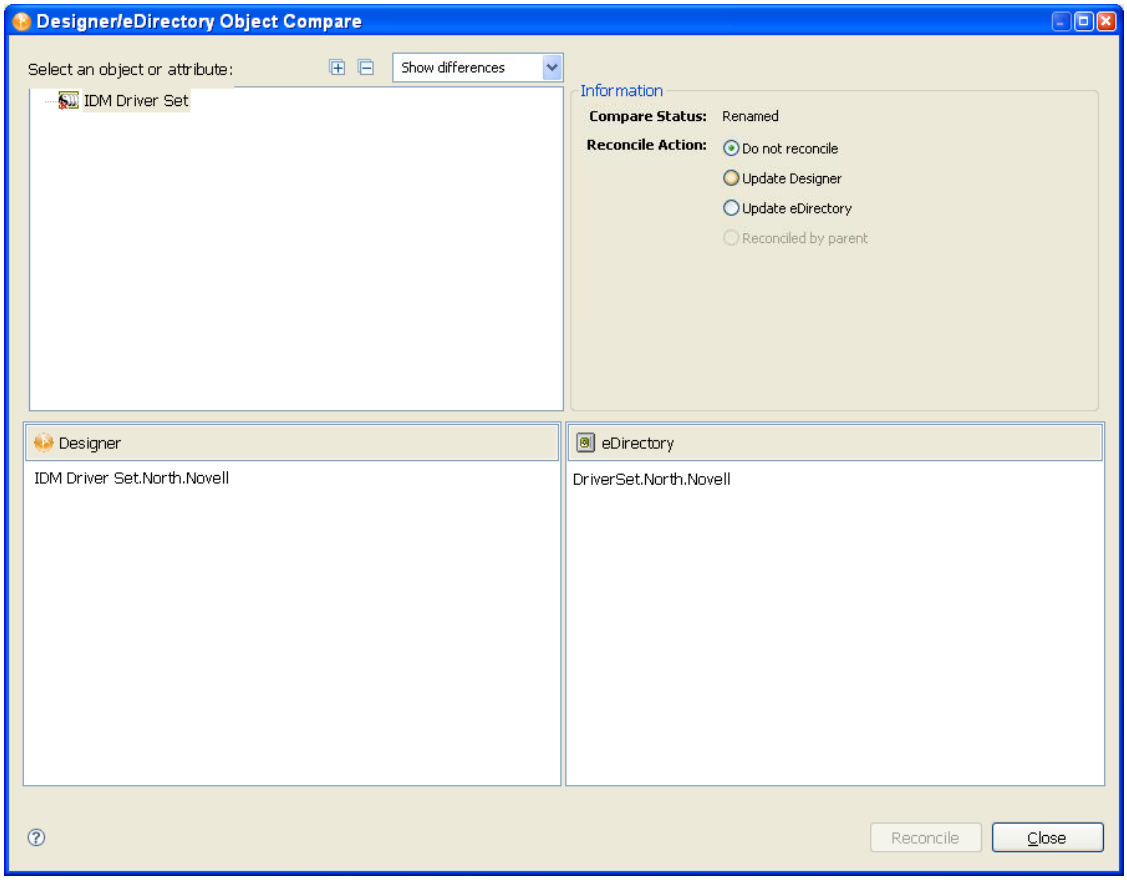


   By default, the Compare windows shows only those attributes that are unequal, but you can select to show deletes, or show all attributes.

15.7.6 Renaming and Deleting Deployed Objects

Designer now tracks objects that are deployed, then renamed in the Designer project. The Designer and eDirectory DNs are displayed in the value fields. The renamed objects are displayed in the Deployment Summary window and the Compare Status entry displays *Renamed*.

Figure 15-1 *Renamed Drivers and Driver Sets*

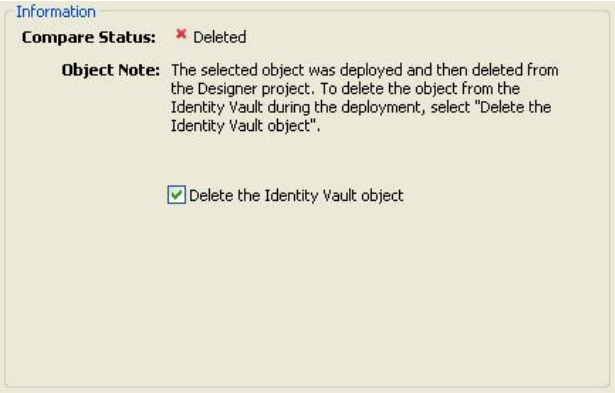


During the deploy operation, the renamed Designer object is renamed in eDirectory. When performing a compare operation, you can reconcile the object by updating either the Designer or eDirectory object name. Only objects that are renamed in Designer are tracked. If an object is renamed in eDirectory, Designer might not locate the associated eDirectory object when building the compare summary.

Designer also tracks objects that are deployed, then deleted from the Designer project. When you deploy the parent of the object that is deleted, you are given the option to delete the object from the Identity Vault. To delete the object from eDirectory during deployment, select *Delete the Identity Vault object*. You can select *Show deletes* from the drop-down menu.

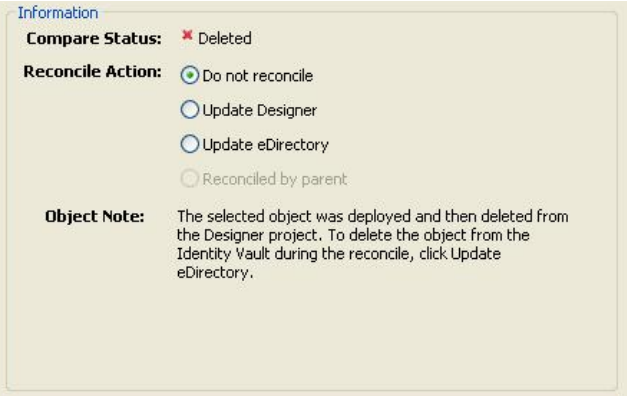
Designer removes the object from the deleted object list if the parent is deployed and the object is not marked for deletion. In the following graphic, a driver was deleted from the driver set.

Figure 15-2 *Deleting an Object in the Identity Vault*



You can use the Compare feature to delete a deleted object from eDirectory or you can re-import the object into Designer.

Figure 15-3 *Reconciling a Deleted Object*



For example, to delete the object from eDirectory, select *Update eDirectory* from the *Reconcile Action* selection. To re-import the object into Designer, select *Update Designer*. Only objects that are deleted in Designer are tracked. If an object is deleted in eDirectory, Designer shows the object as not deployed and creates a new object when you run Deploy or Compare.

15.8 Troubleshooting Deployed Objects

For information on troubleshooting deployed objects, see Section 21.5, “Deploying Identity Manager Objects,” on page 498.

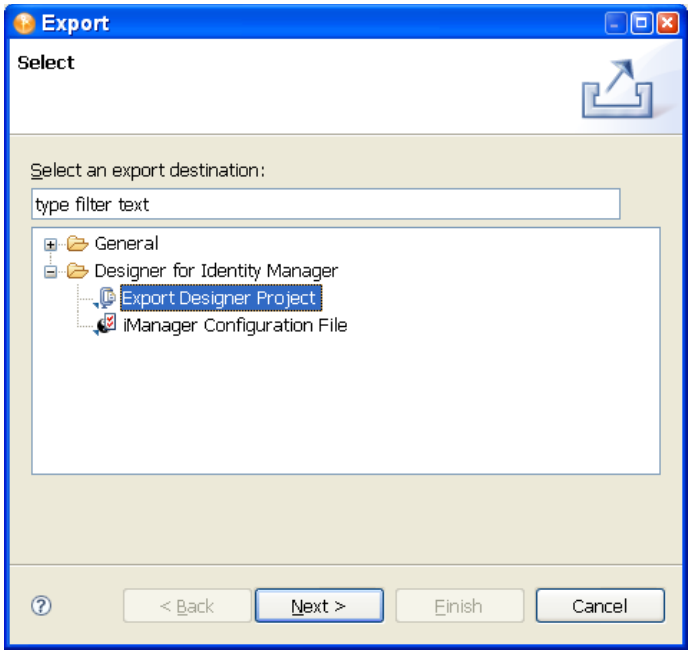
15.9 Exporting a Project

Designer’s export feature allows you to export Projects and Driver Configuration files to a local, removable, or network directory.

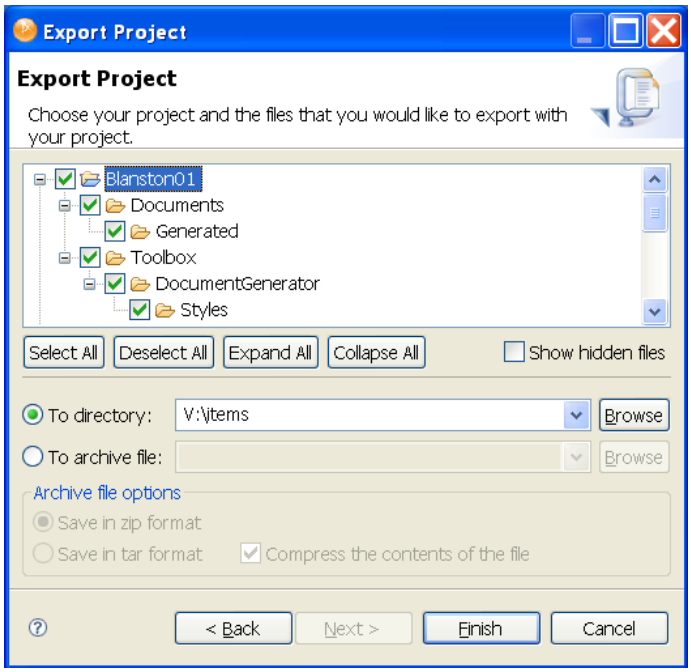
1 Click *File* > *Export*.

You use the Export window to export an existing Identity Manager Project to an archive file or to an iManager configuration file.

2 Select *Designer for Identity Manager* > *Export Designer Project*, then click *Next*.



3 In the Export File System window, select the projects you want to export.



4 Click *Select All* to select all projects in the `designer_workspace` directory (for Windows, the default location is `C:\Documents and Settings\user's_login_name\designer_workspace`).
or

click *Deselect All* to clear the selections. You can then select the projects you want to export. Use the *Expand All* or *Collapse All* buttons to expand or collapse the objects under each project. You can also select *Show hidden files* to display any files that have a period (.) at the beginning of the filename.

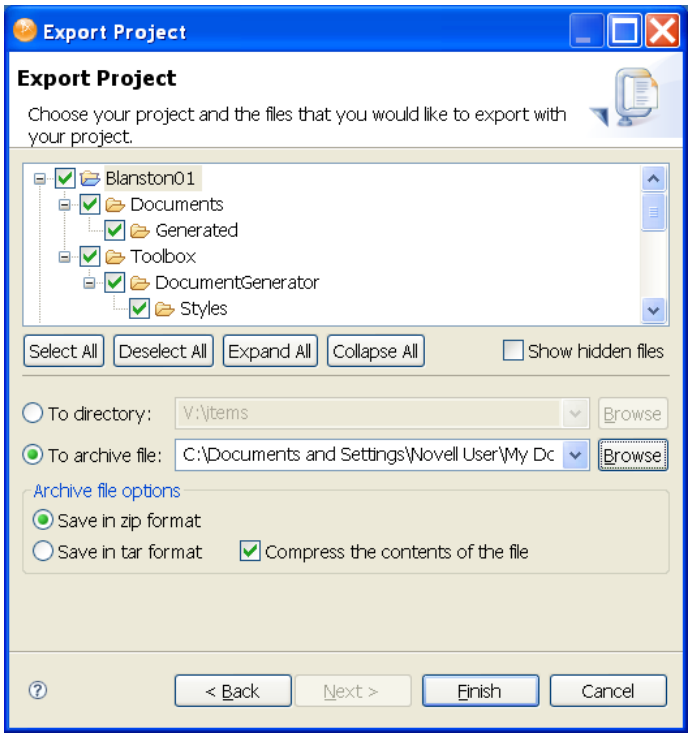
IMPORTANT: You must select all items relating to a project for an import of the project to work.

You can also browse to the directory location where you want to select the resources.

5 After you designate the directory to which to export the projects, click *Finish*.

You can also export projects to an archive file:

1 Select the *To archive file* option in the Export window.



2 Select the projects you want to archive.

3 Designate where you want the archive file saved. You can browse to an already existing file, or type an archive filename.

4 Select the archive format (zip or tar).

5 Select whether you want to compress the contents of the file, then click *Finish*.

With the Project Export Wizard, you don't need to select the model files that are necessary for the project to work, because these files are exported automatically. You can choose to not export any extra files that are included in a project by deselecting them under the project in the Export Project window.

15.10 Exporting To a File

You can use the export feature to export everything you create in Designer, from projects containing all Identity Vaults and their driver sets down to a single policy. If you export a driver configuration file that contains only a policy, Designer creates the parent containment objects, such as a channel, a driver, or a driver set, as part of the exported policy object. These parent containment objects do not contain attributes; they are only the framework of the channel, driver, or driver set.

The exported `.xml` files are compatible with those used by the iManager driver configuration file plug-ins for Identity Manager 2.0.2 and above. This allows you to export configuration files from Designer and import those files through iManager or through Designer's import feature.

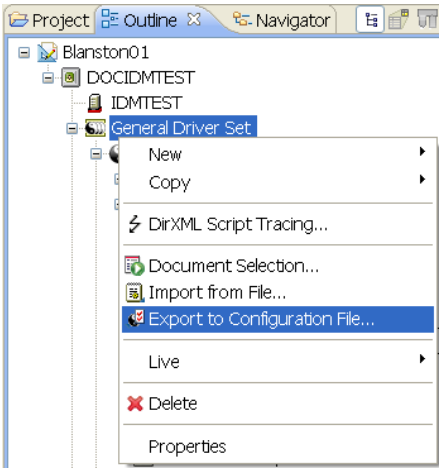
You can export a driver configuration to a file from a number of places, including:

- Section 15.10.1, "The Export Context Menu," on page 357
- Section 15.10.2, "Exporting Configuration Files from the Modeler View," on page 357
- Section 15.10.3, "Exporting Configuration Files from the Outline View," on page 358

15.10.1 The Export Context Menu

To export all driver sets, drivers, channels, and policies found in the Modeler view:

1. Right-click the driver set in the Outline View.
2. Select *Export to Configuration File*.



   Designer uses the name of the driver sets for the `.xml` file.
3. For future reference, name each driver set to denote that it is a driver set and denote the Identity Vault it comes from. You can also add a date to the name. Click *Save*.
4. To close the Export Configuration Results window, click *OK*.

15.10.2 Exporting Configuration Files from the Modeler View

1. Double-click the *System Model* icon under a project name in the Project view to open the project model in the Modeler view.

2 Right-click the Driver Set object inside an Identity Vault icon, then select *Export to Configuration File*.

3 In the Export Driver Configuration window, select a filename and location to use in future references. You can also add a date to the filename if you save a lot of driver iterations.

   By default, Designer uses the name of the driver or driver set corresponding to the object selected. If you right-click an Identity Vault or Driver Set object, you see the Driver Set name in the File Name entry. If you have more than one Driver Set object in the Identity Vault, you see the Export Driver Configuration window with the name of that driver set in the File Name entry for each Driver Set object.

4 Select the directory where you want to store the file, then click *Save*.

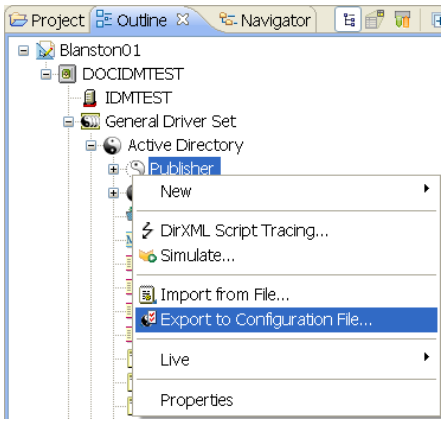
15.10.3 Exporting Configuration Files from the Outline View

You can use the Outline View to save driver sets, drivers, channels, and policy configuration files to local, removable, or network directories. The following procedure documents steps for exporting channels and policies.

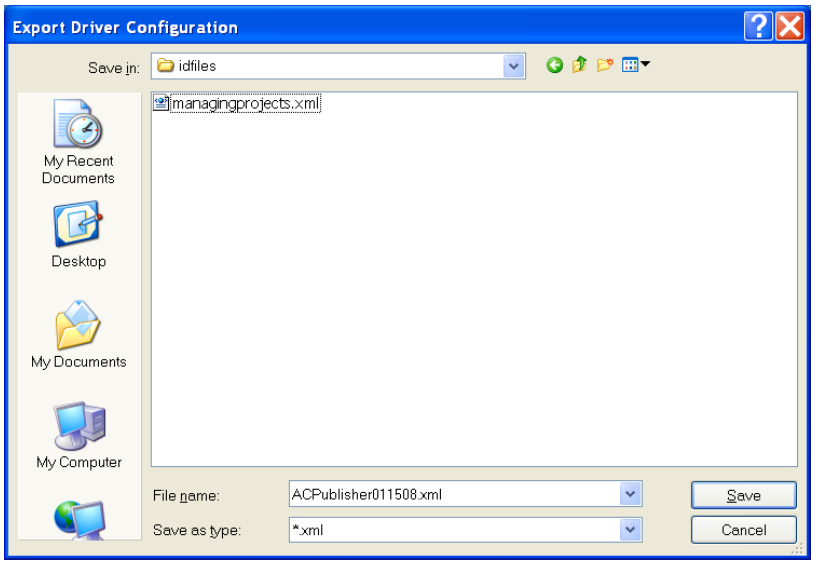
1 Double-click the *System Model* icon under a project name in the Project view. This brings up the project in the Modeler view.

2 Click the *Outline* tab.

3 Right-click a channel object under a driver object, then select *Export to Configuration File*.



4 From the Export Driver Configuration window, select a filename and location to use in future references. You can also add a date if you are backing up multiple iterations of the file.



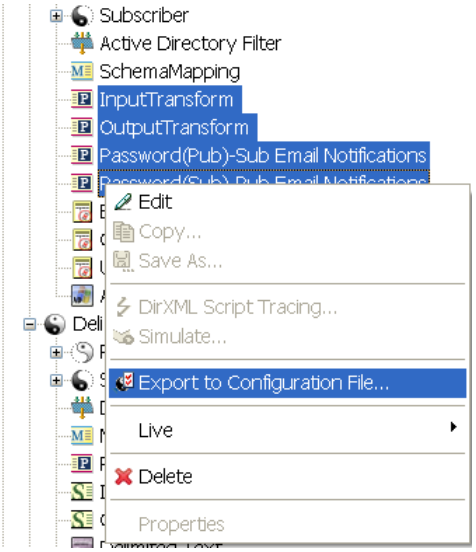
By default, Designer uses the name of the driver or object corresponding to the object selected. You might also need to designate that it is the Publisher channel of an Active Directory driver, along with the date when you saved the file.

5 Click *Save*.

6 In the Export Configuration Results window, click *OK*.

To export one or more policies from a driver or channel:

1 From the Outline view, right-click a Policy object and select *Export to Configuration File*.



You can also use the Ctrl key to select more than one policy, then right-click them as a group and select *Export Policy to Configuration File*.

2 From the Export Driver Configuration window, select a filename and location to use for future reference. You can also add a date if you are backing up multiple iterations of the file. If you are policy files from multiple drivers, include driver and channel information in the filename.

3. Click *Save* for each policy selected; each policy is saved to its own .`xml` file. By default, Designer uses the name of the policy or rule selected.
4. In the Export Configuration Results window, click *OK*.

The Novell XML Editor

16

This section provides an overview of the features of the Novell XML Editor. Topics include:

- Section 16.1, “About the Novell XML Editor,” on page 361
- Section 16.2, “Using the Source Editor,” on page 366
- Section 16.3, “Using the Tree Editor,” on page 369
- Section 16.4, “Attaching a Schema or DTD,” on page 370
- Section 16.5, “Setting XML Editor Preferences,” on page 372

16.1 About the Novell XML Editor

The Novell XML Editor lets you create, edit, and validate XML files.You can edit XML files in either the Source or Tree editor. You can customize the certain behaviors, such as code completion, on the *Preferences* tab.

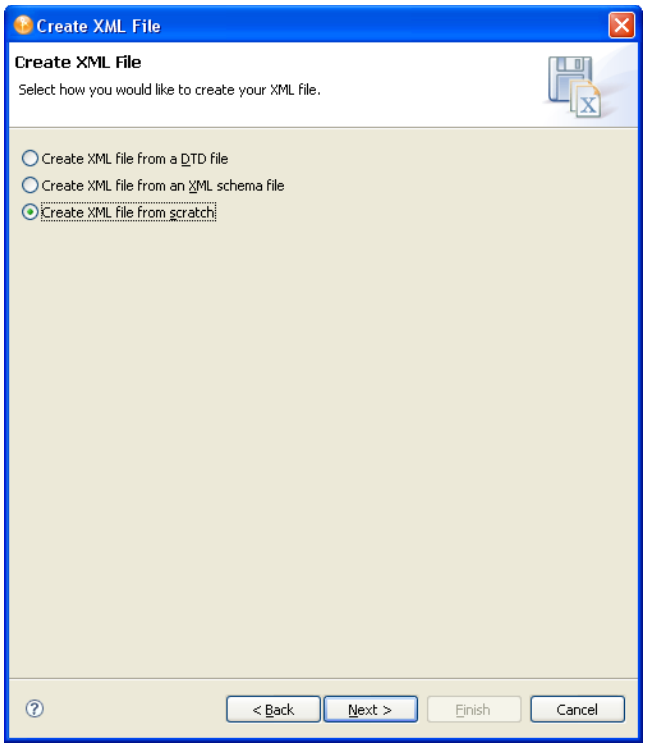
The Novell XML editor is built on the Web Standard Tools (WST) project architecture.

16.1.1 Creating XML Files

You use the New XML File Wizard to create new XML files. The wizard can create an empty XML file or a generated XML file based on an XML schema or DTD. Generated files contain skeleton XML data that is based on a given root element and an XML schema or DTD.

To launch the New XML File Wizard:

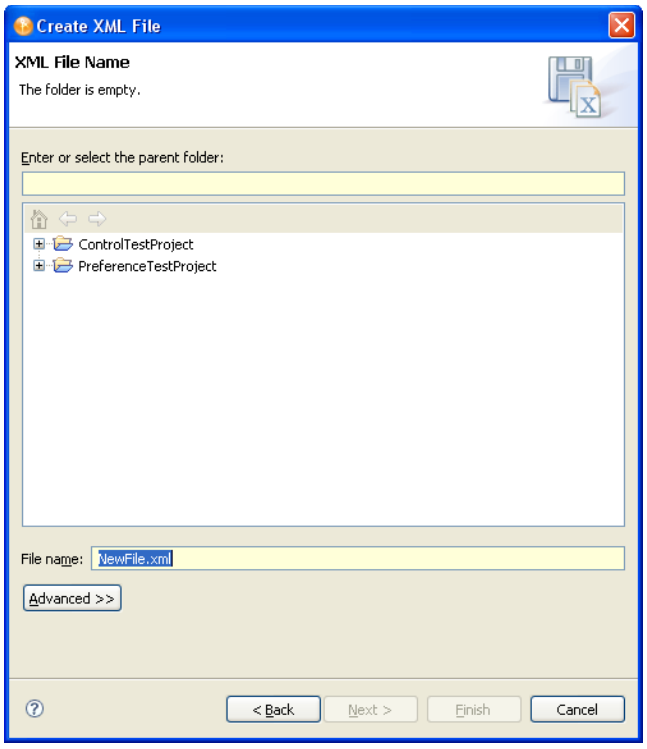
1. Select *File* > *New* > *More Projects.*
2. Open the XML Folder, select *XML*, then select *Next*.



3 Choose one of the source options on which you want to base the new XML file.

Option	Description
Create XML file from a DTD file	Generates an XML document containing a root element and a skeleton based on a DTD that you either import or choose from an existing catalog entry.
Create XML File from an XML schema file	Generates a skeleton XML document containing a root element and skeleton based on a schema that you either import or choose from an existing catalog entry.
Create XML File from scratch	Creates an XML document containing the XML declaration with the version and encoding attributes set to 1.0 and UTF-8 by default.

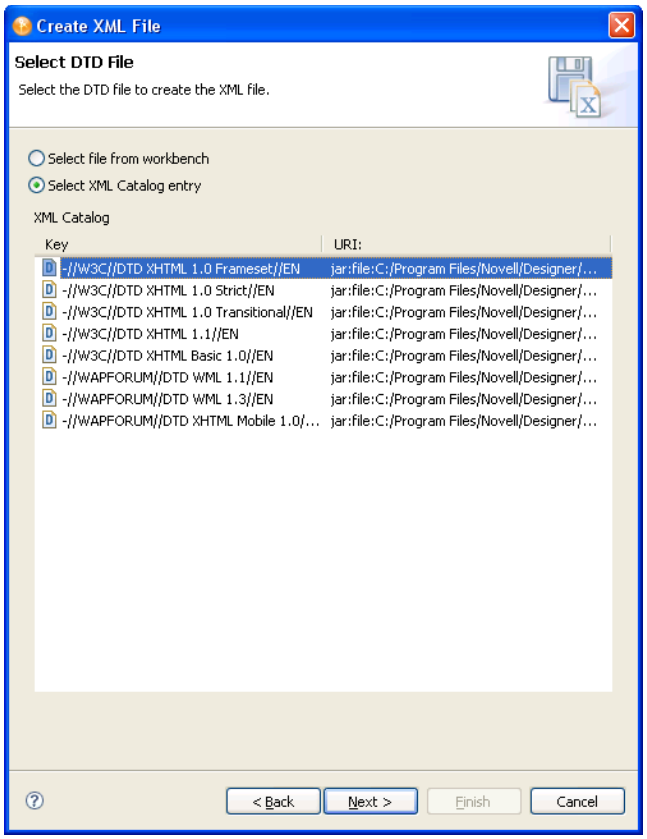
4 Click *Next*. The wizard prompts you to supply the name and location for the new file.



5 Fill in the fields as follows:

Field	Description
Enter or select Parent Folder	Specify where the wizard should create the new file.
File Name	Specify the name of the new file.
Advanced>>	Click this button if you want to specify that the new XML file should link to another file in the file system.

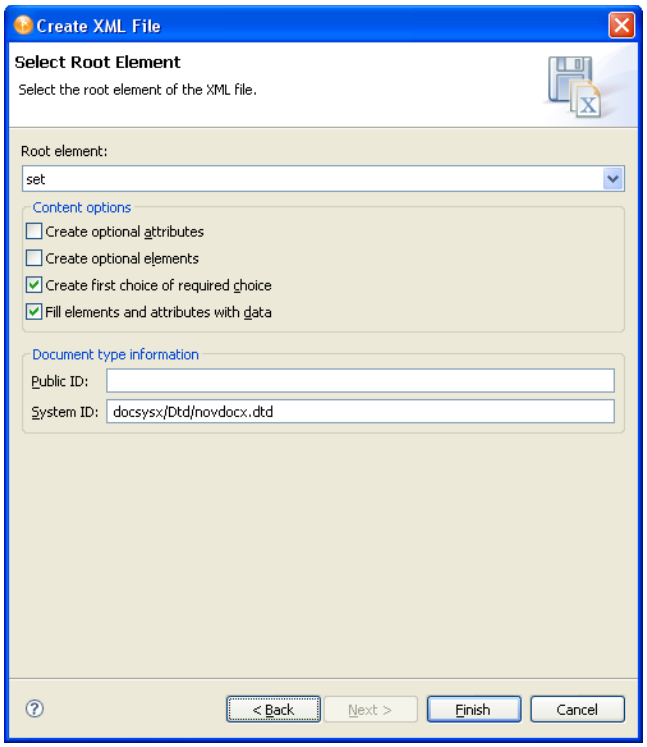
6 If you chose to create the file from scratch, click *Finish*; otherwise click *Next*. You are prompted for the catalog entry (shown below) or the schema or DTD.



7 Fill in the fields as follows:

Field	Description
Select file from workbench	If you choose this option, you must select from a list of DTDs or schemas in your workspace. You can also choose to import a new schema into your project if the schema is not available.
Select XML catalog entry	Choose one of the XML Catalog entries from the list. You can edit this list in *Preferences > Web and XML > XML Catalogs*.

8 Click *Next*. You are prompted to specify the root element.



9 Fill in the fields as follows:

Field	Description
Root element	Choose or type the new document's root element.
Create optional attributes	Select this option if you want the wizard to generate optional attributes.
Create optional elements	Select this option if you want the wizard to generate optional elements.
Create first choice of required choice	Select this option if you want the skeleton XML to always contain the first choice in a required choice. If this is not selected, no elements are inserted for the choice.
Fill elements and attributes with data	Select this option if you want the wizard to generate dummy data in the file for elements and attributes. The generated XML inserts the node name as the data of the elements
Public ID	Specify the file's Public ID
System ID	Specify the file's System ID.

10 Click *Finish*.

You can launch the editor by double-clicking an existing XML file.

16.1.2 Validating Files

You can validate your XML files by using the right-mouse menu in the Source editor. If any validation errors or warnings occur, they are displayed in the Problems view.

16.1.3 Outline View

The editor provides an Outline view containing a tree that displays the structure of the XML document including its nodes are the elements, attributes, text nodes, comments, and so on from the document.

The Outline tree is closely connected to the Source editor and the Tree editor. When you edit a document in either the Source or Tree editor, the Outline tree updates automatically. If the editing results in a document that isn't well-formed, the structure displayed in the tree might seem odd. But the structure corresponds as closely as possible to the well-formed parts of the document.

Editing or generally moving the cursor in the Source Editor or changing the selection in the Tree editor expands and selects the corresponding node (if possible) in the Outline tree. This makes it possible to easily locate the current place in the document.

In a similar fashion, selecting a node in the Outline tree moves the cursor in the Source editor to the textual position of the node (if the Source editor is active) or changes the selection in the Tree editor (if the Tree editor is active). The Outline view provides structural editing capabilities such as inserting and removing nodes.

16.1.4 XPath Navigator

The XPath Navigator view supports context-sensitive editing of XPath expressions and syntax highlighting. It automatically attaches to the currently selected XML editor and uses its Document node as the evaluation context. The namespace context shows all namespaces in scope on its document element.

The view consists of two parts—an editor pane and a results table. When the user types an expression in the editor pane and pauses for 0.5 seconds, the result is shown in the table. If the result is a node list, each row in the table displays an icon for the node type, a short description of the node, and the location of the node in the text (line numbers). Selecting a row in the table selects the text of the corresponding node in the XML editor. However, this is only supported in the Source Editor.

Typing Ctrl-Space, '/', '[' or '(' triggers code-completion—the expression is evaluated up until the cursor location, and insertable elements are shown in a drop-down box.

16.2 Using the Source Editor

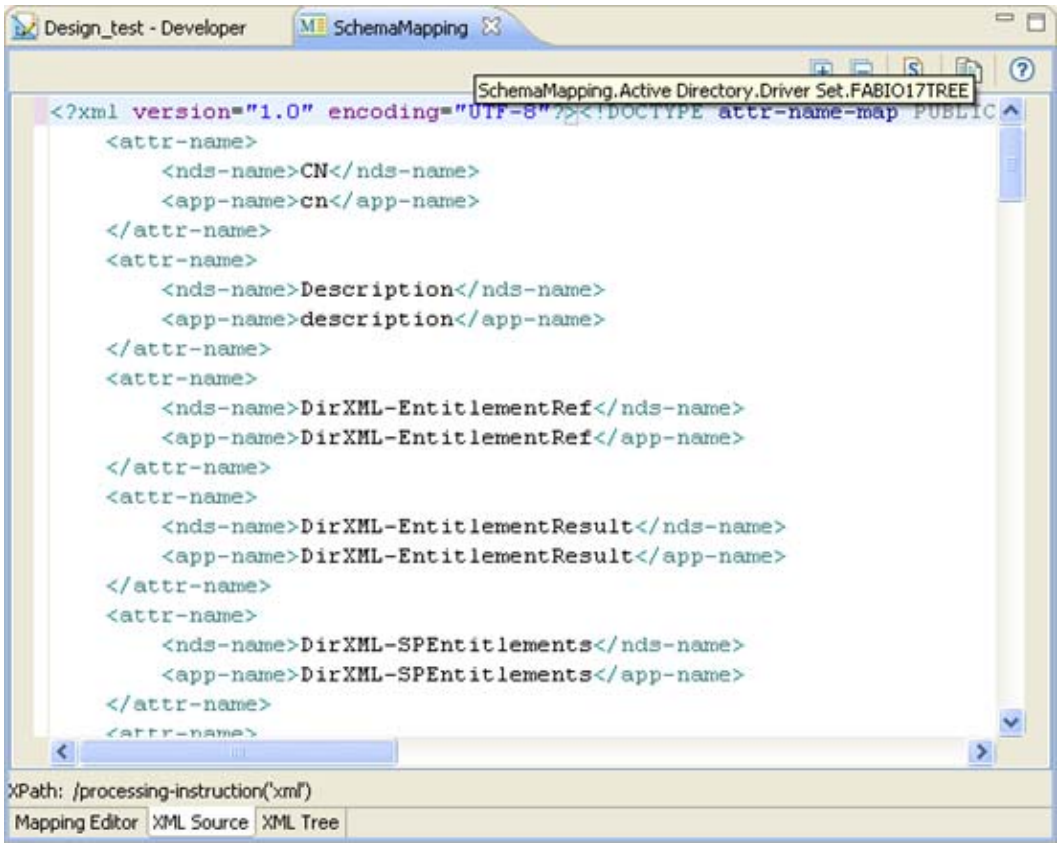
The Source editor supports the following features:

- Syntax highlighting.
- Context-sensitive code completion based on DTD and XML schema.

  The code completion is based on the existing content of the XML document if no DTD or XML schema is associated with the XML document. When code completion is activated and the XML document contains `<root><a><b/></a><a></a></root>`, then type the second <a>, the editor suggests that you add b as a child of the a element.

- As-you-type validation. If the XML is invalid (for example, the > is removed from a tag), the editor indicates the error.
- General text editing operations such as undo, redo, cut, copy, paste, select all.

Figure 16-1 *XML Source Editor*



The XML Editor provides the following toolbar options:

Table 16-1 *XML Editor Toolbar*

Button	Description
	Expands all foldings (if code folding is enabled). You can enable or disable code folding in two ways; • Selecting *Windows* > *Preferences* > *General* > *Editors*. Select *Structured Editors*. Select *Enable Code Folding*. • In the Source editor, right-click in the left ruler to access the *Folding* submenu.
	Collapses all folding (if code folding is enabled)
	Attaches a schema. For more information on using this feature, see Section 16.4, “Attaching a Schema or DTD,” on page 370.

Button	Description
	Shows help

The Source editor right-click menu contains these options:

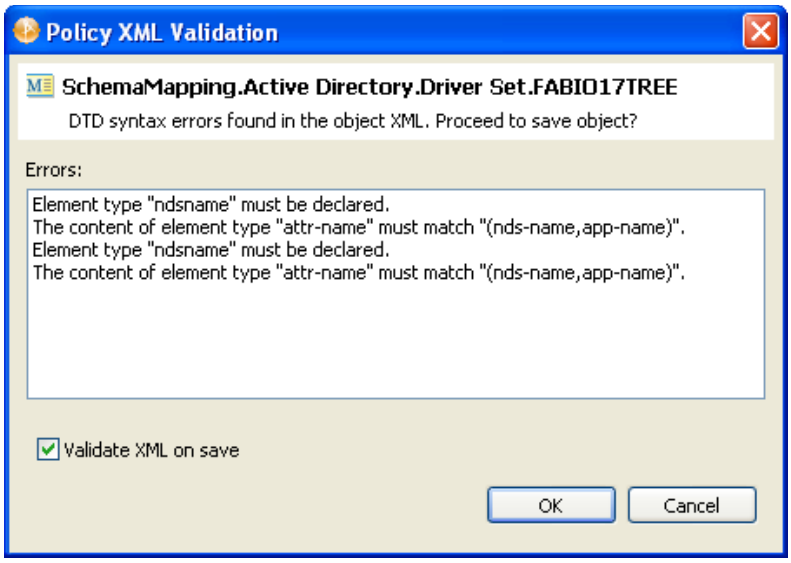
Table 16-2 *XML Editor Source Editor Right-Click Menu Options*

Menu Choice	Description
Revert File	
Cut, Copy, Paste, Undo, Save	Performs the common editor function.
Format	Document: Formats the entire document as specified in the preferences. Active elements: Formats only selected elements.
Clear Validation Errors	Clears reported validation errors from the Problems view.
Validate	Validates the XML document and shows errors in the Problems view.
Preferences	This is the same as setting preferences by using the *Windows* > *Preferences* option. For more information, see Section 16.5, “Setting XML Editor Preferences,” on page 372.

To save XML updates, do one of the following:

- Click *Save* in the Designer toolbar.
- Right-click in the XML Editor, then select *Save*.
- Press Ctrl+S.

When saving, the XML Editor automatically checks the XML to make sure it conforms to the appropriate DTD (Filter DTD, DirXML Script DTD, etc.) It saves non-conforming XML only if you explicitly instruct it to do so. For information about Identity Manager DTDs, see the Identity Manager 3.6 DTD Reference Guide (http://www.novell.com/documentation/idm36/policy_dtd/data/bookinfo.html).



NOTE: You can disable notification of DTD errors in Designer Preferences. To do so, select *Window* > *Preferences*, then select *Novell* > *Identity Manager* > *Configuration* in the left-side navigation. Uncheck *Prompt for errors when validating XML against DTD for all Policy Editors*.

16.3 Using the Tree Editor

The Tree editor supports these features:

Direct Editing: You can directly edit the text fields, including element names, attribute names and values, namespace names and values, text, and comments.

Insertion: You can insert new nodes by using the Tree editor's right-click menu, which allows you to insert nodes as children before or after the selected node. If the node is an element, you can insert attributes. The submenus for *Add Child*, *Add After*, *Add Before* contain the node that can be legally added. If no schema or DTD is associated with the document, the submenus contain *New Attribute* or *New Element*.

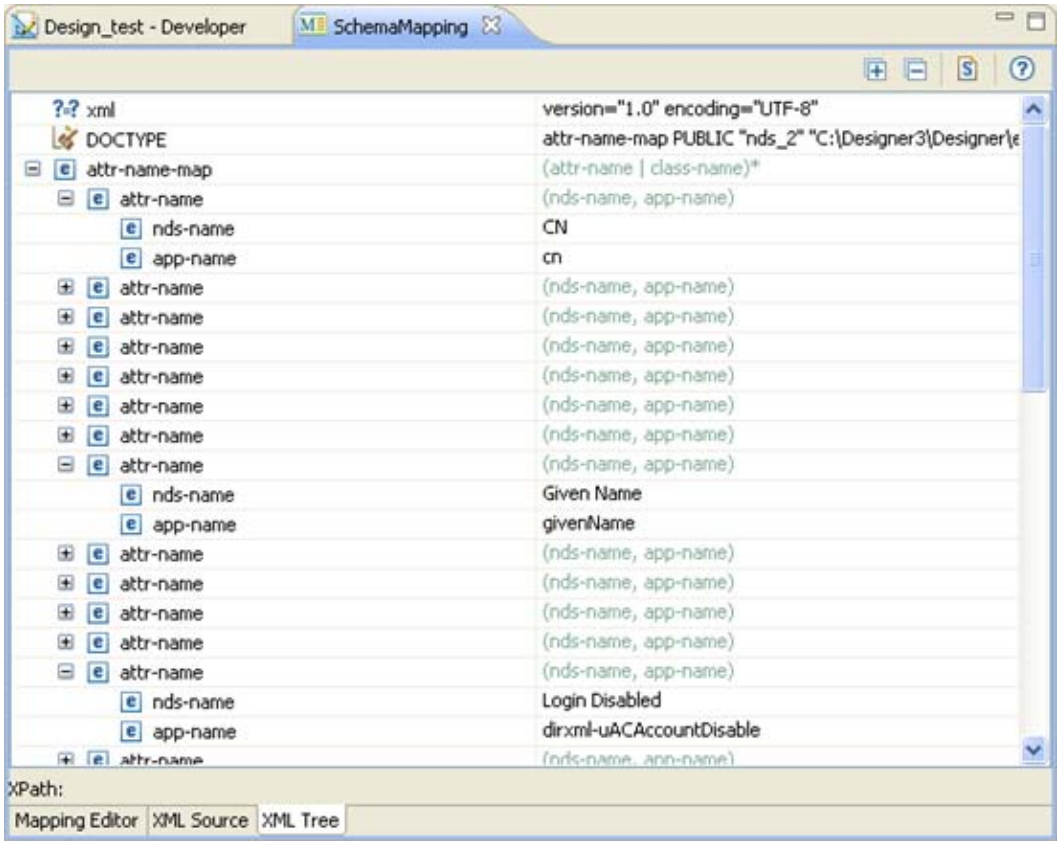
Deletion: To delete a node, select it and either press the Delete key or right-click, then click *Remove*.

Drag-and-drop: Inside the tree and between trees.

General Editing: Operations such as undo, redo, cut, copy, and paste.

The Tree editor displays the XML nodes, with the value of each node displayed in a table cell next to the tree node.

Figure 16-2 *XML Tree Editor*



The Tree editor provides the following toolbar options:

Table 16-3 *Tree Editor Toolbar*

Button	Description
	Expands all nodes.
	Collapses all nodes.
	Attaches a schema. For more information on using this feature, see Section 16.4, “Attaching a Schema or DTD,” on page 370.
	Launches help.

16.4 Attaching a Schema or DTD

Both the Source editor and Tree editor allow you to attach an XML schema or DTD from the toolbar.

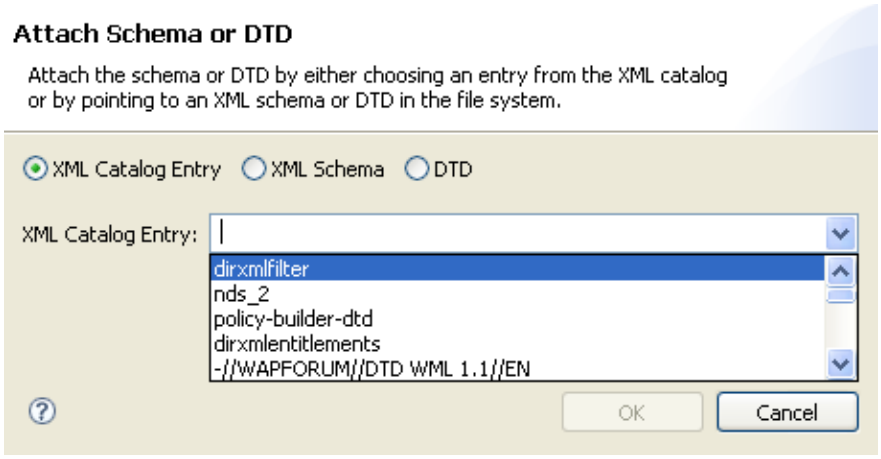
1. In the XML Source or XML Tree editor, click Attach .

   This opens the Attache Schema or DTD dialog box.

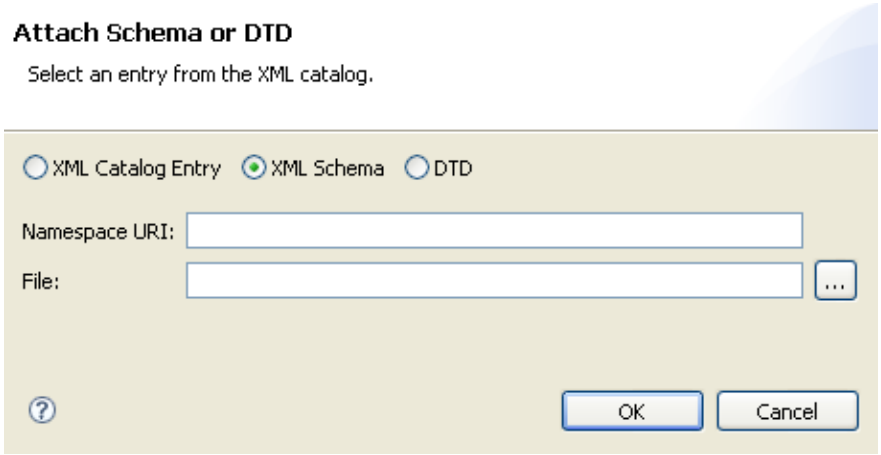
2 Specify the data source, either *XML Catalog Entry*, *XML Schema*, or *DTD*, by clicking the appropriate radio button.

3 Provide the necessary information for the selected data source, then click *OK*.

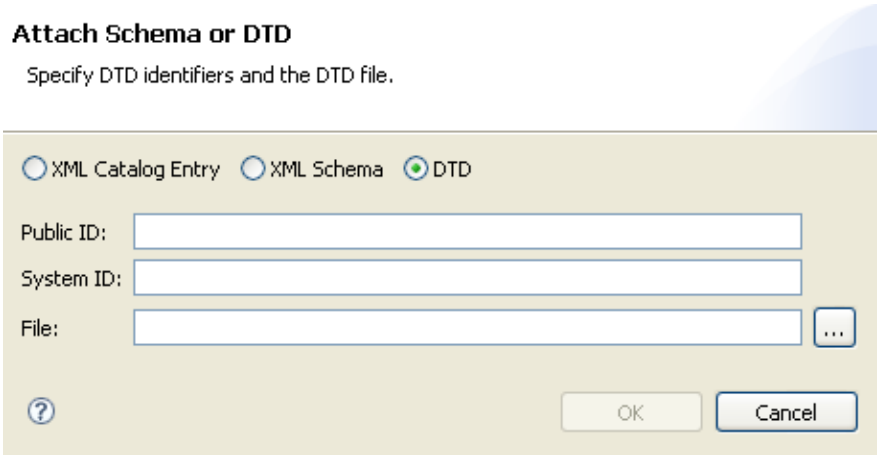
XML Catalog Entry: Choose the approporiate entry from the *XML Catalog Entry* dropdown list.



XML Schema: Specify the namespace URI and the schema file.



DTD: Specify the Public and System IDs and the DTD file.



16.5 Setting XML Editor Preferences

You can customize some Novell XML Editor behaviors by setting preferences. You access the preferences page through *Windows > Preferences > Web and XML*. You can learn more about these preferences in the following sections:

- Section 20.5.4, “Task Tags,” on page 486
- Section 20.5.5, “XML Catalog,” on page 486
- Section 20.5.6, “XML Files,” on page 488

Tools

17

Designer provides a variety of additional tools to help you manage IDM projects. This section describes the tasks available through these tools and services and includes the following topics:

- Section 17.1, “Converting Earlier Projects,” on page 373
- Section 17.2, “Migrating Driver Configuration Data To a New Server,” on page 376
- Section 17.3, “Opening a Web Browser,” on page 382
- Section 17.4, “Launching iManager,” on page 382
- Section 17.5, “Checking Your Projects,” on page 383
- Section 17.6, “Managing Directory Objects,” on page 389
- Section 17.7, “Configuring TLS for eDir-to-eDir Drivers,” on page 392
- Section 17.8, “Using DS Trace,” on page 395
- Section 17.9, “Working with Generic Resources,” on page 399
- Section 17.10, “Updating Designer,” on page 401

For information on managing workspaces, perspectives, and views, see Workspaces, Perspectives, and Views (http://www.novell.com/documentation/designer30/designer_intro/data/projmanageoverview.html) in the *Understanding Designer for Identity Manager*.

17.1 Converting Earlier Projects

Previous Designer workspaces are not compatible with Designer 3.5 and above. Designer stores projects and configuration information in a workspace. These workspaces are not compatible from one version of Designer to another. You need to point Designer 3.5 to a new workspace, and not to a workspace used by a previous version of Designer.

If you have Designer 2.x or 3.5 Milestone projects, import the projects into Designer 3.5 (*File > Import > Project from File System*). Be sure the *Copy project into the workspace* option is selected. Importing the project runs the Project Converter Wizard, making them compatible with Designer 3.5 architecture and placing them under your designated Designer 3 workspace directory (`designer_workspace` by default).

- Section 17.1.1, “Converting Projects With the Project Converter Wizard,” on page 373
- Section 17.1.2, “Running Later Projects on Earlier Designer Versions,” on page 376

17.1.1 Converting Projects With the Project Converter Wizard

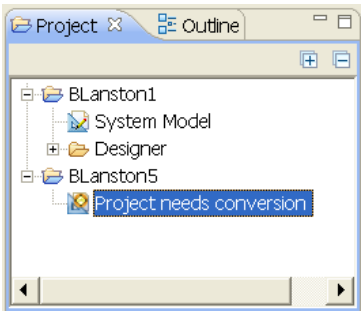
To convert projects that were not open in an editor when Designer was closed, open the project by doing one of the following:

- Double-click the project in the Project view.
- Right-click the project in the Project view, then select *Open*.

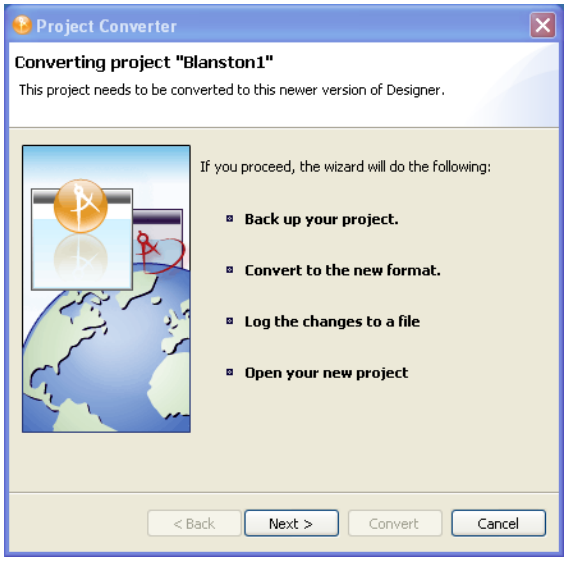
Although you can open a project in the Navigator view by clicking the project's `.proj` file, Novell recommends that you use the Project view instead. Otherwise, the Navigator view takes you into the raw file system.

To convert an earlier project:

1. In the Project view, expand the project, then double-click *Project needs conversion*.

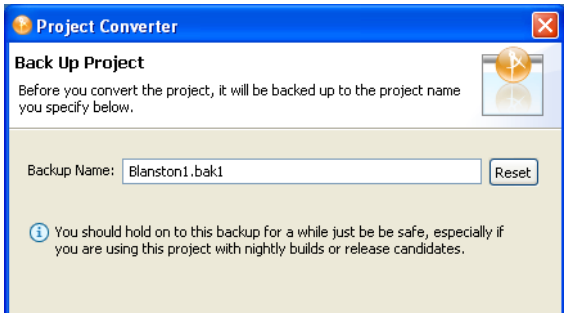


2. Designer opens the project in the Project Converter Wizard. Review the steps, then click *Next*.



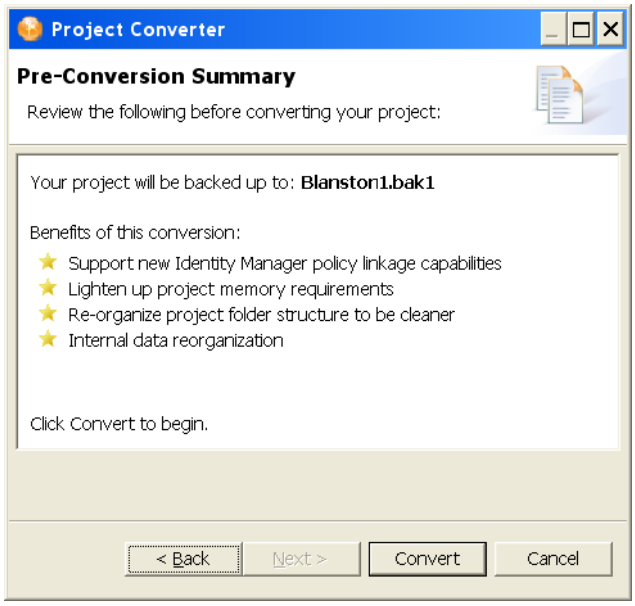
3. Name the project, then click *Next*.

   The Project Converter backs up your project before converting. You can accept or change the default name.



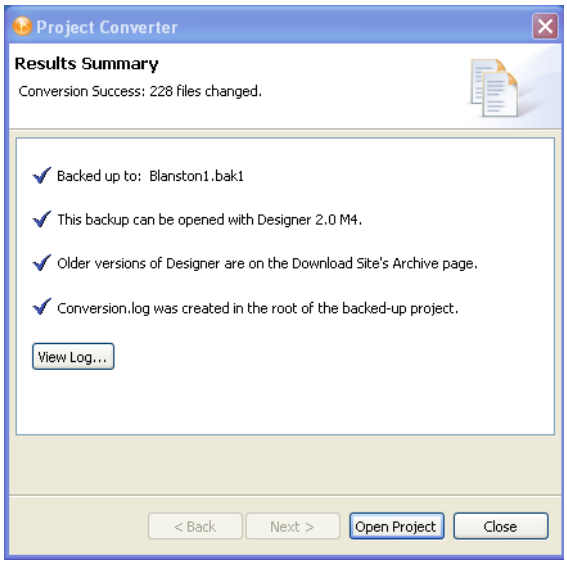
4 (Optional) If you edit the name but want to return to the default, click *Reset*.

5 Convert the project by clicking *Convert*.



The converter changes, adds, and removes references, attributes, and elements. It might also create new files or delete old ones. It converts the project file to the new, correct file format. A progress bar displays during the backup and conversion. Converting very large projects might take a few seconds.

6 View the conversion log by clicking *View Log*.



The `conversion.log` file is in the project folder in the `Workspace` directory (for example, `c:\documents and settings\skopai\digitalairlines\conversion.log`).

7 Open the project.

Regardless of the internal format (for example, Designer 1.2 or Designer 3.5), Designer always deploys to the proper format of the target Identity Manager environment.

WARNING: The converter cannot convert Designer 3.0 M2 projects. Follow the information in the "Click here" portion of the No Converter screen to manually bring the old project data over to Designer 3.5.

The converter ensures only forward compatibility. It does not ensure backward compatibility. You can't convert to a newer release of Designer and then convert back if you want to return to an older release. If you need to go back to an earlier format, use the project's backup file.

17.1.2 Running Later Projects on Earlier Designer Versions

Designer 2.0 or later (including Designer 3.5) does not let you load a project created in later Designer versions if the file format has changed between versions.

For example, Designer 2.0 and 2.1.1 and Designer 3 use different formats. If you create a project in 3.5 and above, you cannot open that project in 2.0. Instead, a message informs you that you can't open the project because it requires the later version of Designer.

Even if the version formats are the same, it isn't a good practice to run later projects on earlier versions of Designer. Later versions of Designer have additional bug fixes and features that might make more use of the existing format. Therefore, going back to an earlier Designer version could result in a degraded experience.

17.2 Migrating Driver Configuration Data To a New Server

If you have added a new server (right-click the Identity Vault and select *New* > *Server*), you might need to migrate the server data from an existing driver set to the new server. You can do this in one of three ways: migrate the driver set to a new server in the same tree, migrate the driver set to a server in a different tree, or migrate server data for a single driver. After the server data is migrated, you must redeploy the driver set to the new server in order for the server to become active.

- Section 17.2.1, "Using the Server Migration Wizard To Migrate the Driver Set," on page 376
- Section 17.2.2, "Migrating a Driver Set To a Server in a Different Tree," on page 378
- Section 17.2.3, "Migrating Server Data for Each Driver," on page 380

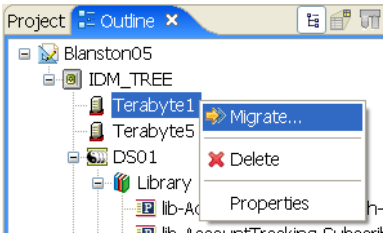
17.2.1 Using the Server Migration Wizard To Migrate the Driver Set

Use the Server Migration Wizard to migrate server-specific data in an existing driver set to a new server. The Server Migration Wizard copies the following server-specific information for the driver set and associated drivers:

- Global configuration values (GCVs)
- Engine control values (ECVs)
- Named passwords

- Driver authentication information
- Driver startup option
- Driver parameters

1 From the Outline view, right-click the server with the associated driver set you want to migrate, then select *Migrate*.



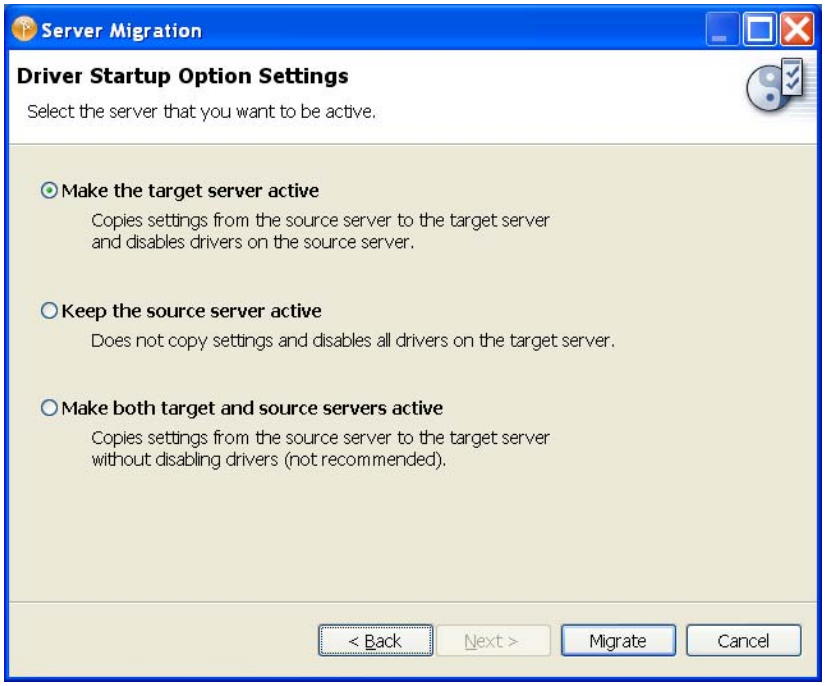
The Server Migration overview page explains that you are migrating a driver set from its source server to a target server along with its server-specific data.

2 Click *Next*.

3 On the Select Target Server page, select the server targeted for driver set migration and select *Next*.

The Target Server list shows only servers that are not presently associated with any driver set and have an Identity Manager version that is equal to or newer than the source server.

4 In the Driver Startup Option Settings page, select the server that you want to be active.



The default selection is *Make the target server active*. This option copies the current driver startup settings from the source server to the target server and disables all of the drivers on the source server.

The *Keep the source server active* option copies the current driver startup settings from the source server to the target server and then disables the drivers on the target server.

The *Make both target and source servers active* option copies the current driver startup settings from the source server to the target server and does not disable any drivers on either server. This option is not recommended; having all service queues active on both servers causes the servers to run the same tasks, which can produce unpredictable behavior.

Settings in the Driver Startup Option Settings page only affect the DirXML-DriverStartOption attribute on drivers and not the migration of other server data. You can also set the driver startup options on the driver's *Properties* > *Driver Configuration* > *Startup Options* tab. Driver startup options are *Auto Start*, *Manual*, and *Disabled*.

5. Select *Migrate*.

   The wizard copies the server-specific information for the driver set and associated drivers to the target server while displaying a progress bar. When the migration completes, you see `The server has been successfully migrated!`

6. Click *Close* to close the Server Migration Wizard.
7. After the wizard closes, right-click the driver set object in the Outline view and select *Live* > *Deploy*.
8. If necessary, fill in any needed information in the Identity Vault Credentials window to authenticate to the Identity Vault, then click *OK*.

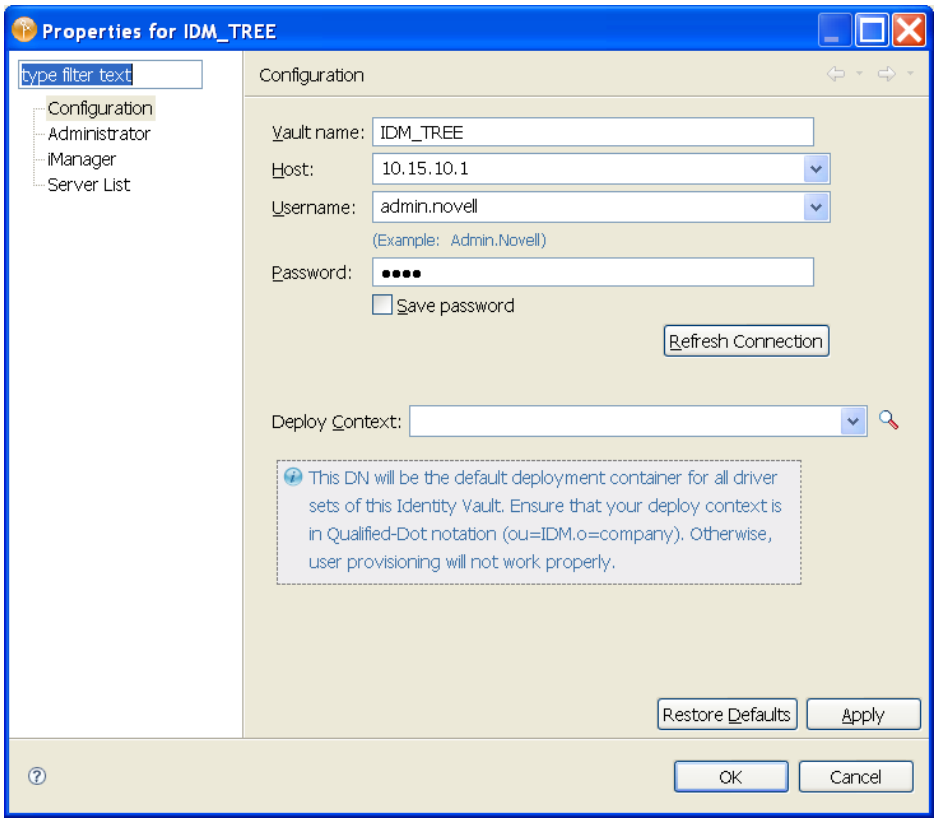
   You see the Operation In Progress window, followed by the Deployment Summary page, which shows what is being deployed to the Identity Vault.

9. Click *Deploy*.
10. If you see errors on the Deployment Results page, click the error to see a summary of the cause and possible solutions. Click *OK* to close.

17.2.2 Migrating a Driver Set To a Server in a Different Tree

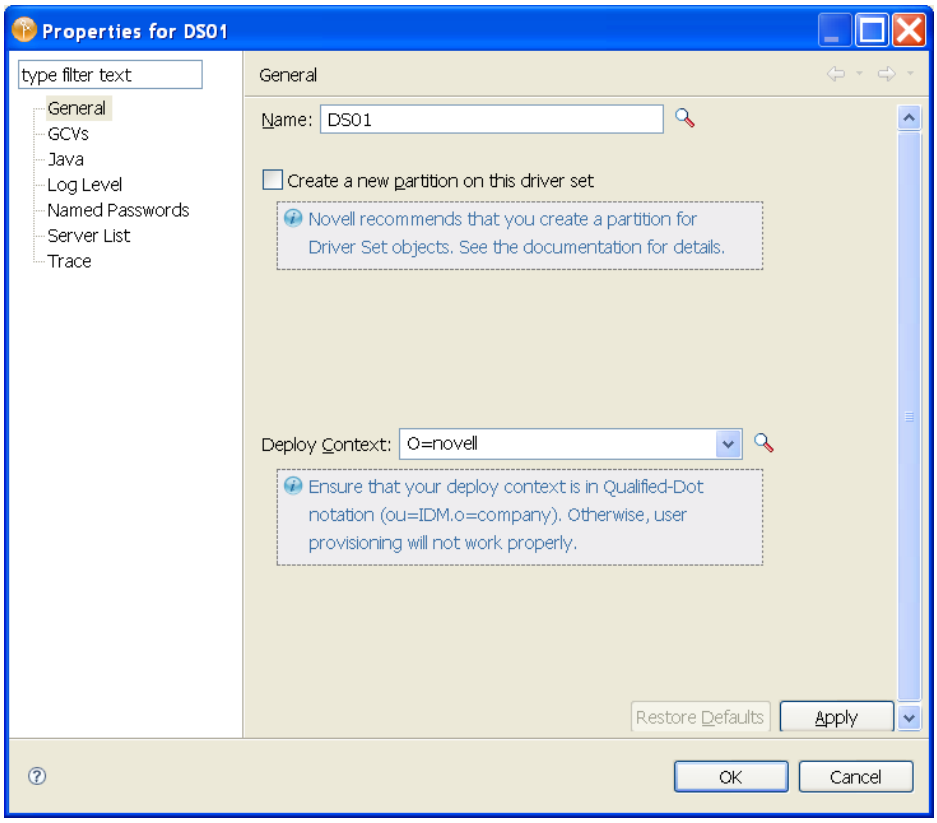
For this procedure, assume that you have created a new tree and server, but you want to use an existing driver set.

1. Right-click the Identity Vault in the Modeler or Outline view and select *Properties*.
2. In the Configuration section, edit the *Host*, *Username*, and *Password* entries to connect to the new tree, then click *OK*.



3 Right-click the driver set in the Modeler or Outline view and select *Properties*.

4 In the General section, edit the *Deploy Context* to reflect the container where you want to store the driver set. Type the name of the correct container or use the Browse button to find the new container, then click *OK*.



5 Right-click the server object in the Outline view and select *Properties*.

6 Under the *General* > *Properties* section, edit the *Name* and *Context* entries to match the server in the new tree, then click *OK*.

7 Redeploy the driver set to the new server by right-clicking the driver set object in the Modeler or Outline view and selecting *Live* > *Deploy*.

You see the Operation In Progress window, followed by the Deployment Summary page, which shows what is being deployed to the new Identity Vault.

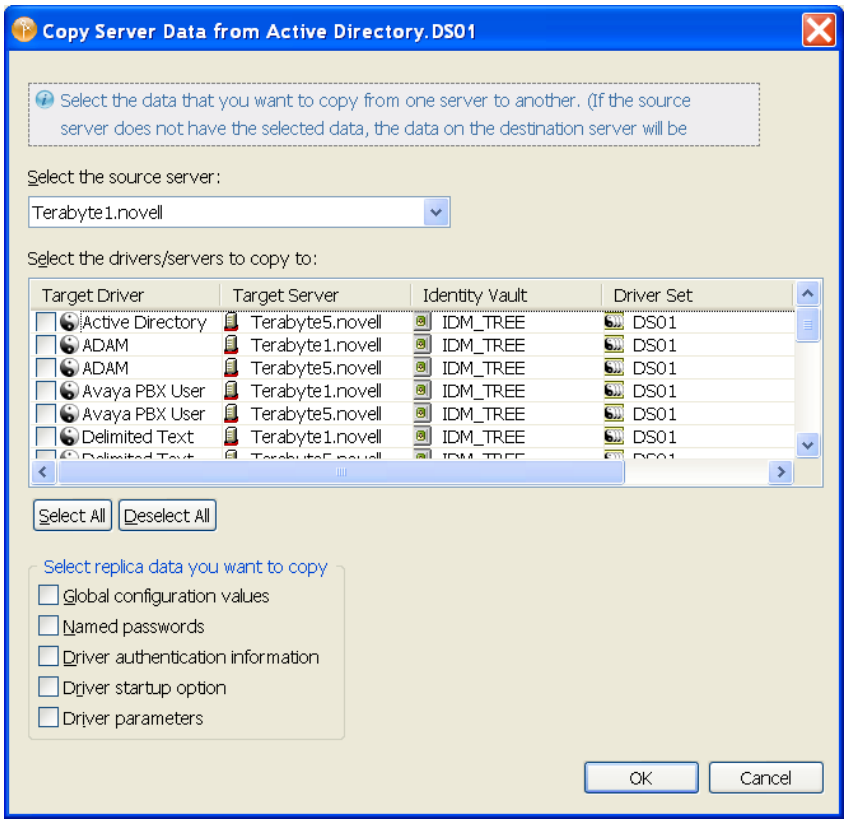
8 Click *Deploy*.

9 If you see errors on the Deployment Results page, click the error to see a summary of the cause and possible solutions. Otherwise, click *OK* to close. All server-specific data for the driver set is copied to the new server on the new tree.

17.2.3 Migrating Server Data for Each Driver

Although using the Server Migration Wizard is the preferred method, you can also migrate server data for a single driver in the driver set. You can either perform this action for each driver in the driver set, or use the Server Migration Wizard as described in Section 17.2.1, “Using the Server Migration Wizard To Migrate the Driver Set,” on page 376.

1 Right-click a driver in the Outline view and select *Copy* > *Server-Specific Settings*.



2 In the Copy Server Data from Driver.Driver Set window, select the source server. This is the server whose data is copied to the selected targets.

3 Under the *Select the drivers/servers to copy to* entry, select the target driver or drivers on the target server that you want to copy to. This example selects the Active Directory driver as the target driver on the Terabyte5.novell target server.

IMPORTANT: Some server data is specific to a driver type, while some data, like the driver startup option, is not. Know what you want to accomplish before copying one driver's server data to other driver types. Otherwise, drivers on the target server might behave eratically or fail.

---

4 In the *Select replica data you want to copy* section, select the data you want to copy to the target server. The copied data includes:

- Global configuration values (GCVs)
- Named passwords
- Driver authentication information
- Driver startup option
- Driver parameters

5 After you select the data, click *OK*, then click *OK* at the Complete window.

You must perform this action for each driver in the driver set, or use the Server Migration Wizard.

17.3 Opening a Web Browser

You can open a Web browser within the Designer utility. The Web browser button is available from the main toolbar.

When you first launch the browser, you are prompted for a home page. After you enter the URL, it is stored in Preferences.

To change the URL:

1. Select *Window* > *Preferences*.
2. Select *Designer for IDM* .
3. Click the *Browser* tab.
4. Type the new URL, then click *OK*.

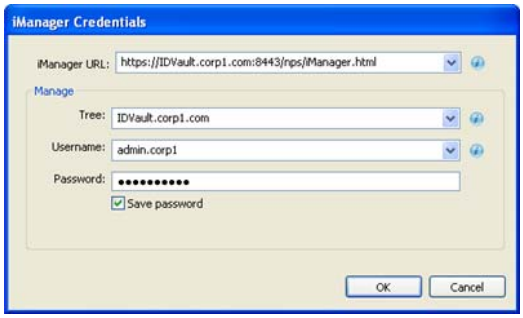
You can also open an internal Web browser view by selecting *Window* > *Show View* > *Other* and then selecting the *Internal Web Browser* option under the *General* heading.

17.4 Launching iManager

To launch iManager from within Designer:

1. Right-click the Identity Vault, then select *Live* > *iManager*.

   You can also select *Tools* > *iManager*.
2. In the iManager Credentials dialog box, specify the appropriate iManager URL and user credentials to access iManager.



   You must specify the iManager URL along with a server name (or IP address) with a replica of the directory tree, username, and password.

   Select Save password to store the credentials in a history.
3. Click *OK*.

17.5 Checking Your Projects

Designer provides the Project Checker so you can check your project. The Project Checker checks for proper design, contexts, server associations, policies, missing user data, and dependency problems that can cause a project deployment into the Identity Vault to fail. You can check a project at any time, but you should definitely run the Project Checker before deploying it.

- Section 17.5.1, “Checking a Project,” on page 383
- Section 17.5.2, “Customizing the Project Checker,” on page 384
- Section 17.5.3, “Items That Are Checked,” on page 386

NOTE: Project Checker only checks the objects in Designer; it does not check the current objects in the Identity Vault.

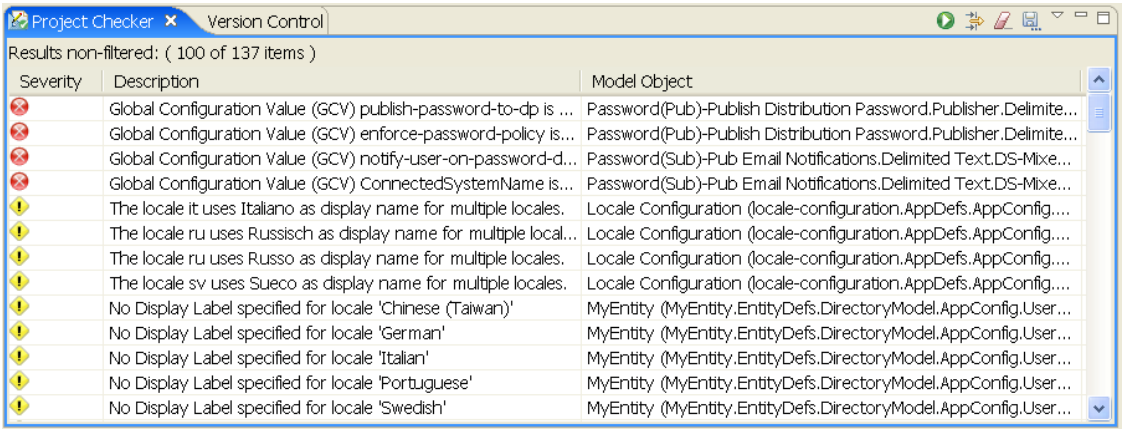
17.5.1 Checking a Project

1. In the Project or Outline view, select the project, then select the *Launch Project Checker* icon in the Designer toolbar.

   The Project Checker is also available from the *Window* > *Show View* menu.

2. Click the *Run the Project Checker* icon .

   If you haven’t saved the project, Designer prompts you to save it.



The Project Checker displays a list of versioning conflicts, errors, warnings, and information messages about the project. In the Project Checker view, you can do the following:

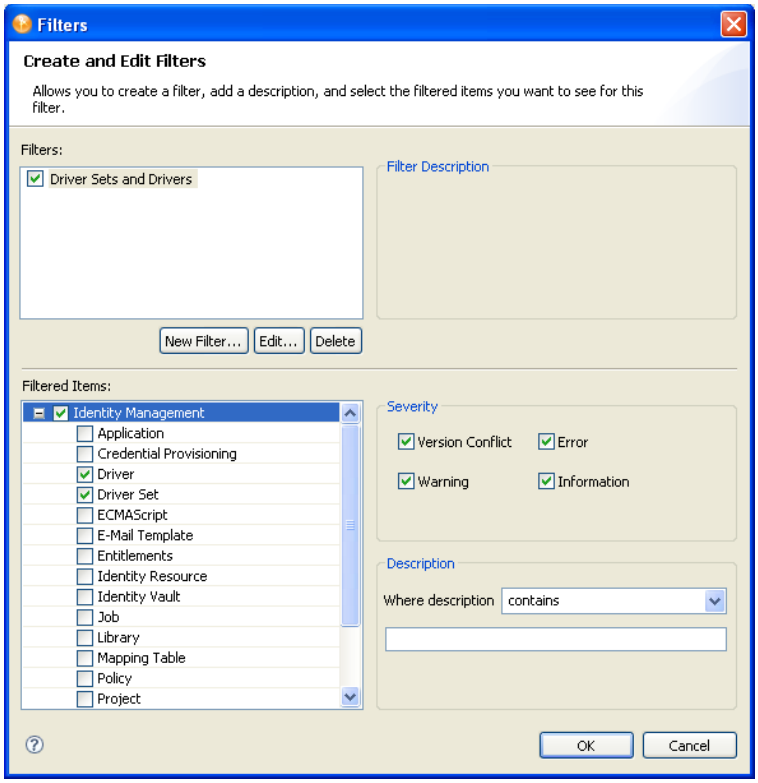
Action	Description
See detailed information about a list item	Double-click a list entry to open a properties page that displays the following information about the entry: ◆ The message severity ◆ A message description ◆ The model object that caused the message ◆ The line number where the problem occurred, if available ◆ Details about the message, if available ◆ A recommended solution for the message, if available
Sort the list	Click on any header in the Project Checker to sort the entry list on that parameter: Severity, Description, and Model Object. By default, Project Checker sorts entries by severity in descending order (most current at the top of the list.)
Filter the list	Click the Configure Filters icon to customize the Project Checker. For more information, see “Customizing the Project Checker” on page 384.
Clear the list	Click the Clear Results icon to clear the Project Checker entry list.
Save the list	Click the Save Project Checker Results to a File icon to save the current Project Checker entry list to a text file so you can review it off-line.
Menu options	Click the Menu icon to select one of the following: ◆ View the messages in a hierarchical layout, according to functions (Identity Manager, provisioning, etc.) ◆ View the messages in a flat layout (default). ◆ Automatically check the project when you save it. ◆ Configure filters ◆ View the Project Checker’s Preferences page.

For more information about the icons in the Project Checker view taskbar, located in the upper-right corner of the Project Checker view, see Section 17.5, “Checking Your Projects,” on page 383.

17.5.2 Customizing the Project Checker

You can customize the Project Checker by creating and editing filters. The filters allow you to receive messages about the items you want to verify. You can create multiple filters, but only one filter can be used at a time.

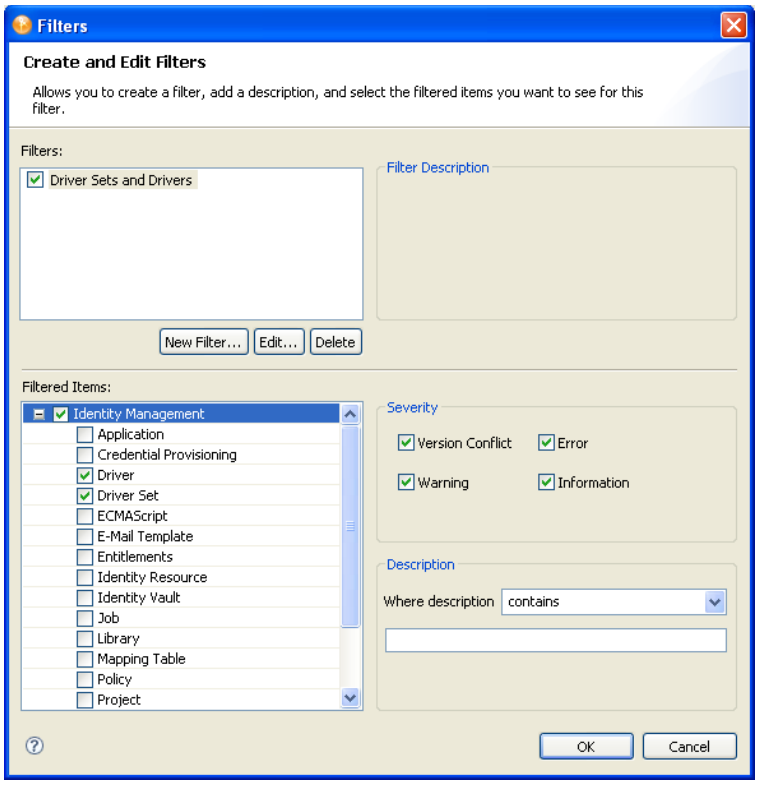
Figure 17-1 *Customizing the Project Checker*



To create a filter:

1. In the Project Checker, click the *Configure Filters* icon .
2. Click *New Filter*.
3. Specify a name and description for the filter.

   You can select which items are checked, what types of messages are returned about the items, and use key words to limit the messages returned. For example, you can search for all messages about the Driver Set and Driver objects that contain the word “attribute.”



4 Click *OK*.

To edit the name and description of the filter:

1 Select the filter, then click *Edit*.

2 After you have completed the changes, click *OK*.

To delete a filter:

1 Select the filter.

2 Click *Delete*.

17.5.3 Items That Are Checked

The Project Checker looks at specific items in the project. It checks the items in the User Application as well as the rest of Identity Manager.

The following table describes the specific items that are checked. The list increases with each release of Designer.

Table 17-1 *Identity Manager Items That Are Checked*

Item	Description
Driver	◆ Checks for the presence of a Schema Mapping policy. ◆ Checks for an invalid Active Directory container. ◆ Checks the trace level setting. If it is set to more than 0, an informational message is displayed. ◆ Checks to see if the LoopBack driver is being used instead of the eDirectory driver. ◆ Verifies that the GUID attribute is set to synchronize on the Subscriber channel. ◆ Verifies that the GUID attribute is not set to synchronize on the Publisher channel. ◆ Checks the classes on the Publisher and Subscriber channels that are set to *Ignore* and verifies that the attributes for these classes are not set to *Synchronize*. ◆ Checks for the presence of a filter and makes sure it is not empty. ◆ Checks to make sure that the Publisher Placement policy does not contain *set operation destination DN* or *set xml attribute* operations. ◆ Checks for the presence of a Publisher Placement policy. ◆ Checks to make sure that no policy on the Publisher channel contains *set operation destination DN* or *set xml attribute* operations. ◆ Checks to make sure that the Subscriber Placement policy does not contain *set operation destination DN* or *set xml attribute* operations. ◆ Checks to see if the Subscriber Placement policy is missing. ◆ Checks to make sure that no policy on the Subscriber channel contains *set operation destination DN* or *set xml attribute* operations. ◆ Checks to make sure that the npsmDistributionPassword attribute and the public-private key pair attributes do not simultaneously exist in the User class. ◆ Checks to make sure that the authentication method on the Active Directory driver is set to *Negotiate* when synchronizing passwords. ◆ Checks the filter for invalid data. ◆ Checks the driver to see if it is publishing both NDS® and Distribution passwords. If it is, this is an invalid setting. ◆ Checks for the presence of the nspmDistributionPassword attribute in the User class in the Filter, if password synchronization is enabled. ◆ Checks that the nspmDistributionPassword attribute is set to sync or notify, if password synchronization is enabled.
Driver Set	◆ Checks to make sure that the deployment context for the Driver Set object is set. ◆ Checks to make sure that a server object is associated with the Driver Set object.
E-mail Template	Checks to see if the e-mail notification template is empty.

Item	Description
Entitlements	◆ Checks to see if the driver supports entitlements. ◆ Checks to see if the attribute DirXML-EntitlementRef is added to the Subscriber channel, if there are policies that use entitlements in the driver. The DirXML-EntitlementRef must be set to Notify or Synchronize for the entitlements to work.
ECMAScript	Checks to see that the ECMAScript object can run only on Identity Manager 3.6.
Identity Vault	◆ Checks to see if the username to authenticate to the Identity Vault is missing. ◆ Checks to see if the hostname for the Identity Vault server is missing. ◆ Checks to see that the password for the user is not stored in the project.
Job	Checks to see that the job object can run only on Identity Manager 3.6.
Library	Checks to see that the library object can run only on Identity Manager 3.6.
Mapping Table	◆ Checks to see that the mapping table object can run only on Identity Manager 3.6. ◆ Checks to see if there is an empty column name. ◆ Checks to see if there is a duplicate column name.
Policy	◆ If there are global configuration values in the policy, it checks to make sure they exist on the Driver or Driver Set object. ◆ Checks to see if local variables are defined before they are used. ◆ Validates the policy against the DTD.
Schema	◆ Checks to see if the class is missing from the schema. ◆ Checks to see if the attribute is missing from the schema. ◆ Checks to see if the attribute for the class is missing from the schema.

Table 17-2 *Provisioning Items That Are Checked*

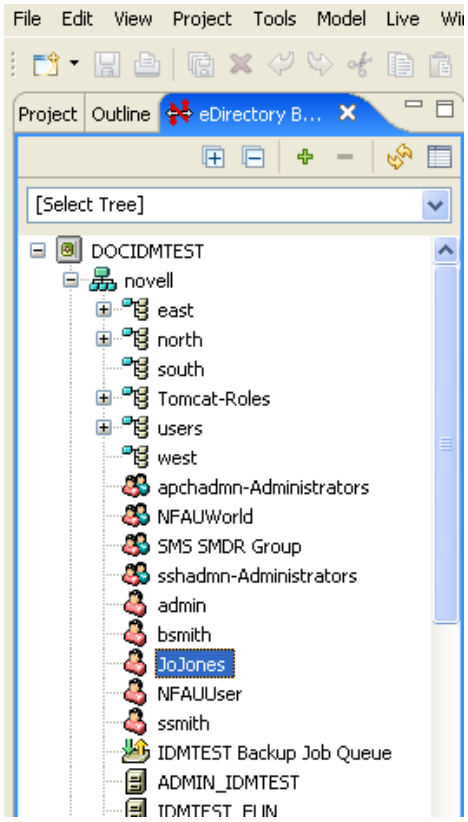
Item	Description
Configuration	Verifies that the XML is well-formed and complies with the schema that defines the elements needed for entities, attributes, lists, relationships, and so on.
Entity	◆ Checks every entity to ensure that references to other entities and global lists are valid. ◆ Ensures that every entity has at least one attribute defined.
List	Ensures that every local and global list contains at least one item.
Org Chart Relationship	Verifies that the entities and attributes of a relationship have been deployed.
Provisioning Request Definition	Verifies that a workflow follows rules for activities and flow paths.

17.6 Managing Directory Objects

Sometimes it is necessary to locate or modify objects during your project development. Rather than using a separate management interface, you can use the eDirectory Browser to browse to and edit attributes of objects in the following locations:

- The Identity Vault
- Other eDirectory trees

Figure 17-2 *Sample eDirectory Browser View*



To launch the eDirectory Browser, use the tool-based method or the task-based method. The method you use is largely a matter of preference and the target directory that you will browse.

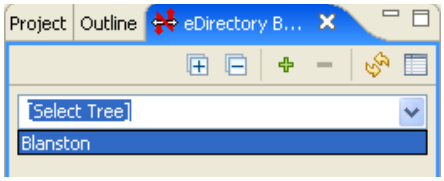
- Section 17.6.1, “Tool-Based Method,” on page 389
- Section 17.6.2, “Task-Based Method,” on page 390
- Section 17.6.3, “Browse, View, or Modify Object Attributes,” on page 391

17.6.1 Tool-Based Method

To access an eDirectory tree other than the Identity Vault, or if there is no Identity Vault defined in your current project, use the tool-based method. You can always launch the eDirectory Browser from the *Tools* menu, even when an Identity Vault isn’t selected in the Modeler.

1 Select *Tools* > *Manage Directory*.

2 Select an Identity Vault.



3 In the Login Credentials dialog box, provide the appropriate authentication credentials, then click *OK*.

4 Access information by using icons on the eDirectory Browser's toolbar

Table 17-3 *Icons on the eDirectory Browser toolbar*

Icon	Descriptions
	Expands all containers in the currently selected tree. **IMPORTANT:** This might be a time-consuming operation if you have a million-object tree.
	Collapses all expanded containers in the currently selected tree.
	Adds a new custom tree, which persists across sessions.
	Removes trees previously added with the Add Tree operation. Automatically discovered trees cannot be removed.
	Refreshes the currently active tree.
	Displays an object's properties.

Expansion states and selection states are persistent between sessions per tree.

17.6.2 Task-Based Method

To use the eDirectory Browser to browse the Identity Vault in your current project:

1 In the Modeler or Outline view, select the Identity Vault, then select *Live* > *Manage Directory*.

You can also right-click the Identity Vault object and select *Live* > *Manage Directory*.

2 In the Login Credentials dialog box, provide the appropriate Identity Vault authentication credentials, then click *OK*.

If you previously saved your authentication credentials, eDirectory Browser automatically populates the fields.

3 In the eDirectory Browser, browse to and select an object.

The eDirectory Browser automatically displays the Identity Vault directory structure.

To use the eDirectory Browser to browse an eDirectory application in your current project:

1 In the Modeler, select the eDirectory application you want to browse, then select *Live* > *Manage Directory*.

2 In the Login Credentials dialog box, provide the appropriate eDirectory authentication credentials, then click *OK*.

3 In the eDirectory Browser view, browse to and select an object.

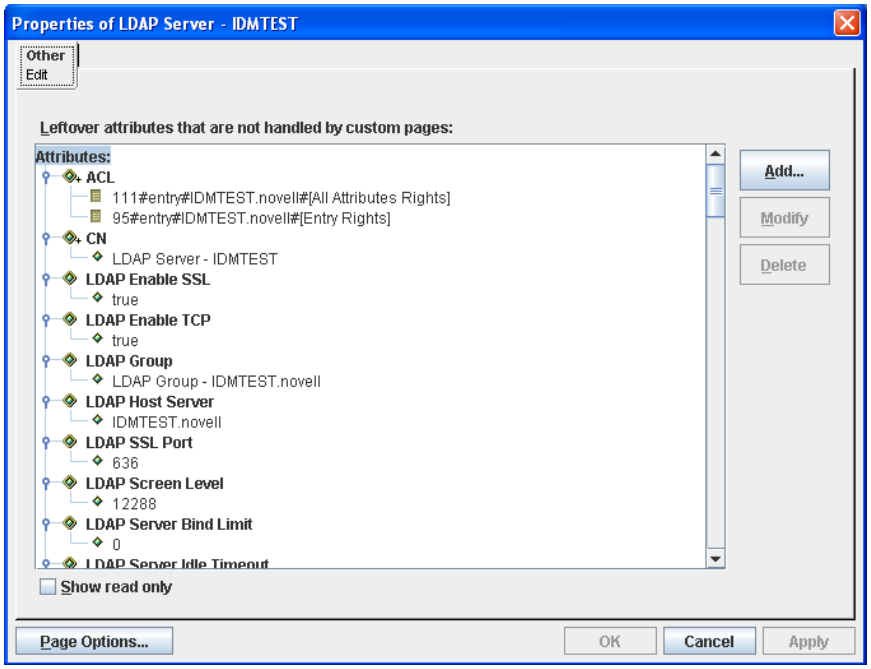
17.6.3 Browse, View, or Modify Object Attributes

After you have populated eDirectory Browser with one or more directories, you can browse the directory tree for specific objects, and view and modify object attributes.

Table 17-4 *Objects That You Can Modify*

Object	Description
Container Object	Double-click to expand a collapsed container or to collapse an expanded container. Right-click and select *Properties* to open that object's attributes page. Select the object and click the *Open properties of this object* icon in the Action bar.
Leaf Object	Double-click (or right-click and select *Properties*) to open that object's attributes page. Select the object and click the *Open properties of this object* icon in the Action bar.

Figure 17-3 *eDirectory Browser Attributes List*



17.7 Configuring TLS for eDir-to-eDir Drivers

- Section 17.7.1, “Prerequisites,” on page 392
- Section 17.7.2, “Enabling TLS,” on page 392
- Section 17.7.3, “Creating Certificates,” on page 394

17.7.1 Prerequisites

- Identity Vaults exist in your physical network tree as well as in the Modeler.
- Each Identity Vault is set up. Otherwise, you are prompted for setup information when you try to create certificates.
- Each driver set is associated with a server.
- Using the eDir-to-eDir driver’s General property page, verify that each driver has a name and a deploy context. The context might be inherited from the driver set.
- The eDir-to-eDir drivers have been deployed. Otherwise, Designer cannot create certificates.

  To find out whether the driver has been deployed:

  1. Right-click the eDir-to-eDir driver.
  2. Click *Live* > *Deploy*.
  3. In the eDir-eDir Driver Deployment dialog box, click *No*.

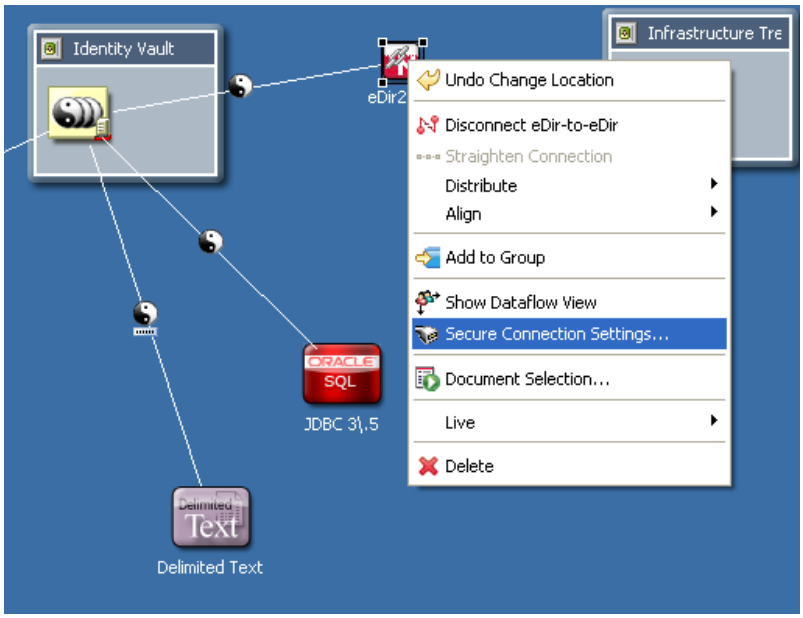
  If the driver has been deployed, the *Compare Status* field in the Deployment Summary dialog box displays *Equal* or *Unequal*. Otherwise, the field displays *Not Deployed*.

  After objects have been deployed, the objects should show as equal unless passwords are set in eDirectory that are not set in Designer. Designer does not deploy passwords unless they are specifically set in Designer. This exception prevents overwriting passwords in eDirectory because Designer cannot import them.

17.7.2 Enabling TLS

1 Launch the TLS Configuration dialog box.

A common way to launch the dialog box is to right-click the eDir-to-eDir application, then click *Secure Connection Settings*.

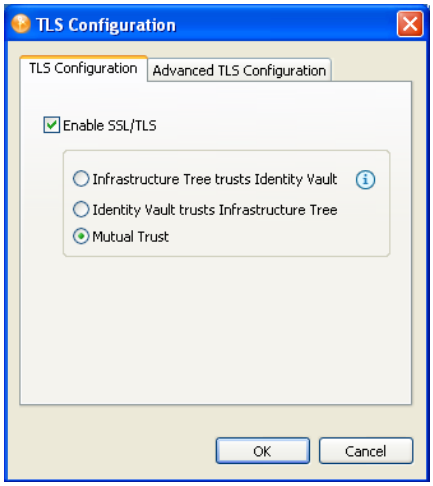


Other launch points:

- Select the eDir-to-eDir application, then click *Model* > *eDir-to-eDir* > *Secure Connection Settings*.
- Right-click *eDir-to-eDir* in the Outline view, then click *Secure Connection Settings*.
- Right-click an eDir-to-eDir driver, click *Properties* > *Driver Configuration* > *Authentication*, then click *Configure TLS*.

  The *Configure TLS* button displays only on eDir-to-eDir driver pages.

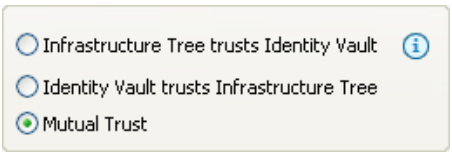
2 Click *Enable SSL/TLS*.



3 (Optional) Use the *Advanced TLS Configuration* to select key size, hash algorithm, and validity period.

The validity period is important for when a certificate has expired and you need to overwrite or create a new one.

4 Select a direction of trust.



These options apply to certificates that Novell creates for eDirectory. The options do not apply to third-party security certificates.

The default is Mutual Trust, which is considered to be the most secure.

Unless you want to use the certificate for authentication, the option that you select doesn't matter. If only encryption is important, you can select any one of the three options.

If authentication is important, select the option that gives you the appropriate trust.

Scenario: JJ Infrastructure Tree Trusts JT ID Vault. JJ Infrastructure Tree is the organizational Certificate Authority. JJ Infrastructure Tree signed a certificate and placed it in JT IDVault. JT ID Vault trusts JJ Infrastructure Tree. The two vaults synchronize data through a secure connection.

If the two vaults break their trusted relationship, JJ Infrastructure Tree can prevent sensitive data from being synchronized by revoking its certificate.

Scenario: JT ID Vault Trusts JJ Infrastructure Tree. JJ Infrastructure Tree creates two certificates. One is placed in JJ Infrastructure Tree, and the other is placed in JT ID Vault. The two vaults synchronize data through a secure connection.

If the two vaults break their trusted relationship, JJ Infrastructure Tree can prevent sensitive data from being synchronized by revoking its certificate.

Scenario: Mutual Trust. JT ID Vault and JJ Infrastructure Tree both sign certificates.

5 Click *OK*.

After you click *OK*, Designer does the following:

- Modifies both eDirectory drivers.
- Locks the *User ID* field, which displays on the driver configuration's Authentication page, because both drivers must use that field.

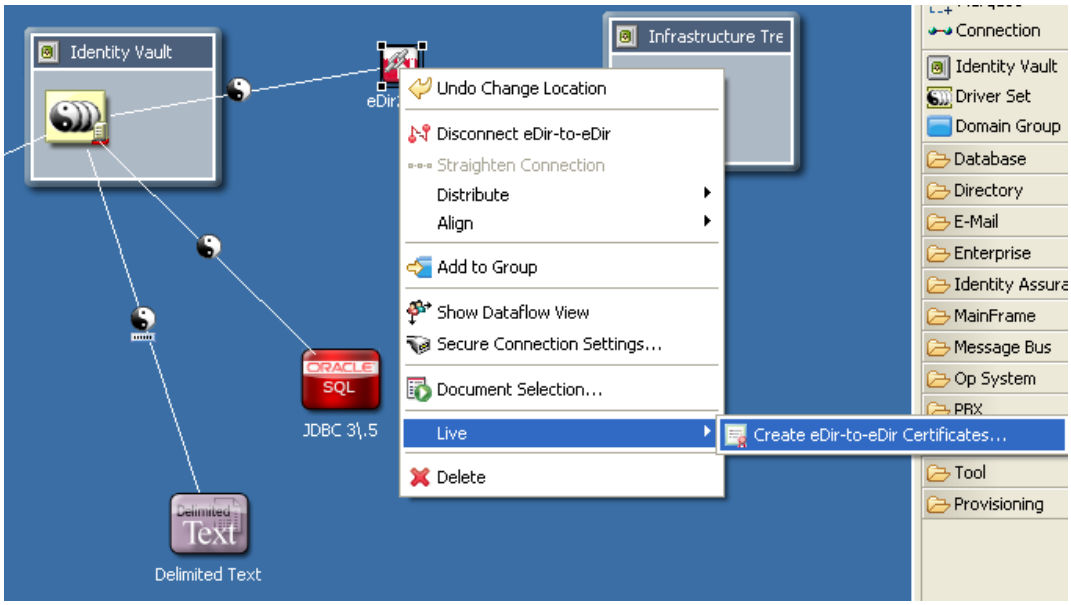
You can enable or configure TLS without immediately deploying the drivers. You can turn the settings on. However, you can't create SSL/TLS certificates unless the drivers have been deployed into their respective Identity Vaults. If you enable SSL/TLS but want to create certificates later, you can do so. When you later deploy the eDir-to-eDir drivers, Designer guides you through steps to automatically create certificates.

17.7.3 Creating Certificates

A driver's Properties page enables you to configure a driver so that you can deploy it. Similarly, the *Enable SSL/TLS* option enables you to set up your configuration for TLS and then, when you are ready, create and deploy the certificates. When you deploy a configured driver set or select *Create eDir-to-eDir Certificates*, Designer creates the certificates in the directory.

This section assumes that you have enabled and configured SSL/TLS for the deployed eDir-to-eDir drivers.

1 Right-click the eDir2eDir application.



2 Click *Live* > *Create eDir-to-eDir Certificates*.

You can also do one of the following:

- Right-click the eDir2eDir object in the Outline view, then click *Create eDir-to-eDir Certificates*.
- The first time that you enable and configure SSL/TLS on driver's *Authentication* tab, click *OK*, then follow prompts. A Create Certificates dialog box appears. Click *Yes*.

Scenario: Enabling TLS. TLS has not been enabled. Sandy selects *Live* > *Create eDir-to-eDir Certificates*. Designer prompts Sandy to enable SSL/TLS. Sandy clicks *OK*, enables TLS, selects a direction of trust, and clicks *OK*. Designer creates certificates.

Scenario: Deploying eDir-to-eDir Drivers. Sandy has configured the eDir-to-eDir drivers and the driver set. A context displays in the driver set's *Deploy Context* field. Sandy is ready to deploy the driver set.

Sandy right-clicks the driver set, then clicks *Live* > *Deploy Driver Set*. Designer prompts Sandy to deploy both eDirectory drivers. (Otherwise, Designer can't successfully create certificates.) Sandy clicks *Yes*. Designer builds a deployment summary, then lists items that are associated with the Identity Vaults and will be deployed. To deploy the drivers, Sandy clicks *Deploy*.

Because the driver set is already configured, Designer creates the certificates.

For additional information on eDir-to-eDir certificates, see eDir-to-eDir SSL/TLS in *Preferences*.

17.8 Using DS Trace

Designer provides DS Trace so you can monitor DirXML events in your Identity Manager environment. DirXML events constitute those events accessible by using the "dirXML" and "dirXML Drivers" switches in eDirectory's DS Trace service.

Designer uses LDAP to obtain this information from the Identity Vault. By default, it uses the default LDAP ports (389 or 636) to establish a connection. If your LDAP service runs on non-standard ports, make sure you specify the correct ports.

DS Trace lets you view both live DS Trace logs, and create and view stored DS Trace log files.

- Section 17.8.1, “Viewing DS Trace Live,” on page 396
- Section 17.8.2, “Creating a DS Trace Log File,” on page 398
- Section 17.8.3, “Viewing a DS Trace Log File,” on page 398

NOTE: The DS Trace view is not the same as the Trace view, which provides information about Designer functionality. For information on the Trace view, see “Trace” on page 455.

DS Trace includes the following icons:

Icon	Description
	The *Resume Trace* icon restarts a live DS Trace session that you have previously stopped. It is not available for DS Trace log files.
	The *Stop Trace* icon stops a live DS Trace session. It is not available for DS Trace log files.
	The *Connect to Server* icon launches the Login Credentials dialog box so you can authenticate to the server where you want to run DS Trace.
	The *Load Trace Log File* icon lets opens a previously saved DS Trace log file.
	The *Save Trace* icon save the current live DS Trace session to a log file.
	The *Search* icon opens a Find/Replace dialog box where you can search the current DS Trace log file for a specific string. It is not available for live DS Trace.
	The *Configure Trace* icon provides access to live DS Trace settings. It is not available for DS Trace log files.
	The *Clear Trace* icon clears all DS Trace entries from the live DS Trace log.

17.8.1 Viewing DS Trace Live

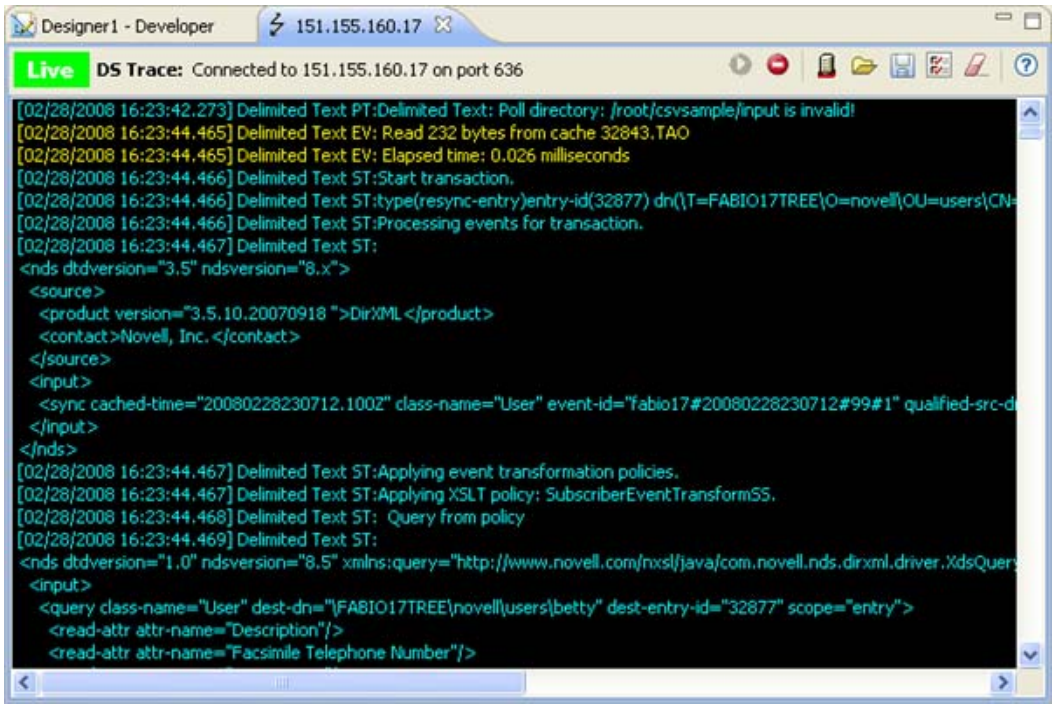
You can view a live DS Trace for any Identity Vault in your Identity Manager environment.

NOTE: Designer provides live DS Trace preferences that let you specify how many entries to keep in the log and whether or not to auto-scroll the log so you can always see the most current entries. You can edit these preferences in *Windows* > *Preferences*, then select *Novell* > *Designer* > *DS Trace* from the left-side navigation.

If the Identity Vault is in your current Designer Project:

1. In the Object view or the Modeler, select an Identity Vault object, then select *Live* > *DS Trace*.

   Alternatively, you can right-click the Identity Vault object, then select *Live* > *DS Trace*.



2 Review the live DS Trace session as needed.

   By default, the DS Trace session is running. You can stop, resume, clear, and save the current trace to a file by using the icons in the DS Trace view toolbar.

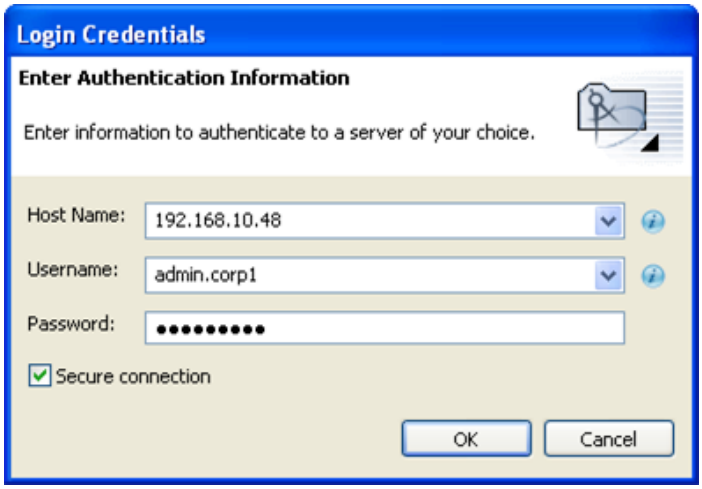
If the Identity Vault is not in your current Designer project:

1 From the main Designer toolbar, select *Tools* > *DS Trace*.

2 In the DS Trace view, click the *Connect to Server* icon.

3 In the Login Credentials dialog box, specify the directory Host name (or IP address), username, and password necessary to connect to the appropriate Identity Vault, then click *OK*.

   Select *Secure Connection* if you need to use SSL to connect to the Identity Vault server.



You can open a DS Trace session to a different Identity Vault server at any time by clicking *Connect to Server* and providing the appropriate authentication credentials.

4 Review the live DS Trace session as needed.

By default, the DS Trace session is running. You can stop, resume, and save the current trace to a file by using the icons in the DS Trace view toolbar.

17.8.2 Creating a DS Trace Log File

DS Trace lets you create log files of DS Trace entries so you can review them off-line. To create a DS Trace log file:

1 From the live DS Trace view, select the *Save Trace* icon .

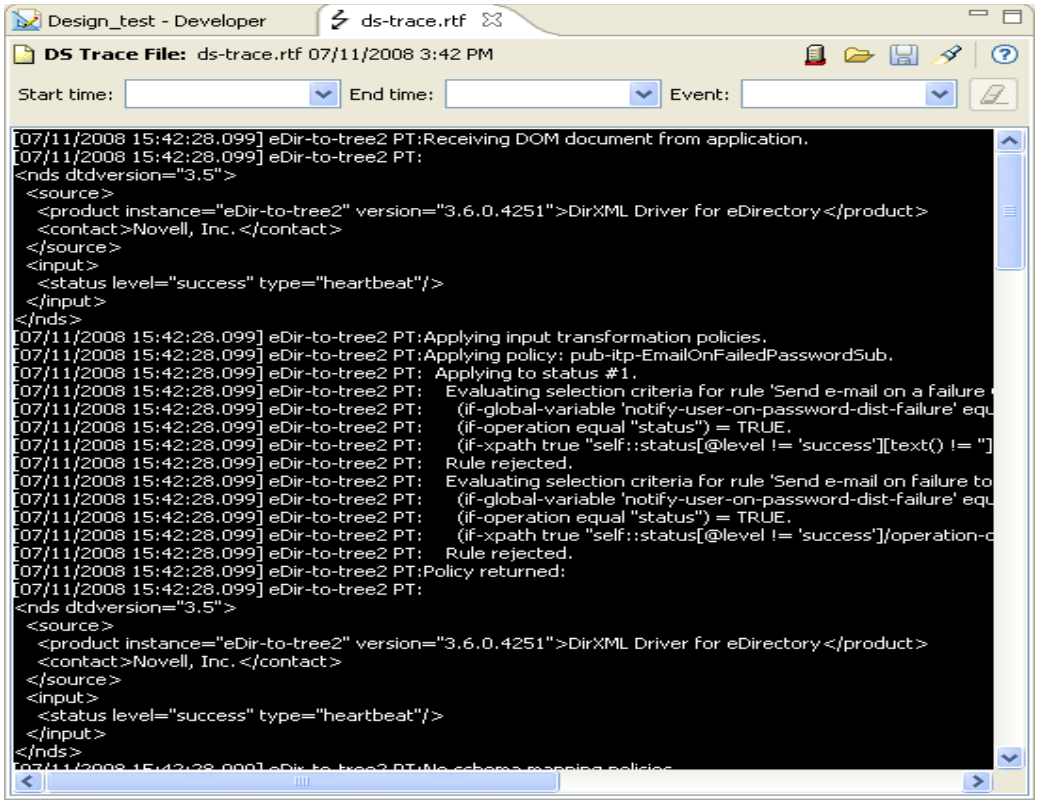
2 Specify a name and location for the log file, then click *Save*.

DS Trace saves the log file as a rich text file (`.rtf`) so it can maintain the color coding used in the live DS Trace view. You can view the log file with any editor that supports the `.rtf` file format.

17.8.3 Viewing a DS Trace Log File

The DS Trace view is an editor that enables you to view DS Trace log files.

Figure 17-4 *The DS Trace View*



To view DS Trace log files:

1 Click *Tools* > *DS Trace*.

2. Select the *Load Trace Log File* icon, then browse to and select the DS Trace log you want to open.
3. Review the DS Trace log file as needed.
   - Use the *Start Time*, *End Time*, and *Event* dropdown lists to filter the trace file. This helps you narrow the displayed trace file data so you can more easily locate specific information.
   - To clear an existing filter, click the *Clear Filter* button .
   - Select the *Search* icon (in the DS Trace icon bar) to open a Find/Replace dialog box that lets you search for a specific string in the DS Trace log file.

NOTE: The Eclipse text editor does not support color, so when you view a DS Trace file in Designer it displays in black and white. However, since Designer saves the DS Trace log file in standard Rich Text Format (`.rtf`), any external text editor that supports color will display the log file in color, as seen in the live DS Trace view.

17.9 Working with Generic Resources

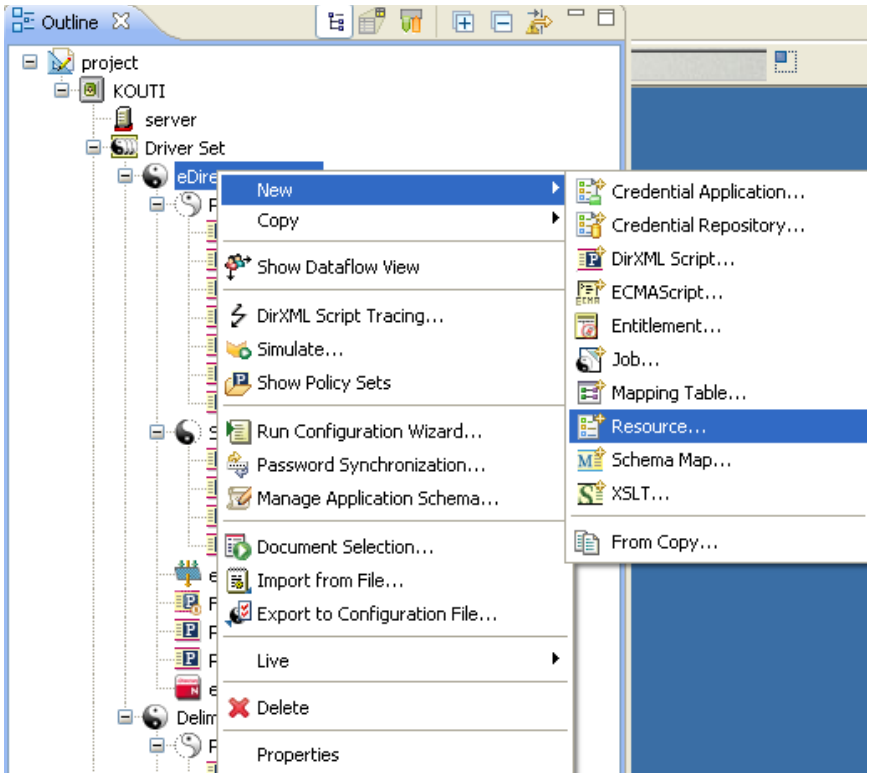
A Resource object is stored in a Driver object or a library. A Resource object stores parameters, which drivers use at any time. When multiple drivers need the same set of constant parameters, the drivers use a Resource object.

A Generic Resource object in Designer enables you to store information in XML or text format. The information can be a piece of documentation, notes, or some piece of data that policies access.

- Section 17.9.1, “Creating a Generic Resource Object,” on page 399
- Section 17.9.2, “Editing a Generic Resource Object,” on page 401

17.9.1 Creating a Generic Resource Object

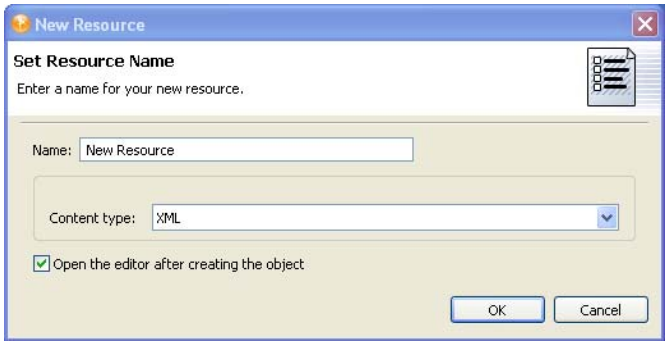
1. In the Outline view, right-click a driver, then select *New* > *Resource*.



You can also do one of the following:

- Right-click a driver, then select *New* > *Resource*.
- With the Dataflow view active, right-click a Subscriber or Publisher channel, then select *New* > *Resource*.
- In the Outline view, right-click a library, then select *New* > *Resource*.

2 Specify the name of the Generic Resource object.



3 Select *XML* or *Text* as the content type.

4 Select *Open the editor after creating the object*, then click *OK*.

5 In the File Conflict dialog box, click *Yes*.

6 Specify the desired XML or text, then press Ctrl+S to save the resource object.

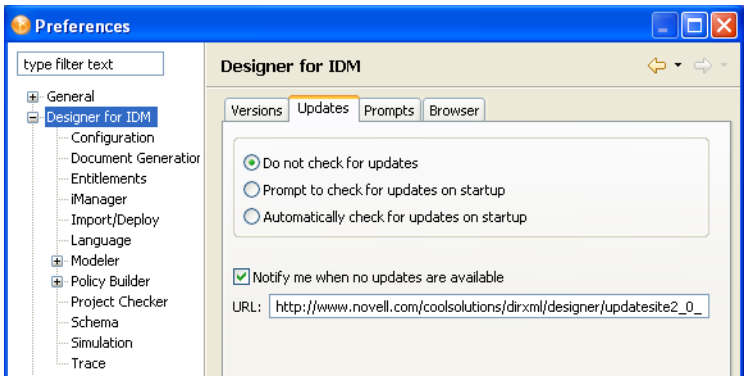
17.9.2 Editing a Generic Resource Object

1. In the Outline view, below the library, right-click the Generic Resource object, then select *Edit*.
2. In the File Conflict dialog box, click *Yes*.
3. Make changes, then save (Ctrl+S).

17.10 Updating Designer

When you start Designer, you are prompted about how you want to receive updates. You can change this setting in Preferences.

Figure 17-5 *The Updates Tab*



If you select to not automatically update Designer, you can get updates by using the Help menu or the Welcome page.

To update from the Help menu:

1. Click *Help* > *Check for Designer Updates*.
   - If your version of Designer is up-to-date, a prompt informs you that no updates are available.
   - If an update is available, a prompt lists components that you can update. Select components, then click *OK*.
   - If your version of Eclipse needs to be updated before you can install Designer 3.6, a dialog box prompts you to click the URL that takes you to the Designer download site.
2. Select updates, then click *OK*.

To update from the Welcome page:

1. Click *Help* > *Welcome*.
2. Click the *What's New* icon.

3. Click *New Updates*.
4. Follow prompts to download and install the latest Designer.

Editing Icons for Drivers and Applications

18

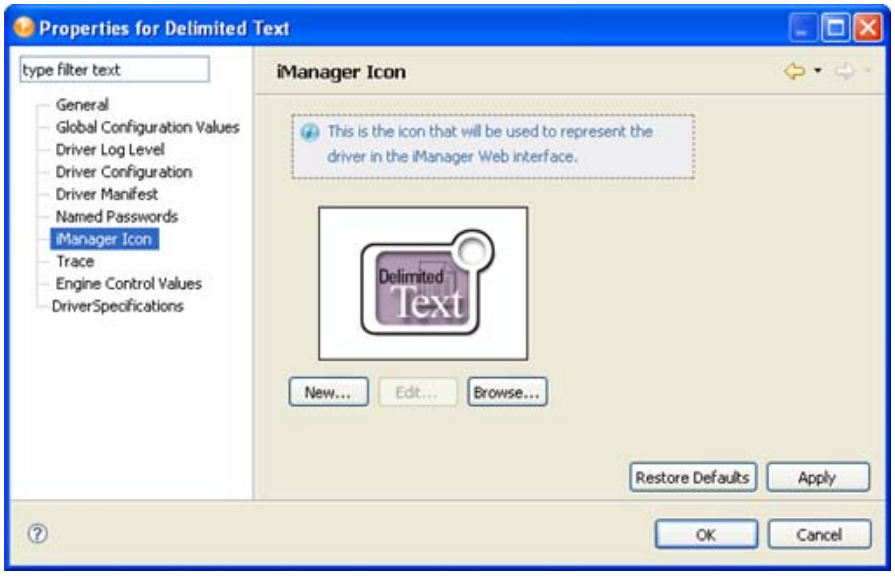
The Icon editor enables you to create customized icons for your drivers and applications. You can enrich predefined templates or images with labels, choose background images and colors, and overlay images. Within minutes, you can create a custom branded icon for your implementation, including your company's logo or name.

- Section 18.1, "Editing Driver Icons," on page 403
- Section 18.2, "Editing Application Icons," on page 406

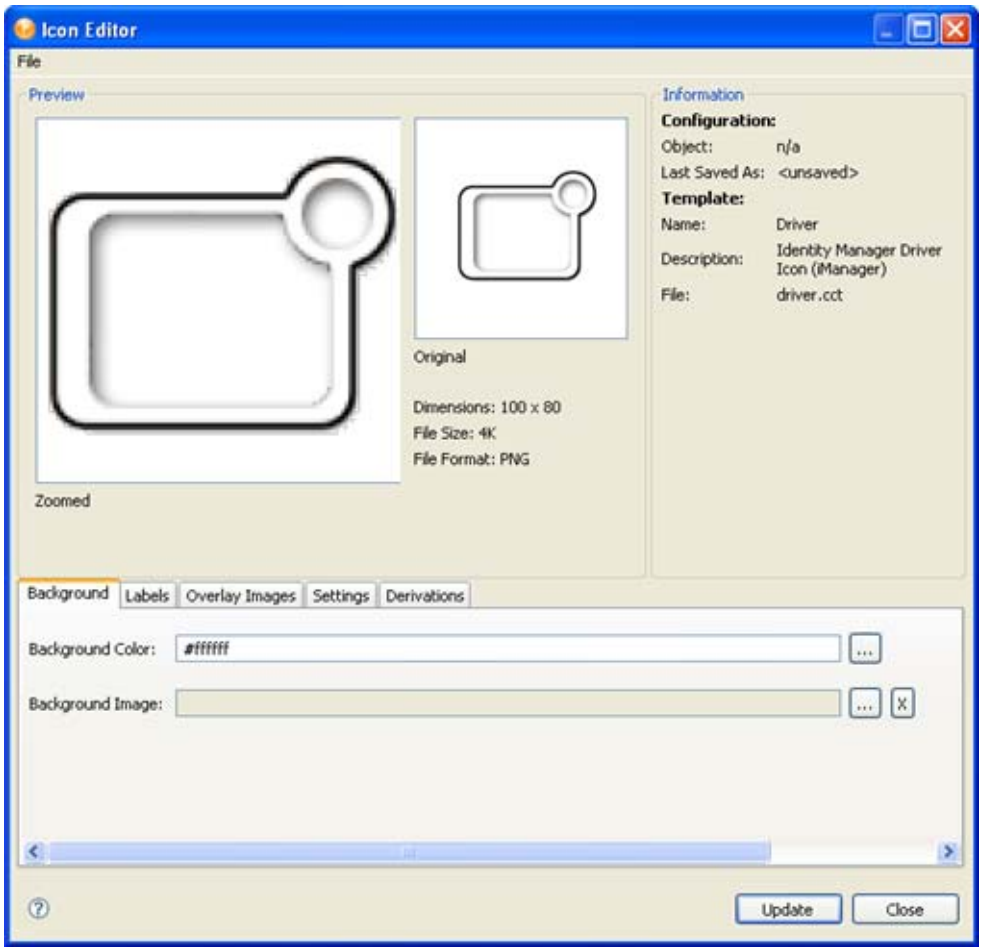
18.1 Editing Driver Icons

1. In the Modeler, right-click a driver, then select *Properties*.
2. In Driver Properties, select the iManager Icon page.

   The object properties dialog box displays the default icon.



3. Click *New* to open the Icon editor.

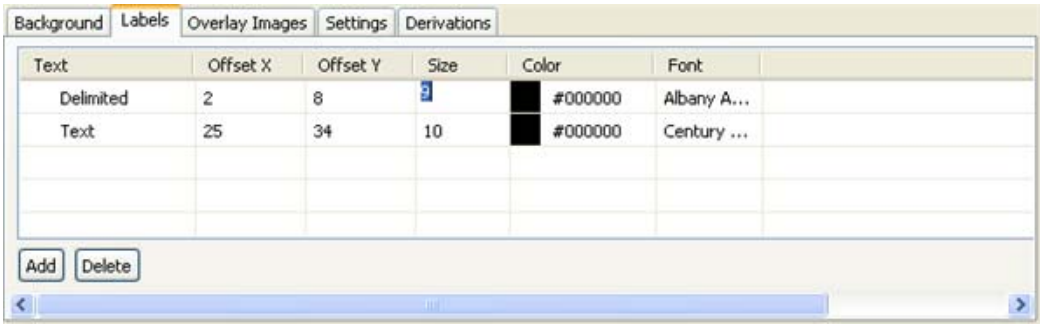


4 To add a background color or image to your icon, select the *Background* tab.

Background Color: Select a background color for the icon.

Background Image: Select a background image. If you select a background image, you can configure how the image displays by using the *Settings* tab, which includes controls for *Brightness*, *Hue*, *Saturation*, and *Gamma*. The Icon editor makes color changes in real time, so you can see the effect of your changes as you make them.

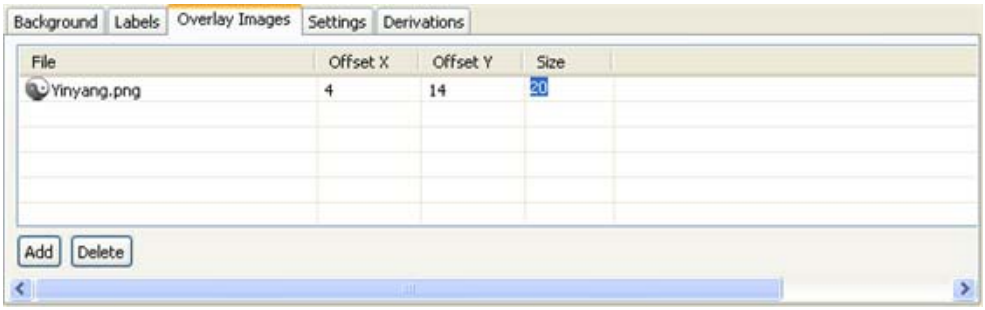
5 To specify icon text, select the *Labels* tab, then click *Add*.



To add a label to the icon, type a new label in the *Text* column. Text does not automatically wrap around. To create multi-line text, create a separate text entry for each line.

You can also control label placement through the Offset X and Offset Y options, and edit font size, font type, and color for each text entry.

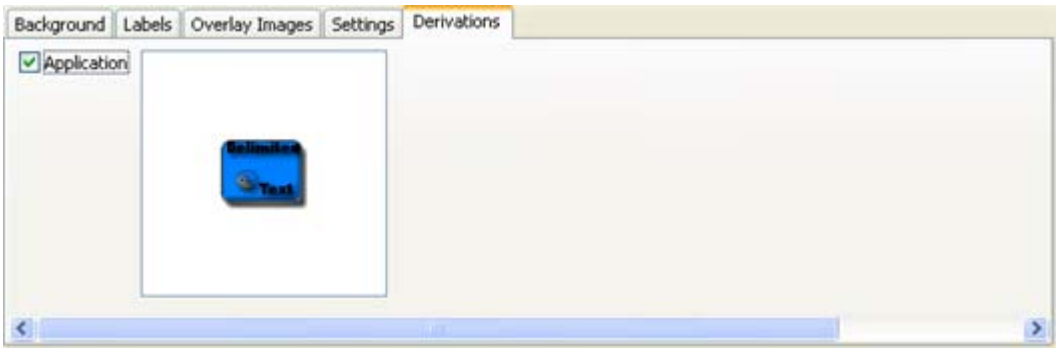
6 To add an overlay image to your icon, select the *Overlay Images* tab, then click *Add*.



To select an image, select a cell in the *File* column. A small icon appears to the right of the file name. Click the icon and browse to the image you want to use as an overlay image.

You can also control image placement through the Offset X and Offset Y options, and control the size of the image in pixels.

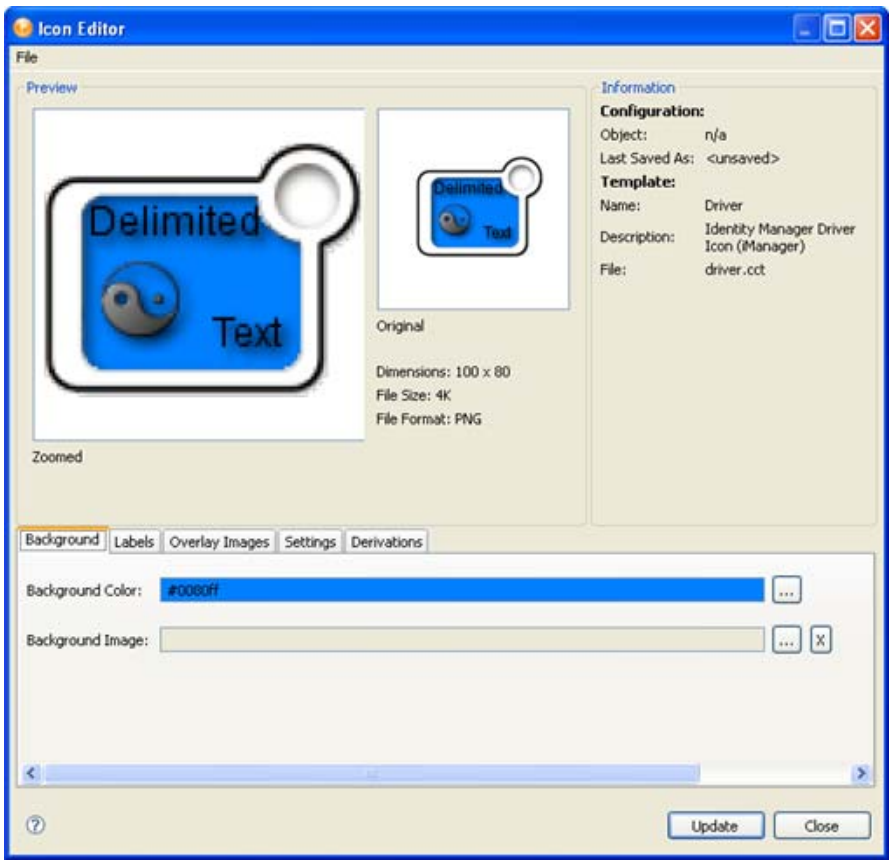
7 To create a similar icon for iManager, select the *Derivations* tab, then select *Application*.



This transfers the icon to the driver's or application's iManager properties. (See the iManager icon on the driver properties page.)

Some icons don't convert cleanly between the Driver and Application icon formats, so you might need to clean it up after you create it.

8 When you are finished editing the icon, click *Update*.

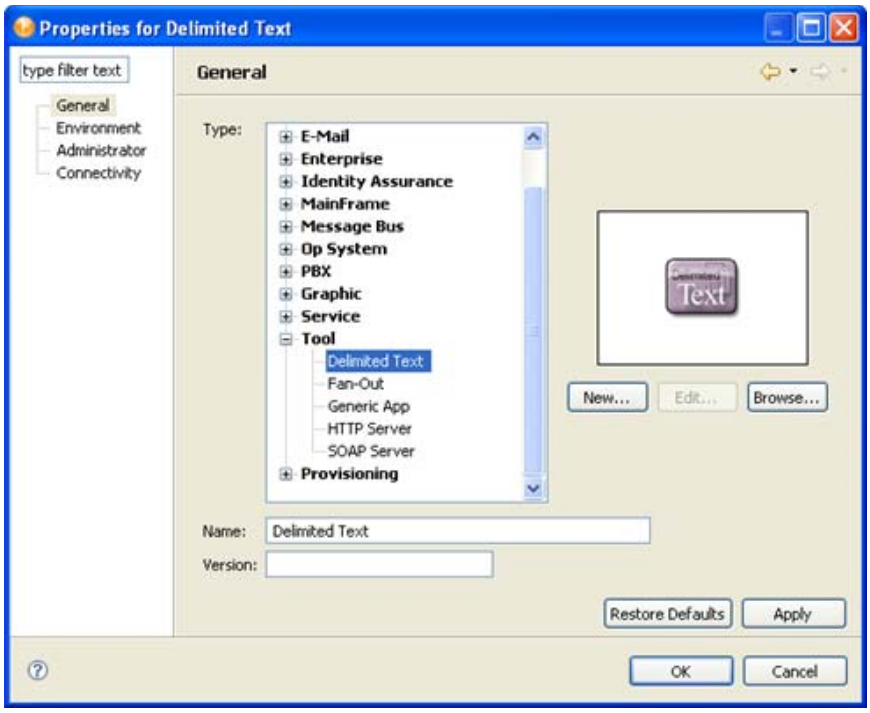


18.2 Editing Application Icons

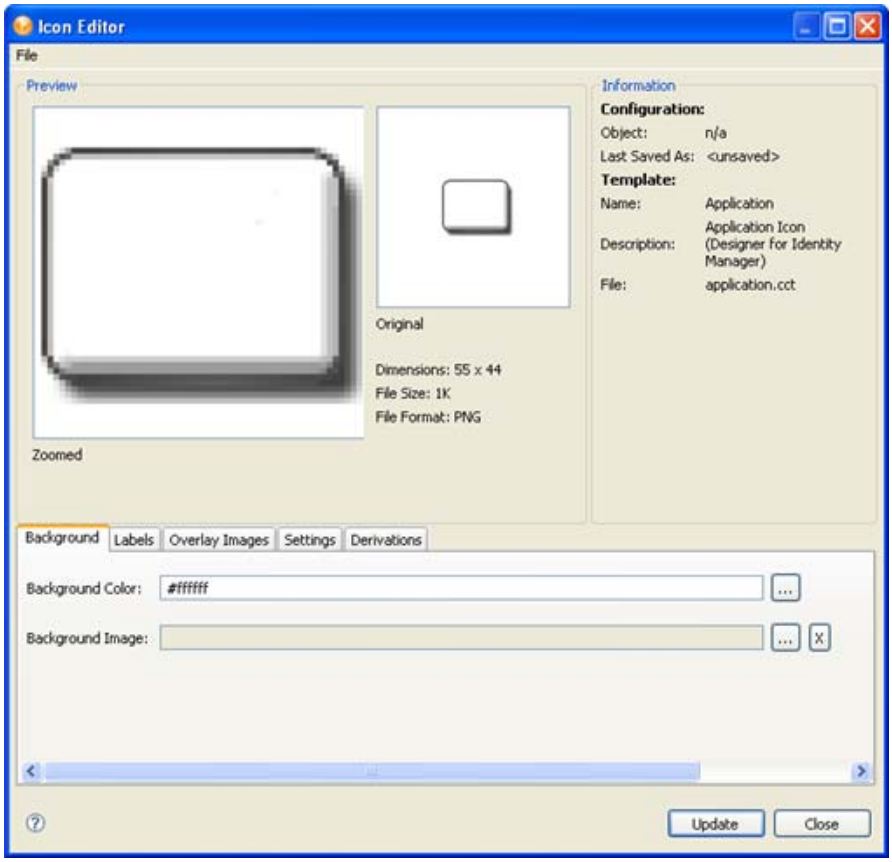
1 In the Modeler, right-click an application, then select *Properties*.

2 In Application Properties, select the *General* page.

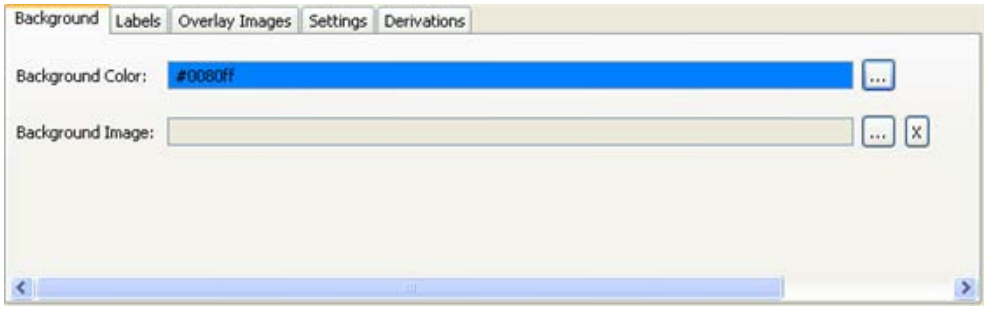
The object properties dialog box displays the default icon.



3 Click *New* to open the Icon editor.



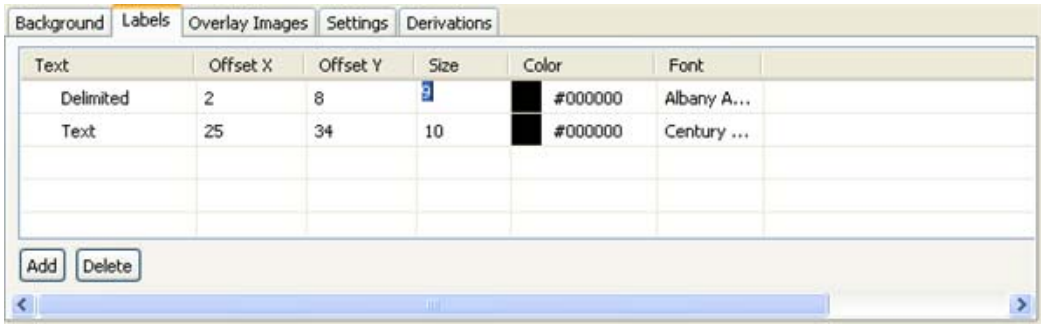
4 To add a background color or image to your icon, select the *Background* tab.



Background Color: Select a background color for the icon.

Background Image: Select a background image. If you select a background image, you can configure how the image displays by using the *Settings* tab, which includes controls for *Brightness*, *Hue*, *Saturation*, and *Gamma*. The Icon editor makes color changes in real time, so you can see the effect of your changes as you make them.

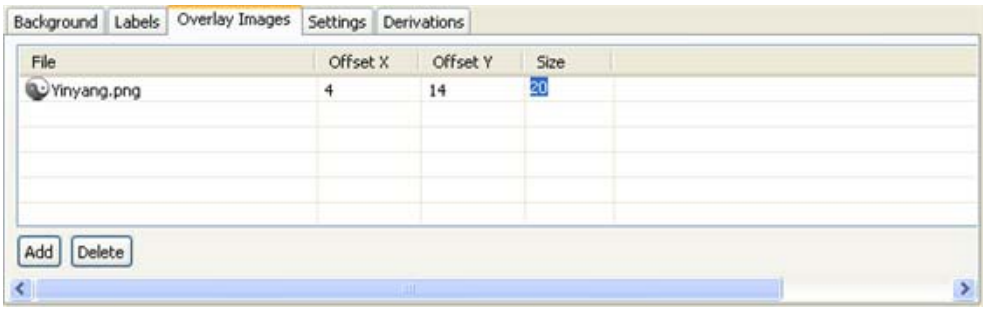
5 To specify icon text, select the *Labels* tab, then click *Add*.



To add a label to the icon, type a new label in the *Text* column. Text does not automatically wrap around. To create multi-line text, create a separate text entry for each line.

You can also control label placement through the Offset X and Offset Y options, and edit font size, font type, and color for each text entry.

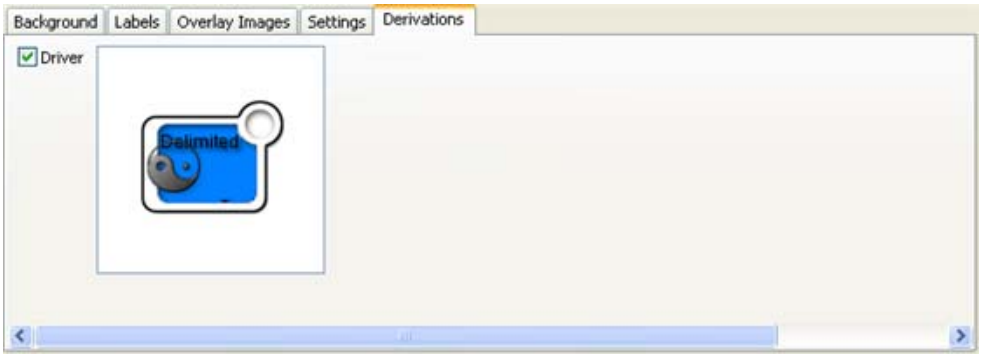
6 To add an overlay image to your icon, select the *Overlay Images* tab, then click *Add*.



To select an image, select a cell in the *File* column. A small icon appears to the right of the file name. Click the icon and browse to the image you want to use as an overlay image.

You can also control image placement through the Offset X and Offset Y options, and control the size of the image in pixels.

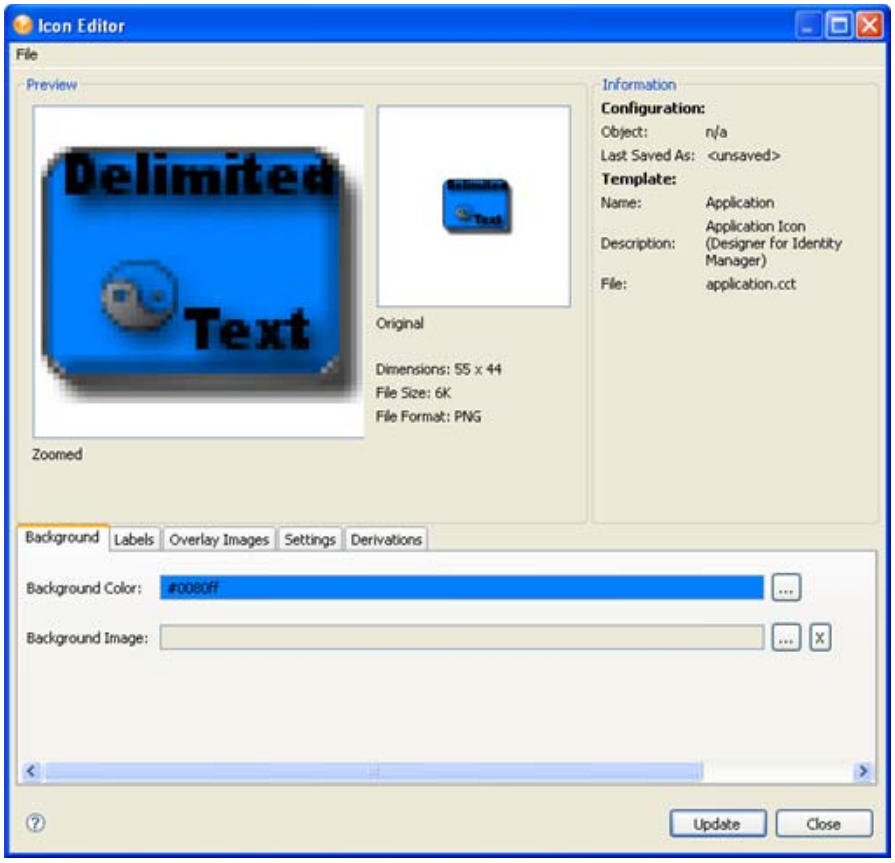
7 To create a similar icon for iManager, select the *Derivations* tab, then select *Driver*.



This transfers the icon to the driver's or application's iManager properties. (See the iManager icon on the driver properties page.)

Some icons don't convert cleanly between the Application and Driver icon formats, so you might need to clean it up after you create it.

8 When you are finished editing the icon, click *Update*.



Version Control

19

Designer's version control enables you to do the following:

- Provide simple document management by tracking revisions of your project, along with all the objects and files in that project
- Share those revisions with other members of your team
- Manage the history of your objects
- Make sure that every member of your team is using the same version of your project and Designer

Designer supports the Subversion version control system. All Subversion versions through version 1.5.*x* are supported. Subversion is a stable open source product that is available for no cost and is released under the Apache license. For information on Subversion, see Tigris.org (http://subversion.tigris.org/). You can also find some pertinent information about Subversion working with version control in Designer at Appendix E, "Version Control with Subversion and Identity Manager Designer," on page 565, as well as Section 19.6, "Version Control Best Practices," on page 437.

Version control allows teams to work together across continents or just across the hallway, in groups or as a single user. The Version Control view gives you information about changes that your teammates are making in real time. The version control framework allows you to update, merge, and resolve conflicts with your teammates. If you are a single user, version control allows you to make backups, restore older versions, and have the freedom to explore project changes without risking data.

With version control, you can manage the history of your project, and you can go back to a previous revision and create tagged revisions for better release management. Anyone with permission can access these revisions. The Compare Revisions feature allows you to easily scan the history of your project, find relevant changes, and resolve project issues.

Version control functionality is available for all Identity Manager objects as well as for the contents of the Documents and Toolbox folders. Version control is not available for Novell Analyzer when Analyzer is integrated with Designer.

- Section 19.1, "Installing a Subversion Server on Windows," on page 411
- Section 19.2, "Checking In a Project To a Version Control Server," on page 413
- Section 19.3, "Importing a Project from a Version Control Server," on page 417
- Section 19.4, "Accessing the Version Control View," on page 419
- Section 19.5, "Comparing Revisions and Resolving Conflicts," on page 428
- Section 19.6, "Version Control Best Practices," on page 437

19.1 Installing a Subversion Server on Windows

You can either install a Subversion server or use an existing Subversion server. Designer's version control works with Subversion server platforms up through version 1.5.*x*.

This section provides a quick start for a basic Subversion server on Windows to use with Designer for Identity Manager. For more in-depth information on installing Subversion on Windows and on other platforms, see Subversion's installation documentation at Installing Subversion (http://svn.collab.net/repos/svn/trunk/INSTALL).

- Section 19.1.1, "Downloading and Installing the Server," on page 412
- Section 19.1.2, "Configuring the Server," on page 412

19.1.1 Downloading and Installing the Server

1. Go to the Tigris.org (http://subversion.tigris.org/servlets/ProjectDocumentList?folderID=91) Web site and download the most recent 1.5.*x* version of Subversion..
2. Run the installer and accept the license agreement.
3. Specify the location to install Subversion.
4. Specify a location in the *Start* menu.
5. Follow the on-screen instructions to complete the installation.

19.1.2 Configuring the Server

1. Use Windows Explorer to create a directory to contain the Subversion server repository. You will seldom open this directory manually.
2. Run the `svnadmin create` command to create the repository at that directory location:

   ```
   svnadmin create [location_of_Subversion_repository]
   ```
3. Go to the `[location_of_Subversion_repository]\conf` directory, which was created when you installed the Subversion server.
4. Edit the `svnserve.conf` file by uncommenting the following lines in the General section (there should be no spaces at the beginning of the lines):

Line to Uncomment	Result
`anon-access = read`	Anonymous users can read your repository.
`auth-access = write`	Authenticated users can edit your repository.
`password-db = passwd`	Usernames and passwords are stored in a file named `passwd` in your `conf` directory.

5. Edit the `passwd` file in the same directory.
6. Remove the sample users from the `Users` section and add your own users.
7. Open a command prompt and start your server by using the following command:

```
svnserve --daemon --root [location_of_Subversion_repository]
```

8. Open a second command prompt.
9. Create a `trunk` folder in your repository with the following command:

   ```
   svn mkdir -m "Creating a trunk directory." svn://localhost/trunk
   ```
10. Authenticate to Subversion.

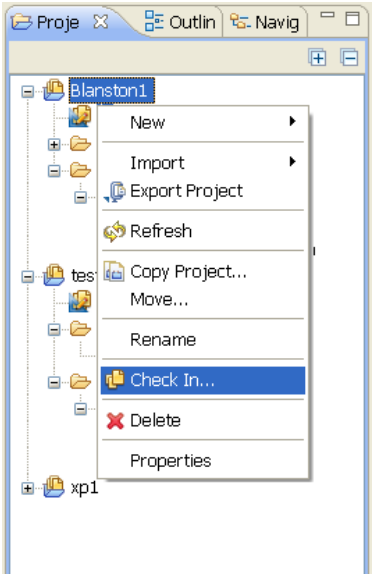
If your username is the same as your Windows username, enter your password. Otherwise, press Enter at the password prompt and enter a username when prompted.

You can also access this server from other computers by substituting `localhost` for the network name of the server machine in the URL.

You are now ready to import or add projects to version control by using Designer for Identity Manager. You might want to create a more complete directory structure before adding Identity Manager projects. For more information on how to best use Subversion with Designer's version control, see Appendix E, "Version Control with Subversion and Identity Manager Designer," on page 565.

19.2 Checking In a Project To a Version Control Server

1 In the Project view, right-click a project name, then click *Check In*.

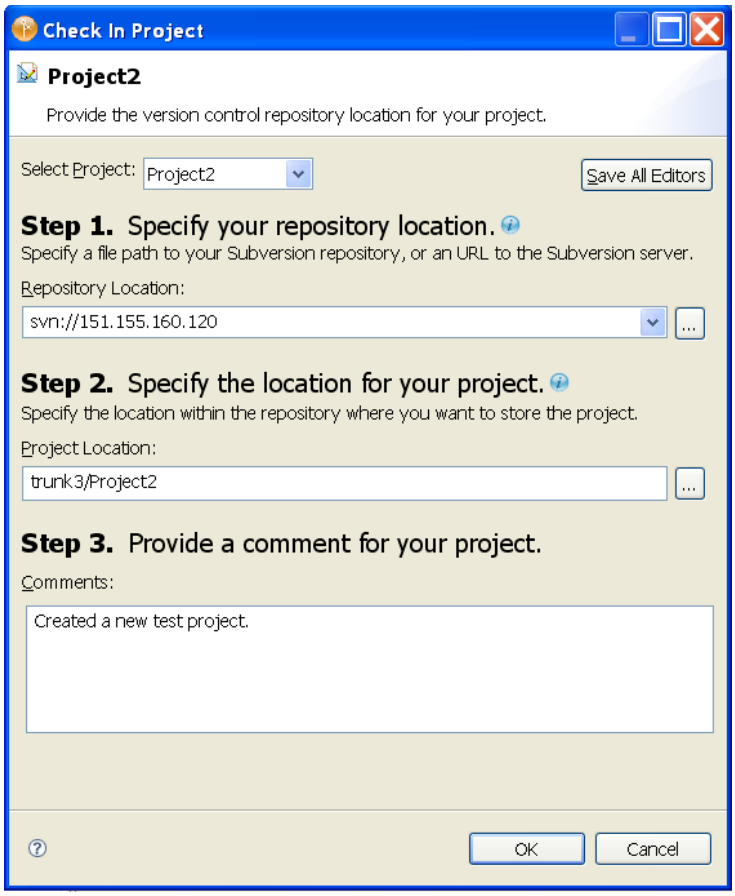


You can also select the *Check Project Into Version Control Server* icon on the main toolbar.

2 If the project you are checking in already exists on the version control server, skip to Step 8.

or

If the project you are checking in does not exist on the version control server, you see the Check In Project page displayed. Continue with Step 3.



3 If you have multiple projects in the Project view and you clicked the *Check Project Into Version Control Server* icon, select the project you want to check into the version control server from the *Select Project* drop-down list. If you select *Check In* from the Project view, you won't see the *Select Project* list.

4 Under *Step 1. Specify your repository location*, provide a URL pointing to where you want the project to reside on the version control server. The Check In Project page gives three examples:

- `c:/subversionrepo`
- `http://subversionserver.mycompany.com`
- `svn://localhost`
- `https://subversionserver.mycompany.com/svn/myrepository`

The list of supported protocols includes:

- svn
- http
- https
- file
- svn+ssh

You can click the *Browse* button to browse for folders that are saved either locally or on a network drive. You can also create a new folder from the Browse For Folder page.

5 Under *Step 2. Specify the location of your project*, type the folder name that will contain this project on the version control server.

You can also click the *Browse* button to bring up the Version Control Server Browser page. This browser helps you determine the correct URL where projects are stored and only shows base folders and corresponding projects.

The base folder cannot be a directory of a Designer project. However, the base folder can contain multiple projects as subdirectories. You create base folders through an external svn client.

6 (Optional) Under *Step 3. Provide a comment for your project*, type a comment concerning the project, then click *OK*.

Whenever you perform an operation that affects the contents of the server, you are prompted for a comment. Comments are useful when keeping track of the changes you make from one session to another.

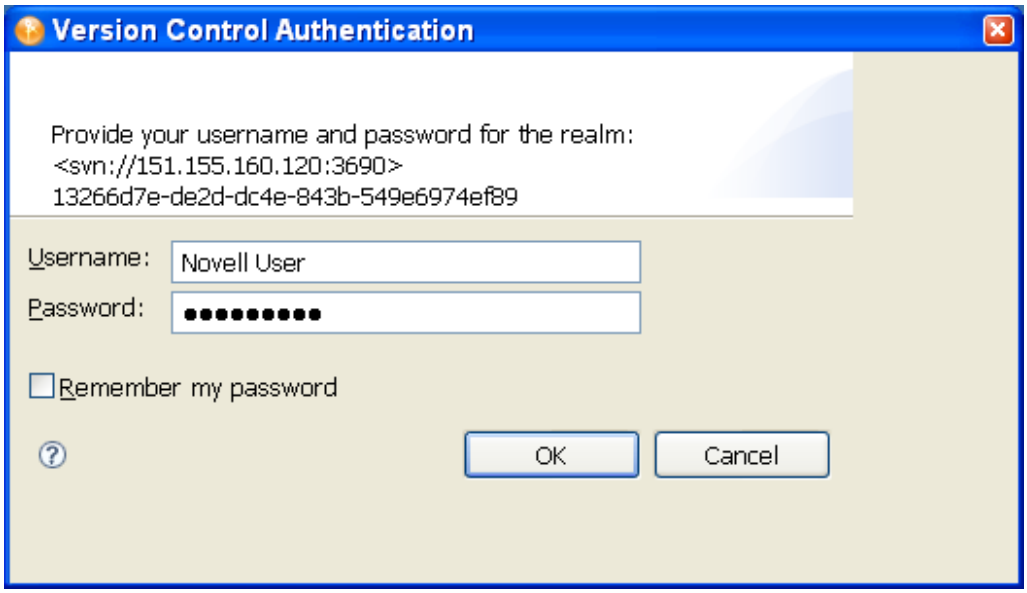
7 (Optional) If you have made changes to more than one project in the Modeler view, you need to save those changes before checking a project into version control.

7a Select *Save All Editors* to bring up the Save Resources page, which allows you to save all open projects.

7b On the Save Resources page, click *OK*. You are returned to the Check In Project page.

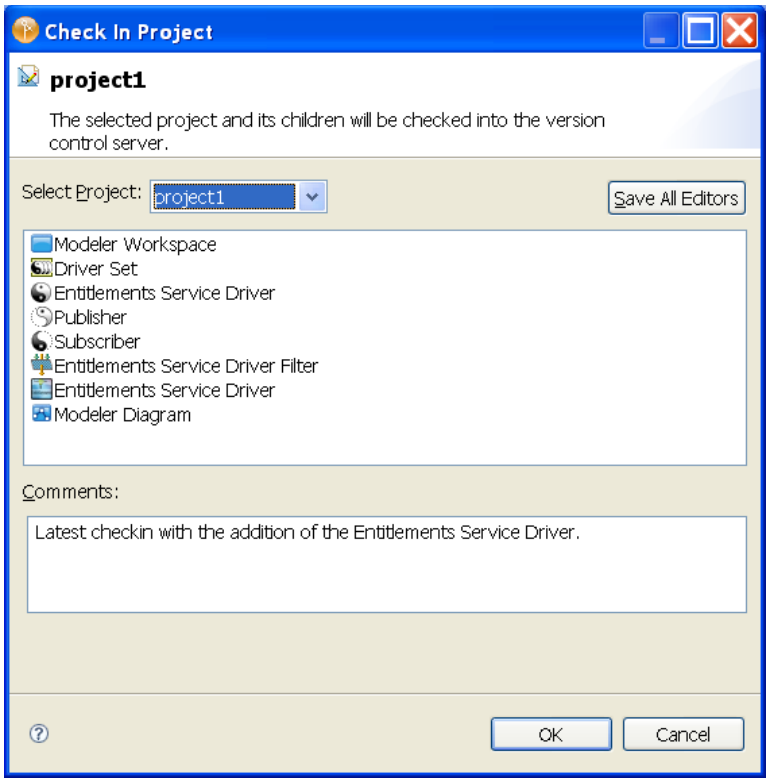
8 Provide authentication to the Subversion server if required.

Depending on the type of security you have set up, you might need to supply SSH authentication, SSL client certificate authentication, or basic HTTP authentication.



9 If you are updating an existing project on the version control server, add new information to the Comment section of the Check In Project page.

If you are updating a project, you see the selected project, the object's children that have changes to be checked in, and objects that depend on the project and need to be checked in. If you choose to check in a single object, you only see that object in the Check-in page.



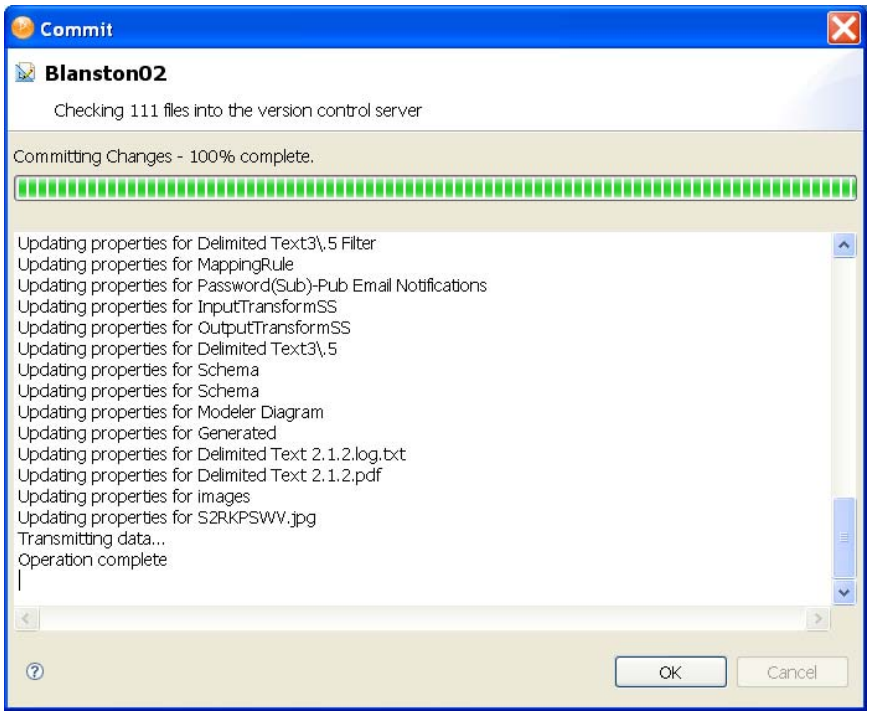
10 If you have more than one project open in the Modeler view, click *Select Project* to choose which project you want to save to version control.

11 (Optional) If you have made changes to more than one project in the Modeler view, you need to save those changes before checking a project into version control.

11a Select *Save All Editors* to bring up the Save Resources page, which allows you to save all open projects.

11b On the Save Resources page, click *OK*. You are returned to the Check In Project page with an updated list of what is being checked in.

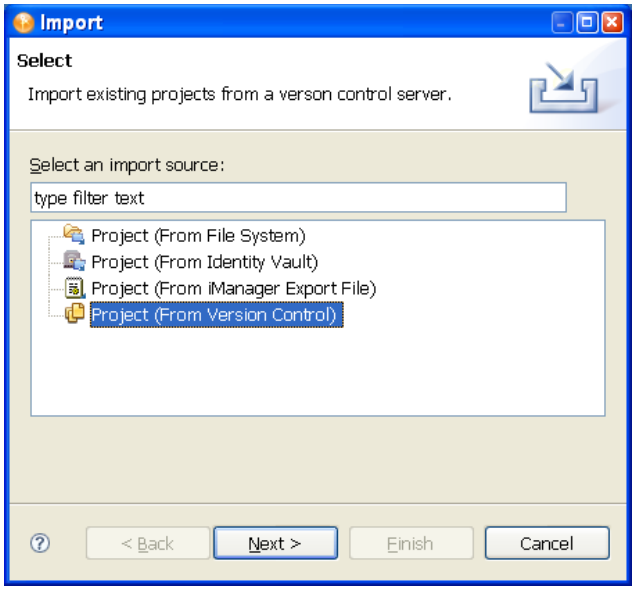
12 Click *OK* to commit the files to the version control server. When the files are committed, click *OK* to close the Commit page.



19.3 Importing a Project from a Version Control Server

Designer's Import dialog box lists projects and enables you to select projects that you want to import. There are a number of ways to access the Import dialog box in order to import projects from a version control server, and this example covers one of those methods.

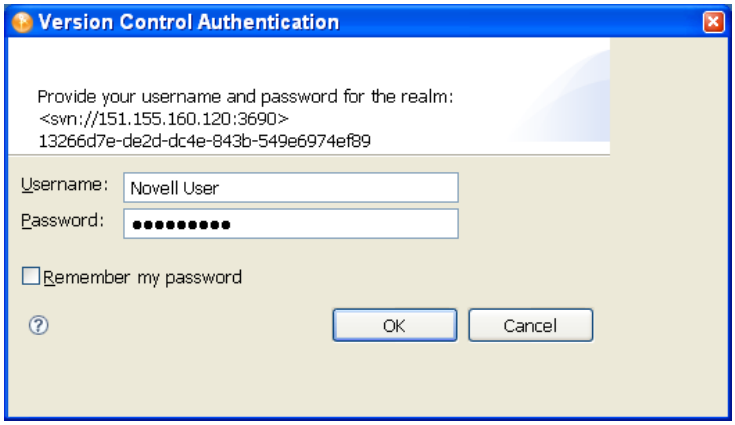
Figure 19-1 *The Import Wizard*



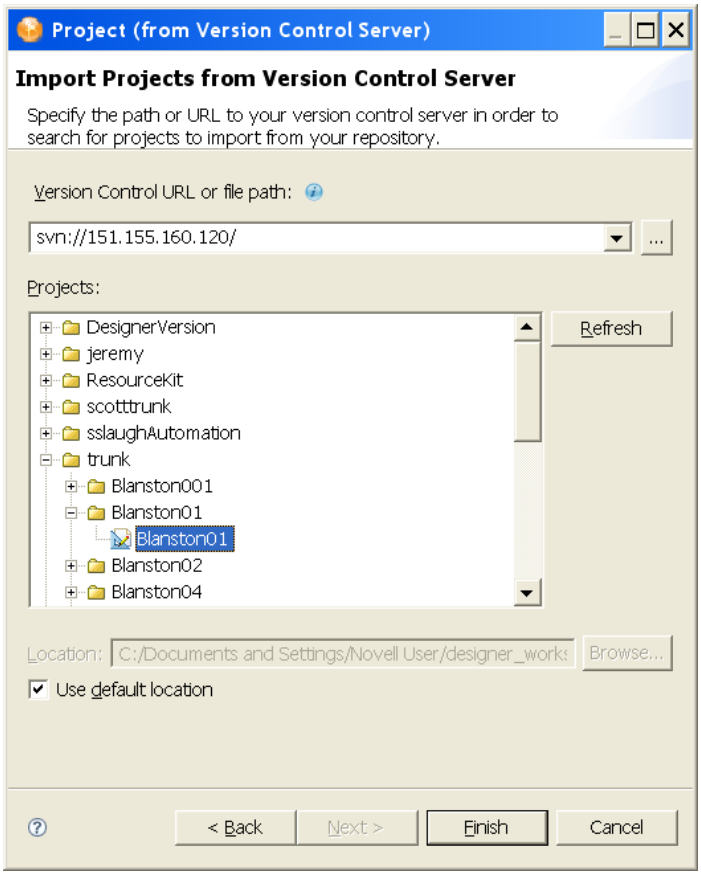
1. Select *File* > *Import*.
   or
   If no projects are available, select *Import from version control* from the Project view.
2. Click *Project (From Version Control)* > *Next*.
3. Type a URL in the *Version Control Server URL or file path* field, then press Enter. For example:

   `https://sun.provo.novell.com/svn`

   `svn://123.123.131.120/trunk`
4. (Optional) You can also type in a file path to the version control repository, or select the Browse button to browse to the directory where the repository resides.
5. Provide authentication to the Subversion server if required. Depending on the type of security you have set up, you might need to supply SSH authentication, SSL client certificate authentication, or basic HTTP authentication.



6 The projects appear under the *Projects*: heading in a tree structure. Select a project file under the directory. Use the *Refresh* button to see current changes to the repository.



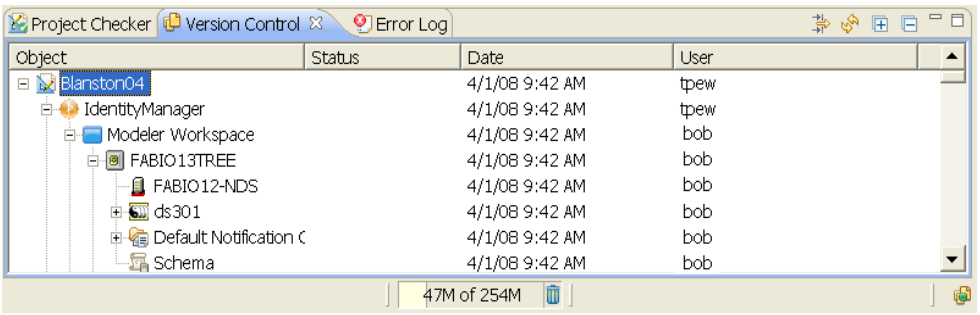
7 Click *Finish*. On the Version Control page that shows you the version control server status, click *OK*.

The projects are imported into Designer and are added to the Project view and the Version Control view.

19.4 Accessing the Version Control View

You access version control functionality by using the Version Control view.

Figure 19-2 *The Version Control View*



The Version Control view does the following:

- Gives you a dashboard status of your interaction with version control
- Lists the files that you are working on
- Displays the changes that your teammates have made in real time

The Version Control view is the main interface with version control. You find most of the version control operations and information in this view. This view is empty until you import from or check in a project to the version control server.

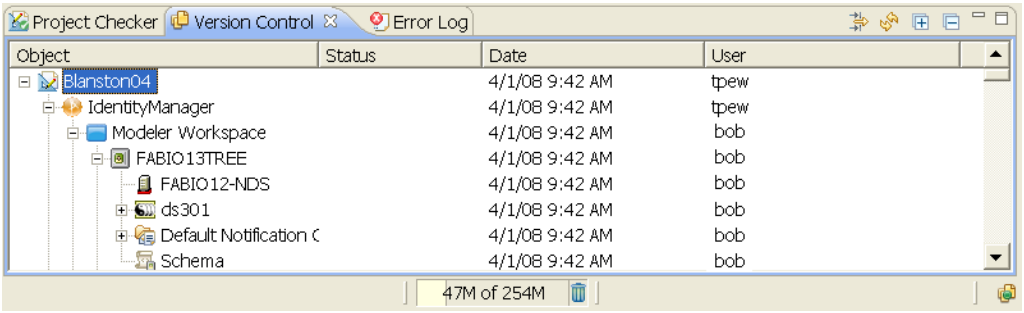
The Version Control view automatically displays when you import an existing project from a version control server or check in a project to a version control server. To open the view manually, select *Window* > *Show Views* > *Version Control*.

- Section 19.4.1, “Version Control Icons,” on page 420
- Section 19.4.2, “Version Control View Headings,” on page 421
- Section 19.4.3, “Version Control Options,” on page 422

19.4.1 Version Control Icons

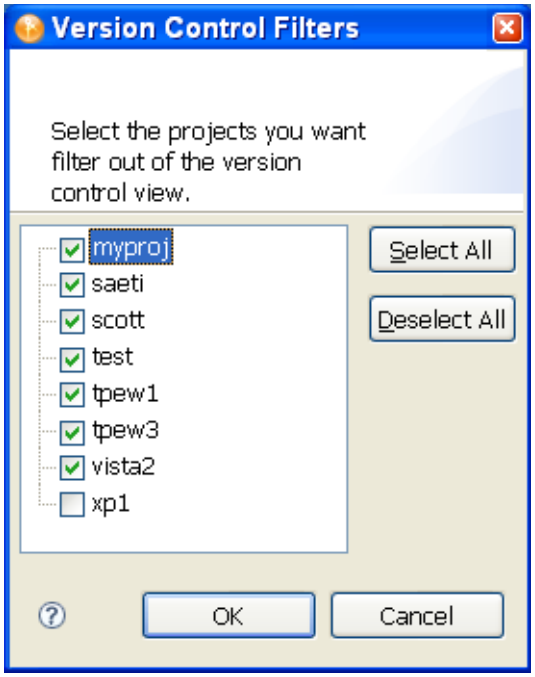
The Version Control view contains seven icons that allow you to interact with version control. Six icons are to the right of the Version Control tab. They are the *Filter* icon, the *Refresh* icon, *Expand All* and *Collapse All*, and the *Minimize* and *Maximize* icons. The seventh icon is the *Version Control Project Status* icon, which is located in the bottom right corner of Designer.

Figure 19-3 *Details in the Version Control View*



Filter Icon: Use the *Filter* icon to limit the number of projects that are displayed in the Version Control view. Click the *Filter* icon, then select the projects you want to filter out of the Version Control view.

Figure 19-4 *Version Control Filter Page*



Refresh Icon: Click the *Refresh* icon to refresh the Version Control view. Designer communicates with the Subversion server and refreshes the Version Control view with any updates performed by other users who are modifying the same projects.

Expand All/Collapes All Icons: Click the *Expand All* icon to expand all items in the Version Control view. Click the *Collapse All* icon to collapse all items in the Version Control view.

Minimize/Maximize Icons: Click the *Minimize* icon to minimize the Version Control view. Click the *Maximize* icon to maximize the Version Control view.

Version Control Project Status Icon: Mouse over the *Version Control Project Status* icon to see the status of the objects in the Version Control view. The *Version Control Project Status* icon gives you a quick status for version control and works like a traffic light. You can move this icon to a different location in Designer to suit your preferences.

Table 19-1 *The Version Control Project Status Icon Colors and Description*

Icon Status	Status Description
	Green. Everything is up-to-date.
	Yellow. Updates are available from the version control server.
	Red. There are conflicts between the local version and the version control server.
	Grey. Designer is unable to contact the version control server.

19.4.2 Version Control View Headings

The Version Control view has four headings: Object, Status, Date, and User.

Object: This column displays the objects that are connected to the project that is stored on the version control server. Right-click an object in the Version Control view to display the available options. These options are covered in Section 19.4.3, “Version Control Options,” on page 422.

Status: This column displays the current state for objects in a project, as indicated by the following icons:

Table 19-2 *Status Icons*

Status Icon	Description
(none)	This object is up-to-date, with no new revisions available.
–	Unversioned. This object has not been added to the version control server.
×	Deleted. This object has been deleted from the version control server.
	Updates with Merge. This object has updates that might conflict with the changes you have made (see Section 19.5, “Comparing Revisions and Resolving Conflicts,” on page 428).
	The project object has been updated from an older version selected from the Revision History page. The object changes back to normal when you update (see “History” on page 425).
	This object has new child objects available.
	This object has new updates available.
	This object has been modified locally.

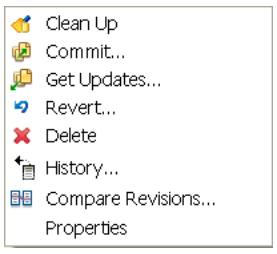
Date: This column shows the date when the last changes to the objects in the Version Control view occurred. The date and time change when you modify an object and commit those changes to the version control server.

User: Displays the name of the last person who has updated the object.

19.4.3 Version Control Options

Right-click an object in the Version Control view to display the available options.

Figure 19-5 *Available Version Control Options*



The options affect the object selected, as well as any child objects that correspond to the selected object. For example, performing a Revert on the project object affects the entire project, but performing a Revert on the Subscriber channel of a Lotus Notes driver only affects the Subscriber channel and any objects (such as policies) that depend on the Subscriber channel.

- “Clean Up” on page 423
- “Commit” on page 423
- “Get Updates” on page 424
- “Revert” on page 424
- “Delete” on page 425
- “History” on page 425
- “Comparing Versions” on page 428
- “Properties” on page 428

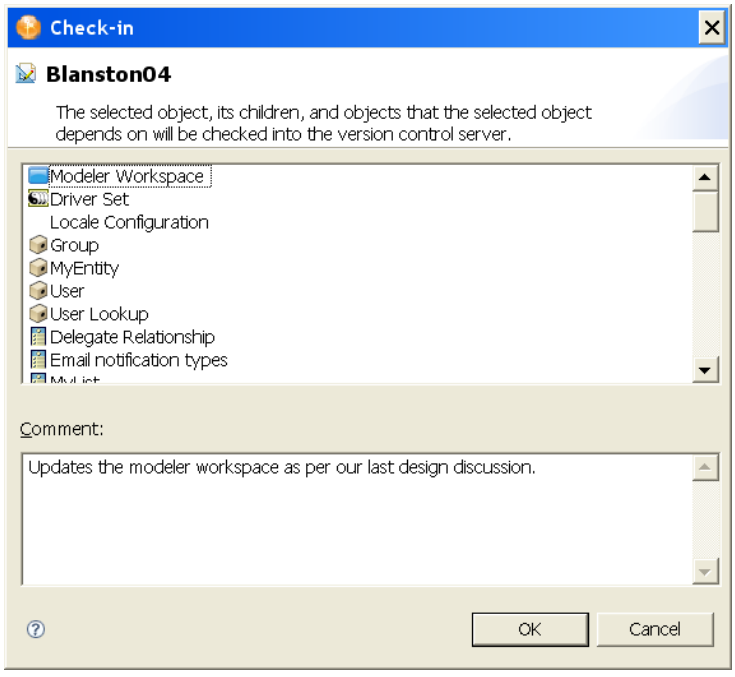
Clean Up

Use the *Clean Up* option only when you are prompted to. Sometimes a project is in a “locked” state; at this point, version control requires you to run Clean Up before it lets you do anything else with the project, and you receive a message telling you to run the *Clean Up* option.

Commit

Use the *Commit* option to have your local changes checked into the version control server for the object you have selected.

Figure 19-6 *Checking In an Object to the Version Control Server*



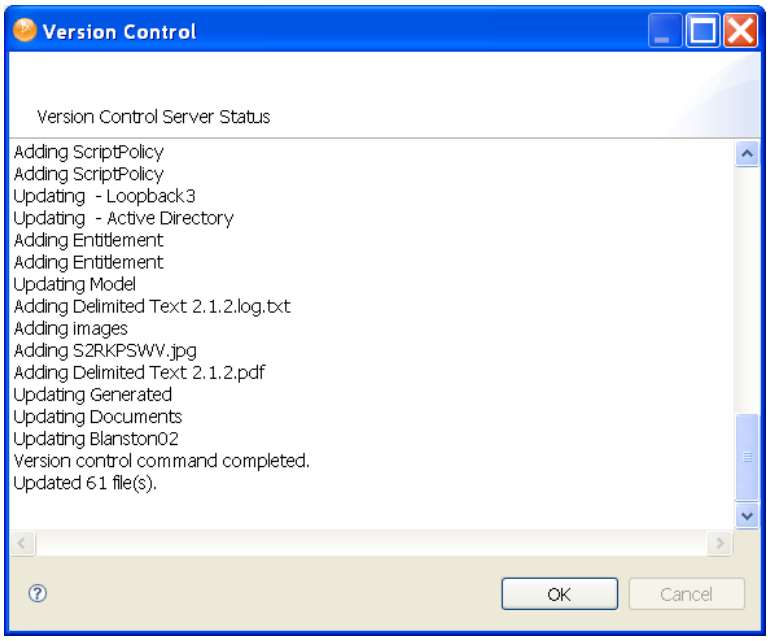
Once you click *OK*, the check-in is committed to the version control server. Click *OK* to close the Commit screen.

There are also *Check-in* capabilities in the Project and the Outline view (right-click a project and select *Check In*), and an *Check In* icon in the main toolbar .

Get Updates

Use the *Get Updates* option to get the latest version of the selected object from the version control server.

Figure 19-7 *Receiving Updates from the Version Control Server*



If you have more than one project open that is checked in to the version control server, select which project you want to update from the Update page, then click *OK* to begin the update. If there are conflicts between your local version and the version control server, you see the Conflict Resolution page, which includes a method to resolve those conflicts. For more details, see Section 19.5, “Comparing Revisions and Resolving Conflicts,” on page 428.

There are also *Update* capabilities in the Project and the Outline view (right-click a project and select *Update*), and an *Update* icon in the main toolbar .

Revert

Use the *Revert* option to return the selected object to the version you last checked out from the version control server. This allows you to cancel your recent changes; you see a message screen displayed, confirming your choice to revert. You can also use this option to restore files that you have deleted since the last time you checked in. Revert will not affect files in your project that have not been checked in to the version control server.

WARNING: By using this option, you lose any changes you have made since the last time you checked it in.

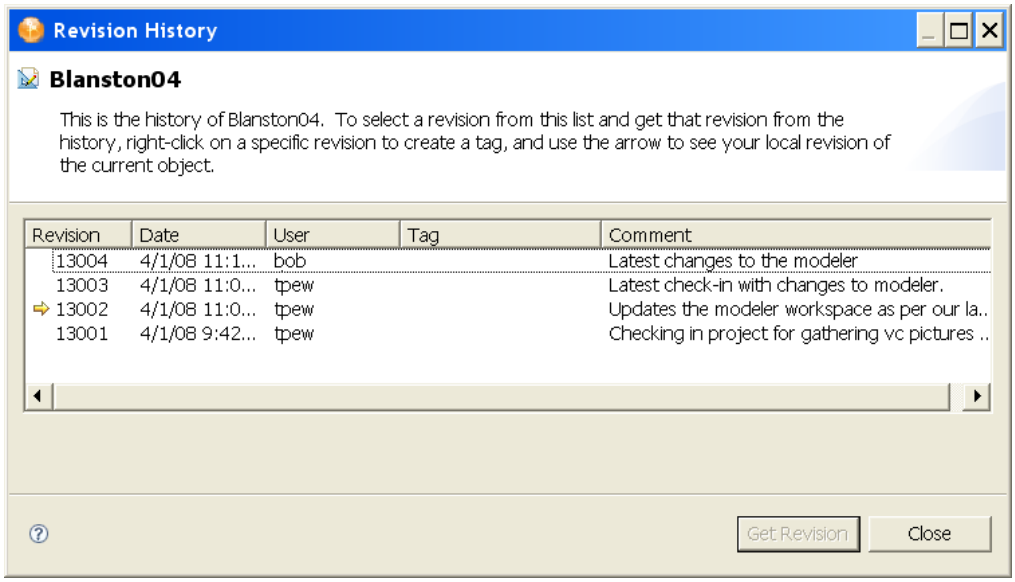
Delete

Use the *Delete* option to delete a project from the version control server. This option is only available for project objects. Although you can delete objects within a project from other views in Designer, you can remove the entire project only through the Version Control view. Selecting the *Delete* option immediately deletes the selected project, and you are prompted for a comment for your actions.

History

Use the *History* option to view the revision history of an object and all the changes that have been made to that object. You can also use this option to select an earlier version of a project.

Figure 19-8 *Revision History in the Version Control View*



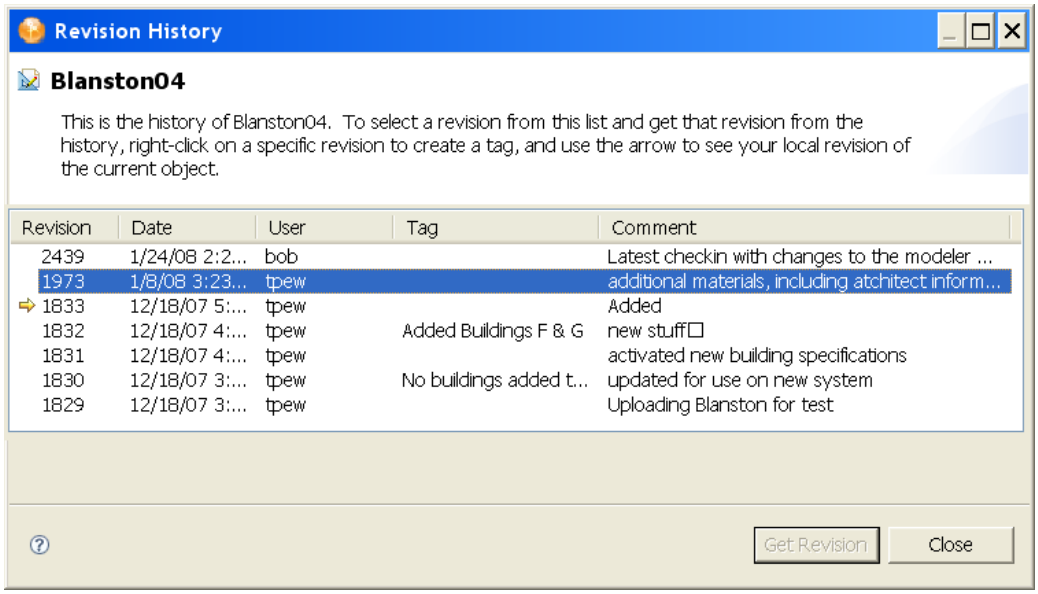
You can use Revision History page to see who made a change, when the change was made, the tag name (if it is filled out), and the comment provided for the change. The yellow arrow indicates your currently loaded version.

Version numbering of projects and how numbering works with the objects in a project is a very complex issue. For more information on how revision numbering works in Subversion, see Section E.1.1, “How Revisions Work In Subversion,” on page 565.

The Revision History Page For a Project

You have more options when you right-click a project object in version control and then select *History*.

Figure 19-9 *Revision History of Projects*



If you select *History* for a project object, the Revision History page allows you to select a version of a project object from the list of revisions. You can then view the contents of earlier versions and bring those versions up-to-date with your latest revision.

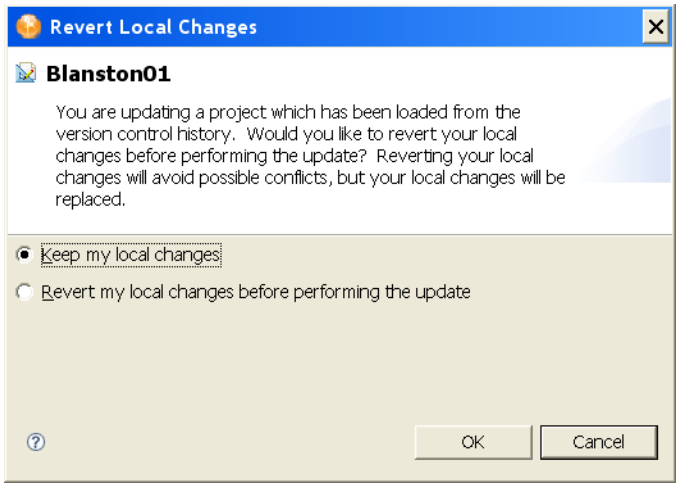
Get Revision

Select the revision for the project you want to work with, then click *Get Revision*. Answer *Yes* to save all of the editors in this project. That version of the selected object is downloaded from the version control server and becomes the version of the project you are working on.

If you select an older version of a project, the project has a special status icon in the Version Control view. This icon indicates that your project came from history instead of being out-of-date, but its status returns to normal after you select *Update*.

If you make changes to the historical version and select *Update*, you are presented with a Revert Local Changes page, allowing you to keep your local changes or to revert your local changes.

Figure 19-10 *Reverting Local Changes*

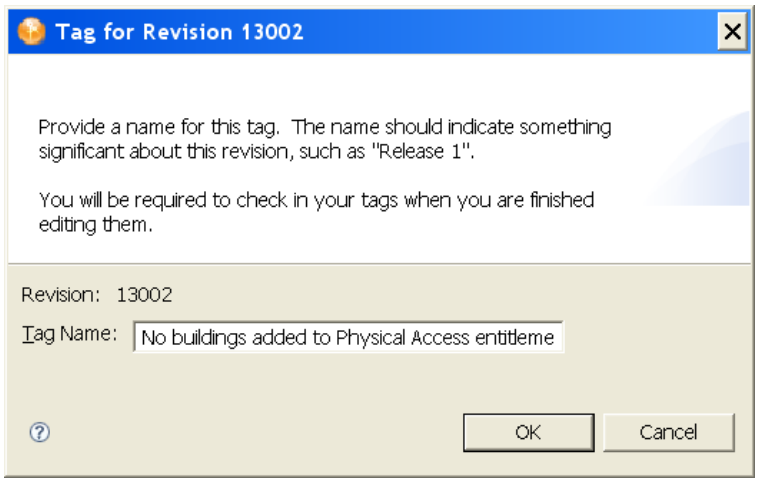


If you have made deliberate changes and want to now save those changes to the version control server, select *Keep my local changes* (default). If you made inadvertant changes to the project, or if you just wanted to see what was in this historical version, select *Revert my local changes before performing the update*.

Creating a Tag for a Project

If you select a project object, you can create a tag for any of the revisions listed in the Revision History page. This allows you to give a revision project a more memorable name instead of a revision number. To create a tag, right-click a revision and select *Create Tag*. This brings up the Tag for Revision page.

Figure 19-11 *Adding a Tag To a Selected Revision*



Provide a tag name that is significant to this version of the object and click *OK*. The tag name is added under the *Tag* heading in the Revision History page. When you close the Revision History page, you are asked to add a comment to all of the tag names that you have added.

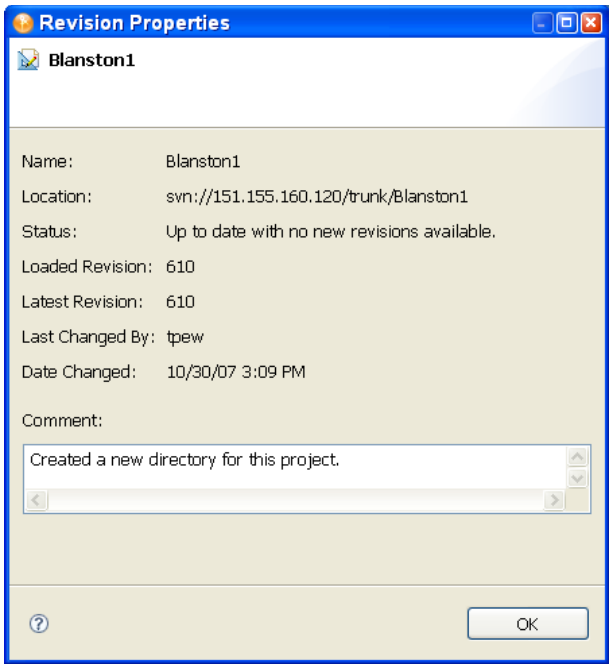
Comparing Versions

See “Comparing Revisions” on page 428.

Properties

Use the *Properties* option to view the properties of an object that has been added to version control.

Figure 19-12 *Object Properties Page*



Important information includes the location of the object on the version control server, the loaded revision number, the latest revision number, and any comment concerning the most recent check-in. You cannot make changes to this information.

19.5 Comparing Revisions and Resolving Conflicts

- Section 19.5.1, “Comparing Revisions,” on page 428
- Section 19.5.2, “Resolving Conflicts,” on page 431
- Section 19.5.3, “The Modeler View Layout In a Team-Enabled Environment,” on page 434
- Section 19.5.4, “Provisioning Objects,” on page 436

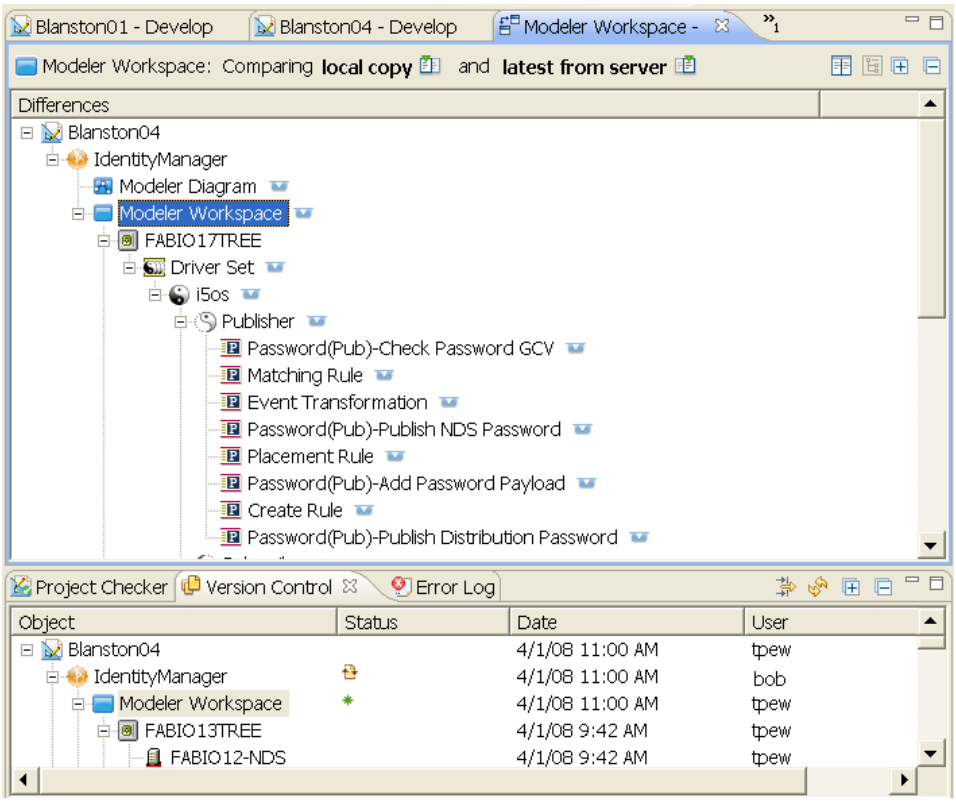
19.5.1 Comparing Revisions

Use the *Compare Revisions* option to compare what has changed between your local copy and the latest copy on the version control server. You can compare any object that has been checked in to the version control server. Use this option to compare historical versions to your local copy, or to other historical versions.

NOTE: For the *Compare Revisions* option to work, you must be able to communicate with the version control server. If the version control status icon at the lower right of Designer is grey , Designer is not communicating with the version control server. Mouse over the version control status icon for further connection information.

To use the Compare Revisions option, select a project or any other object in the Version Control view and select *Compare Revisions*.

Figure 19-13 *Comparing Revision Changes*



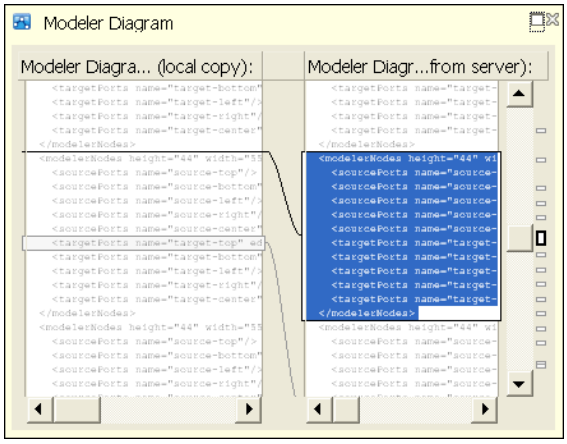
The Compare view appears in the main editor section of Designer and is displayed as a tree with the object highlighted. The top bar indicates the object that is selected and which revisions are being compared.

Version control uses a left-to-right display of information. The left side shows local copy information and the right side shows the version from the version control server. Because there is no information in the Outline view, you can double-click the *Compare view* tab to expand the view to fill Designer. Double-click the *Compare view* tab again to have it return to its normal size, or click the *Restore* button in the lower right corner.

You can select the *Change left-side revision* or *Change right-side revision* buttons to view the other versions that you have saved to the version control server. For example, if you want to compare your local copy to a different version on the server, click the right-side button. If you want to compare the server version to an earlier server version, click the left-side button. When you select a different version from the History page, the top bar title changes to reflect the different copy comparisons. Click the *Expand All* or the *Collapse All* icon to expand/collapse all items in the Compare view.

To see a snapshot of the changes in an object, click the overview icon to the right of the object to bring up the Overview page for the selected object.

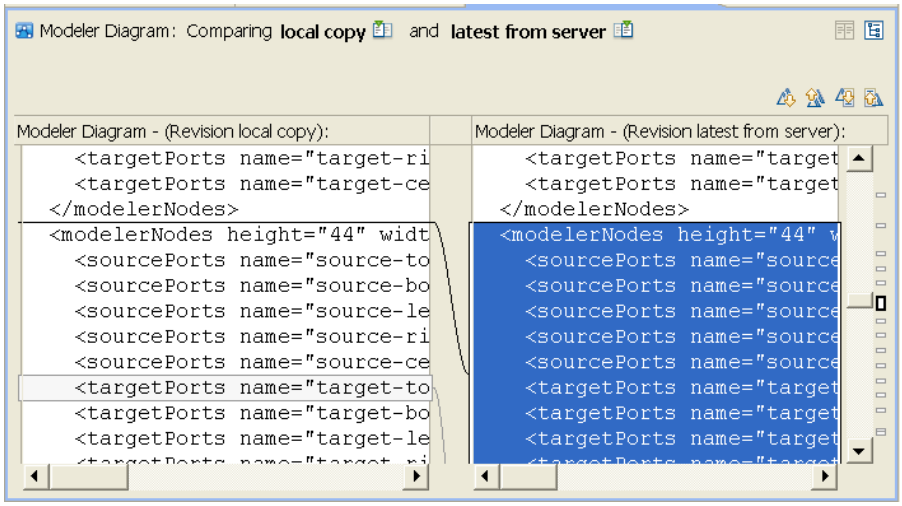
Figure 19-14 *Viewing a Quick Overview of Changes*



If the object you selected is made up of more than one file, you see a drop-down menu listing the files. Select a file from the menu to view the changes to that file.

To view the actual changes in more detail, click the Expand button in the Overview page or double-click the object in the tree view. You can also click the *Compare selected item* button next to the tree-view button.

Figure 19-15 *Double-click the Object To See a Detailed Description of the Changes*



You can use the Next Difference/Previous Difference icons or the Next Change/Previous Change icons to move between the file's changes. You can also click the blocks on the right side to jump to the file's changes. After you have drilled down and have seen the differences at an object level, click the tree-view button to return to the tree view.

When To Use Compare Revisions

There are three good reasons to use the *Compare Revisions* option.

- **Finding Problems.** You can use the *Compare Revisions* option to locate when a specific problem was introduced to a project. You can determine when a change was made, who made that change, and why the change was made. If someone on your team broke a policy, you can see when it was broken, who broke it, and what their comment was when they checked it in.
- **Change Overview.** You can also use the *Compare Revisions* option to get an overview of the changes that have been made to a project. By choosing different revisions, it is easy to see all of the changes that were made to a project in a given period of time.
- **Conflict Resolution.** The *Compare Revisions* option can help you resolve conflicts. When you compare your local version and the latest from the server, the conflicts are highlighted in red and you can see the specific conflicts. See Section 19.5, “Comparing Revisions and Resolving Conflicts,” on page 428.

19.5.2 Resolving Conflicts

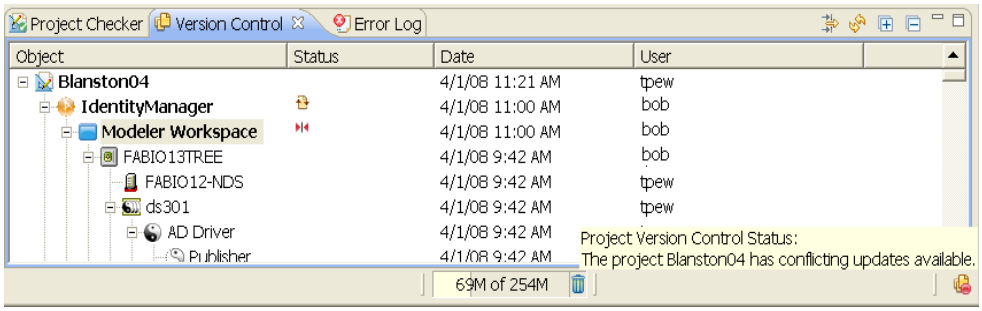
- “Example 1: Checking In Changes to the Same Object” on page 431
- “Example 2: Core Model Object Conflicts” on page 433
- “Example 3: Deleted Projects” on page 433

Example 1: Checking In Changes to the Same Object

If Bob and Terri are working on a project and they both try to edit the object in the version control server at the same time, they have a conflict.

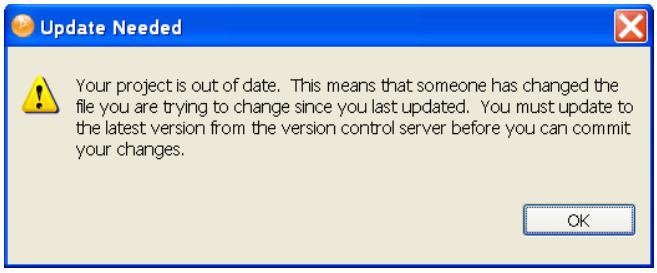
Suppose Bob checks in first. Designer is communicating with the version control server in the background and collects status information on all of the objects that are checked out. If there is a conflict, the *Version Control Project Status* icon changes to red and Terri sees a warning message when she mouses over the icon.

Figure 19-16 *Receiving a Conflict*



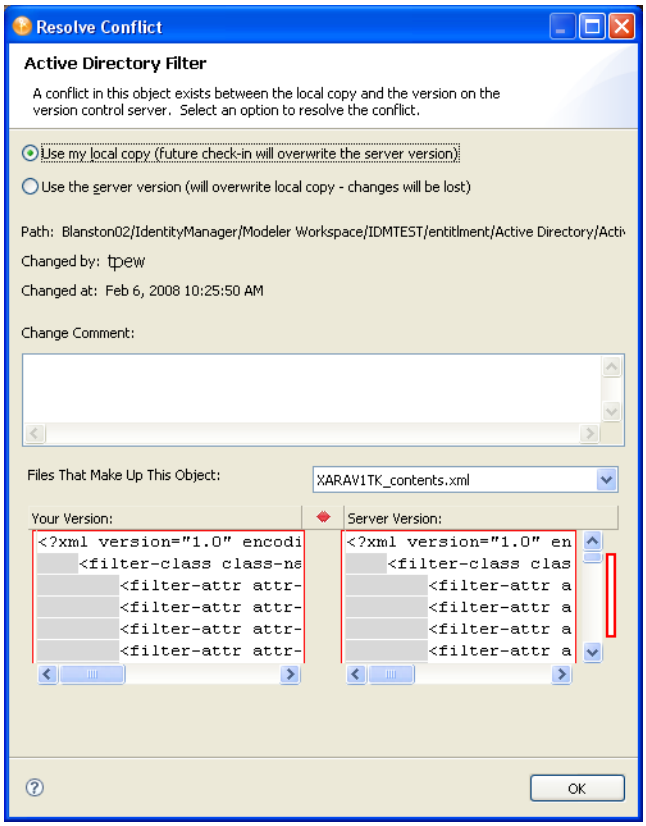
When Terri attempts to check in, she receives an error message telling her to update before she checks in.

Figure 19-17 *Conflict Message*



If she clicks *OK* and performs the update, version control tries to automatically merge the differences between Bob's and Terri's changes. However, if their changes cannot be automatically merged and Terri tries to update, she sees the Resolve Conflict page, allowing her to see the differences between her local version and the version on the version control server.

Figure 19-18 *Choose Either the Local Version or the Checked-In Version*

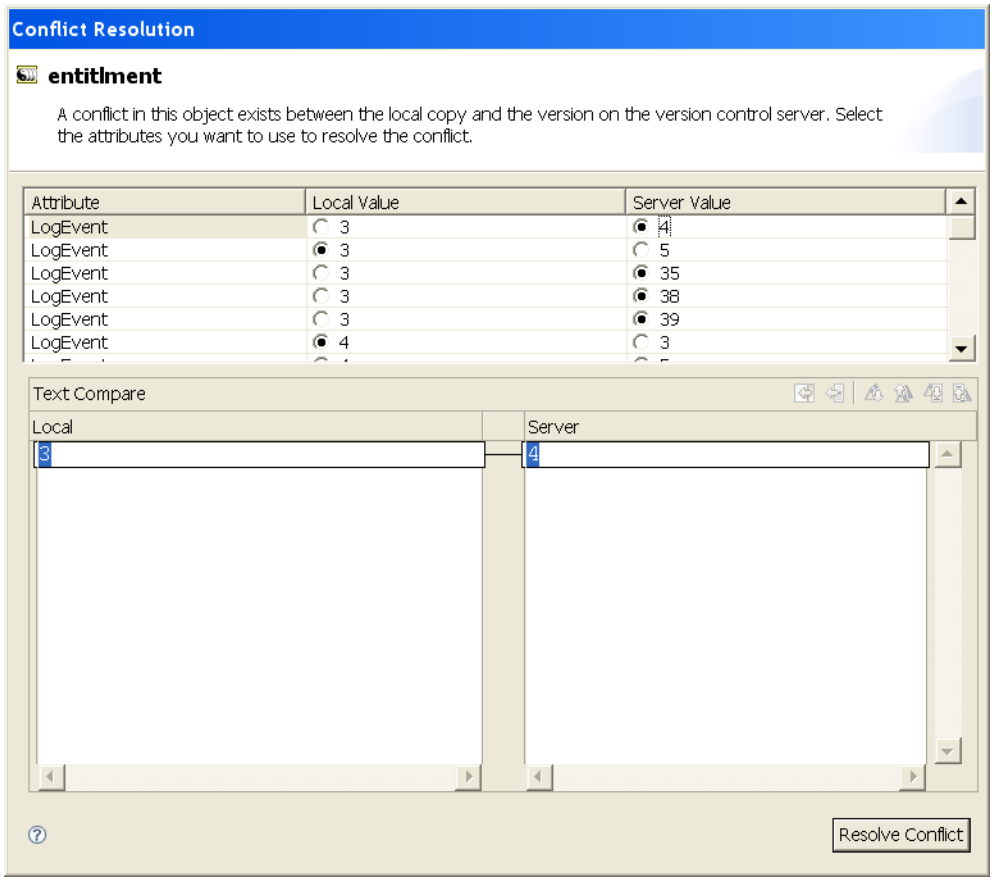


The red markers on the right side of the Resolve Conflict page show the data that is in conflict, and the blue markers show the modified local data. Terri can then choose to either keep her local version or to overwrite her local version with the one on the version control server. The Resolve Conflict page also shows the path of the file with the conflict.

Example 2: Core Model Object Conflicts

In some conflicts, the core model objects can merge manually at an attribute level, allowing you to change the attributes so that they are no longer in conflict. If the conflict is of this nature, you see the Conflict Resolution page, allowing you to manually resolve the conflicts.

Figure 19-19 *Resolving Attribute Conflicts*

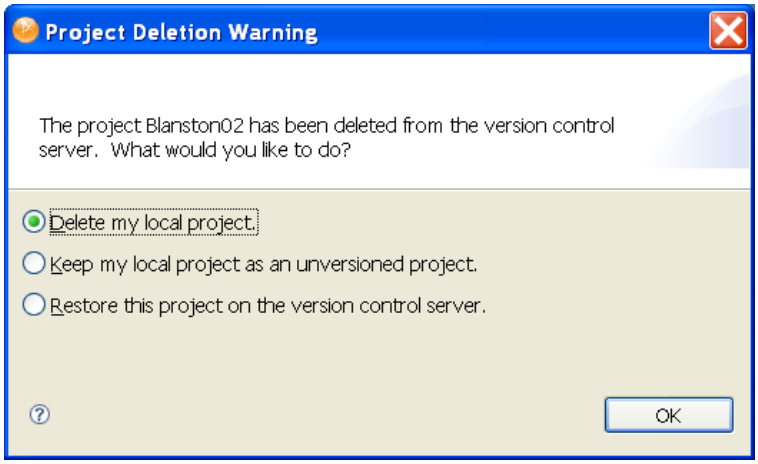


Once you have made the necessary attribute changes, select *Resolve Conflict*.

Example 3: Deleted Projects

If the project has been deleted from the version control server, you are given three choices: delete the local project, keep the local project as an unversioned project, or restore the project on the version control server.

Figure 19-20 *Choosing What to Do with Deleted Projects*

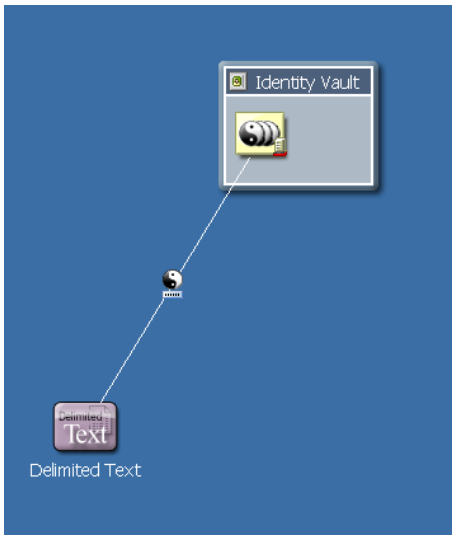


19.5.3 The Modeler View Layout In a Team-Enabled Environment

Designer 4.0 handles saves by multiple users in a complex manner. Your personal Modeler view layout in a team environment changes as others change their Modeler view layout and check in their changes to the version control server. When you perform an Update from the version control server, you get the last Modeler view layout that was checked into version control. Note that it's just the layout that is changing and not the data.

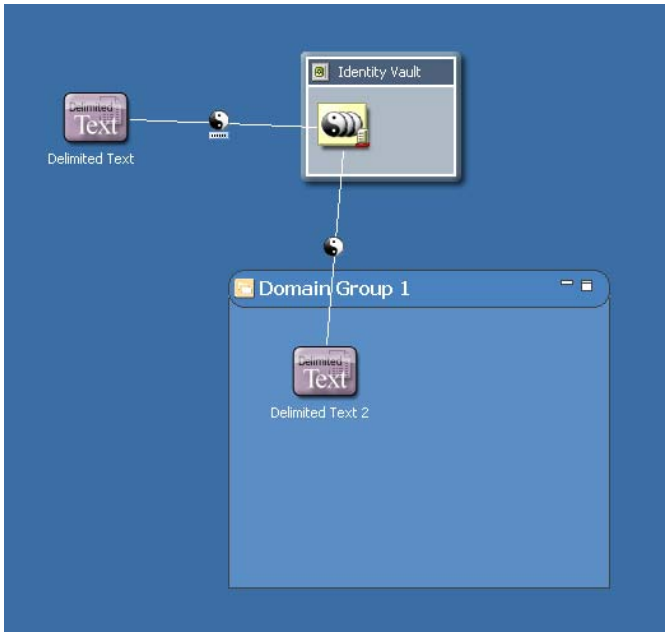
For example, suppose Bob and Terri are working on a new project. Terri creates the project and checks the project into version control.

Figure 19-21 *Terri Creates a New Project*



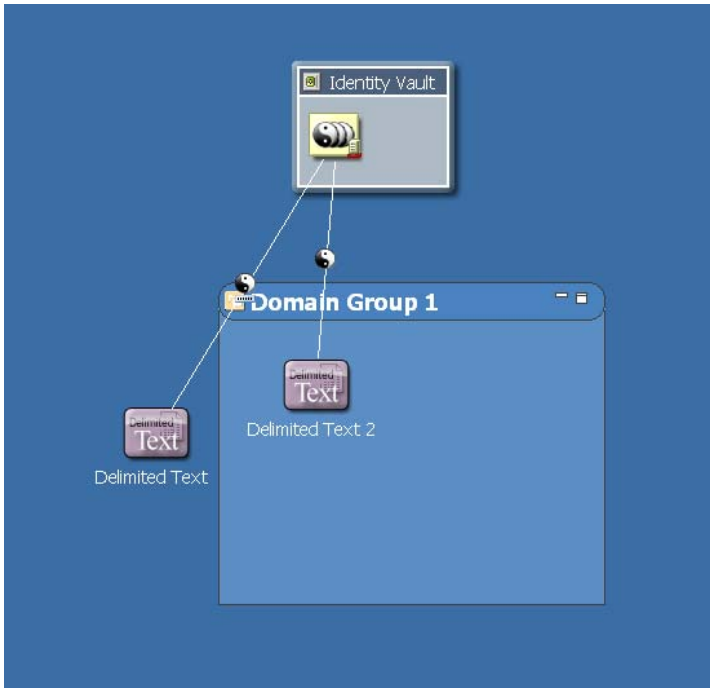
Terri tells Bob about the new project and Bob imports the project from the version control server. Bob then adds a domain group and another driver, and checks those changes into version control.

Figure 19-22 *Bob Adds Information to the Project and Checks It Back Into Version Control*



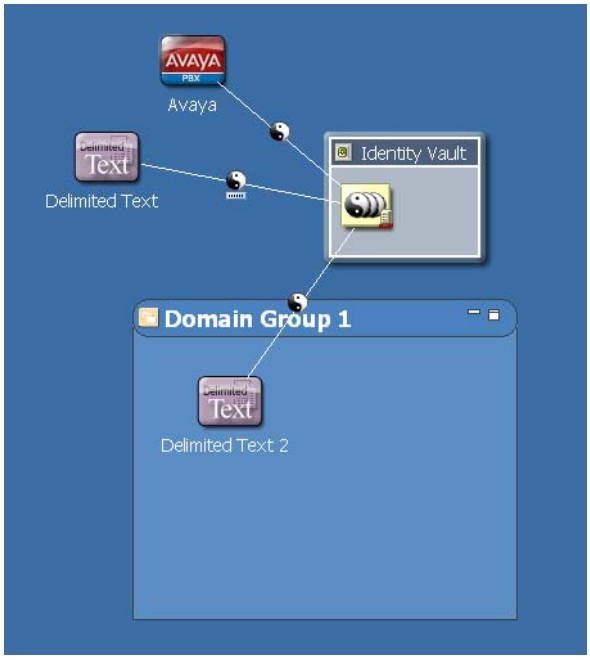
During this time, Terri was working on the first driver and made only minor changes to the Modeler view, but they were enough to create local differences. When Terri saves her changes locally, then updates the project from the version control server, she sees that her Modeler view changes are merged with Bob's Modeler view changes.

Figure 19-23 *Terri's Modeler View Changes are Merged with Bob's*



However, if Bob changes the Modeler layout again (and checks in) and Terri does not (no conflict), Terri gets Bob's Modeler layout the next time that Terri updates from the version control server.

Figure 19-24 *Bob's Last Check-in*

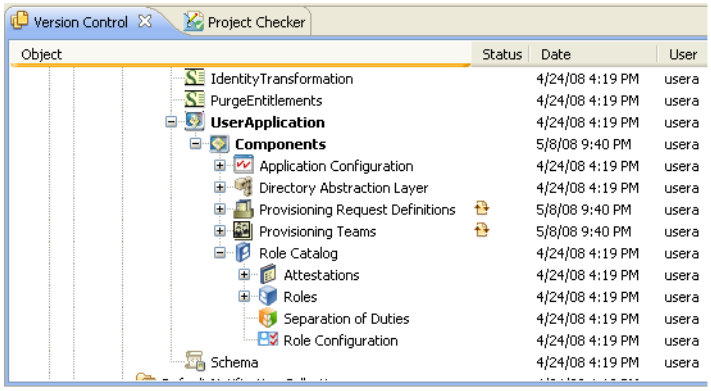


For best practices, define a Modeler layout that the team can live with and leave it alone.

19.5.4 Provisioning Objects

In Designer 4.0, provisioning objects such as the directory abstraction layer, Provisioning request definitions, teams, and roles, can all participate in version control. The Version Control view below illustrates how provisioning objects appear.

Figure 19-25 *Provisioning Objects in the Version Control View*



The Version Control view reflects provisioning objects in a slightly different hierarchy than the Outline view. Under the *User Application* entry, you see a node called *Components*. This is the main node under which all provisioning objects are located. *Application Configuration* and *Locale Configuration* are also new nodes in the tree. System objects and unsupported objects are also visible in the Version Control view.

19.6 Version Control Best Practices

Managing a team environment with version control can be a challenging task. Combining version control and Identity Manager Designer has its own set of issues. This section includes some tips and best practices for using version control with Designer.

We encourage you to send in your version control best practices so we can make the user experience better for our product. Send your version control best practices via email to dvandenbos@novell.com. Entitle the correspondences “Version Control Best Practices.”

- Section 19.6.1, “Best Practices,” on page 437
- Section 19.6.2, “Best Practice Scenarios,” on page 437
- Section 19.6.3, “Subversion and Version Control Interaction Rules,” on page 441

19.6.1 Best Practices

- **Coordinate all Designer upgrades with your entire team.** When you upgrade to a new version of Designer, many of the files in your project are changed by the project converter, so you need to coordinate with the rest of your team. In the ideal upgrade process, everyone checks in all of their changes, one team member runs the project converter and checks in the converted project, then everyone installs the new version of Designer and re-imports the project.
- **Coordinate deployment.** When you are using version control and the same eDirectory server with multiple people, it is possible to overwrite changes. You should coordinate deployment with your team members to make sure that you do not overwrite other team members’ changes. Best practice is to assign one person to deploy a project to a production environment.
- **Assign policies.** Assign one team member to a policy rather than having multiple team members work on one policy. Multiple team members writing and modifying shared policies in a driver is a recipe for disaster.
- **Define an acceptable modeler layout for the team.** Personal Modeler layouts in a team environment are only maintained if there is a version control conflict on the Modeler layout between your Modeler view layout and another’s Modeler view layout. If there is no conflict and you perform an Update from the version control server, you get the last Modeler layout that was checked into version control. Choose a Modeler layout that the team can live with and leave it alone.

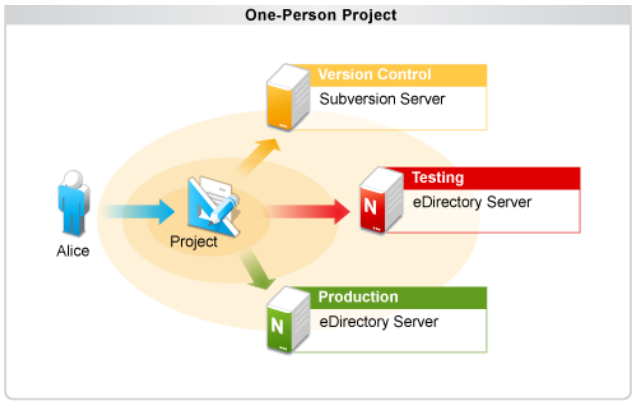
19.6.2 Best Practice Scenarios

There is no one-size-fits-all scenarios for using version control with Designer. This section identifies some user situations that we used for best practice scenarios. These scenarios are specific step-by-step guides to be used in addition to those outlined in the Best Practices section.

- “One-Person Project” on page 438
- “Small Team with One Shared eDirectory Server” on page 438
- “Small Team with Individual eDirectory Servers” on page 439
- “Medium-Sized Team with a Shared Test and Production Environment” on page 440
- “Single Consultant Working for Multiple Companies” on page 441

One-Person Project

Figure 19-26 *One-Person Project*



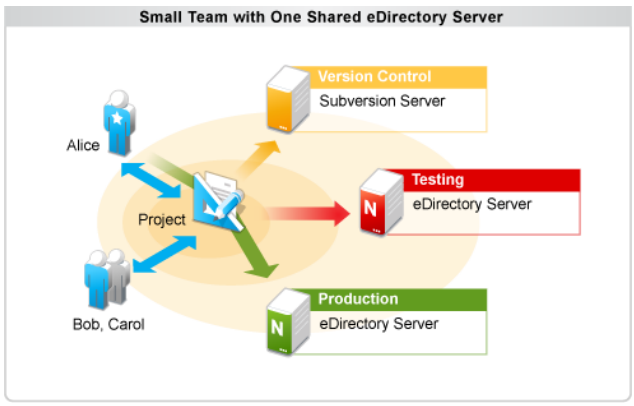
Version control is very useful in a team environment, but it is also very useful in an individual environment. Version control allows a single developer to make backups, restore older versions, and have the freedom to explore project changes without risking data.

Alice decides to work on a project alone. She creates a new project and checks that project into the version control server. She makes changes to the project and deploys them to a development server for testing. She frequently checks her changes into the version control server so she can easily explore the history of her project later.

Alice can optionally use tagging to specify which project revisions are stable revisions. If she is unsatisfied with any project changes, she can revert those changes or get an older copy of her project from history. When she is happy with her changes, she deploys the project to an eDirectory server in the production environment.

Small Team with One Shared eDirectory Server

Figure 19-27 *Small Team Scenario #1*



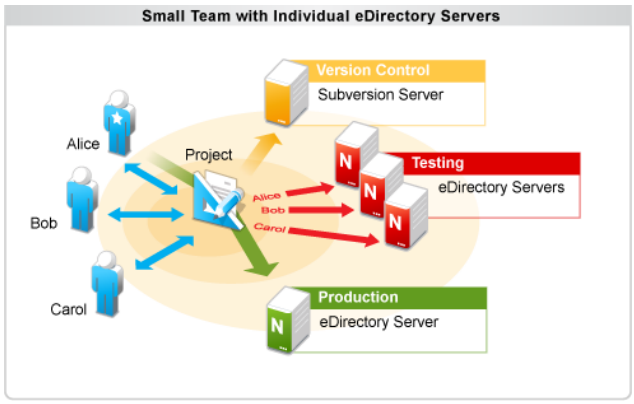
Alice, Bob, and Carol are working together on a project. Alice (the administrator) creates the new project and checks it into the version control server. Bob and Carol import that project and they all work on the project together. Alice, Bob, and Carol agree on ownership of Identity Manager objects and do not often edit each other's objects. Everyone is diligent about updating frequently in order to avoid conflicts.

When Alice, Bob, or Carol want to deploy their changes to the shared development environment, they are careful to deploy just their own changes and not overwrite each other's changes. They all deploy to the same shared development server so they can test their changes in the same environment. When each team member is happy with the results, they check in their changes to the version control server.

When they are ready to deploy their project to an eDirectory server in the production environment, Alice performs an update to get the latest changes from the version control server and then deploys the project to the production server. Alice manages all deployment to the production server so the team maintains control over the changes in the production environment.

Small Team with Individual eDirectory Servers

Figure 19-28 *Small Team Scenario #2*



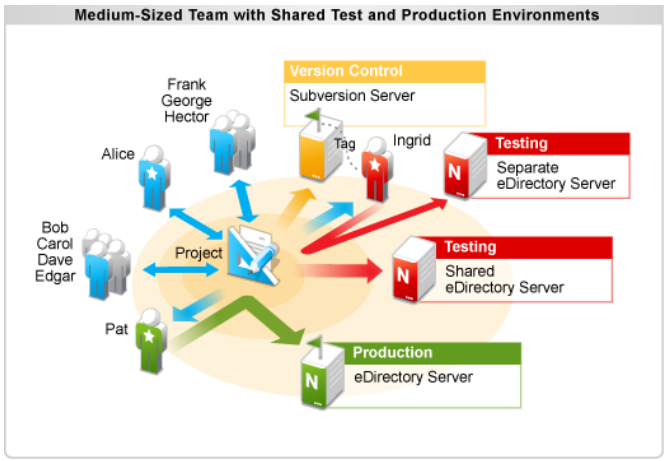
Alice, Bob, and Carol work together on a project. Alice (the administrator) creates a new project and checks it into the version control server. Bob and Carol then import that project and they all work on the project together. Alice, Bob, and Carol don't need any boundaries for object editing and they are all welcome to edit every object in the project. They update frequently and resolve conflicts when they occur.

Alice, Bob, and Carol each have their own eDirectory development server to deploy to and can deploy changes without the need to consult each other. They change, deploy, and test their changes and then check them into the version control server.

When they are ready to deploy to the production server, Alice updates her project to get the latest changes from version control and then deploys them to her development server. After she has verified that everything works as expected, she deploys the changes to the eDirectory server in the production environment. Alice manages all of the deployment to the production server to make sure it is a controlled environment.

Medium-Sized Team with a Shared Test and Production Environment

Figure 19-29 *Medium Team Scenario*



Alice, Bob, Carol, Dave, and Edgar all work together on a project. Frank, George, and Hector work part-time on this project and consult for other projects. Alice (the administrator) creates the project and checks it into the version control server. Bob, Carol, Dave, and Edgar import the project from the version control server and they all begin working on the project and deploying to the same eDirectory development server.

Frank, George, and Hector work mostly in an advisory capacity and do not own any objects in the project. They consult with Alice before making changes. Frank, George, and Hector are careful when they deploy changes so that they don’t overwrite the changes of the object owners.

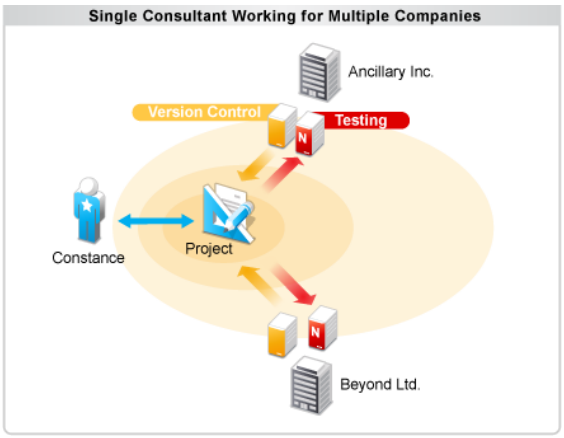
Alice, Bob, Carol, Dave, and Edgar mostly focus on changing their own objects, but Ingrid (the integration test engineer) focuses on testing the entire project on a separate development server. She imports the project from version control and updates frequently to get changes from the rest of the team. She deploys those changes in the controlled development environment and tests them there. Ingrid makes only the changes necessary to deploy to the test server and does not check any changes into the version control server.

When Ingrid is satisfied with a version of the project, she creates a project tag in version control and certifies that revision of the project as deployable to the production environment. She then asks Pat (the production environment administrator) to deploy the project to the production server and tells him which tag should be deployed.

Pat imports the project from the version control server. He then uses the Get from History function to get the specific revision that Ingrid has tagged. After he has that version, he makes only the changes necessary to deploy the project to the production server and deploys the project. The rest of the team can continue to work on the project during this time because Pat has locked his version of the project to the revision that Ingrid has certified as deployable to the production environment.

Single Consultant Working for Multiple Companies

Figure 19-30 *Working for Multiple Companies*



Constance (the consultant) works for multiple companies, helping them with their Identity Manager projects. On Monday, she works for Ancillary Incorporated. She imports the project from the version control server at Ancillary Inc. and deploys the project to the Ancillary development server. Constance communicates frequently with the Ancillary Inc. team members and makes sure to never overwrite the objects from the Ancillary Inc. team on the eDirectory production server.

On Tuesday, Constance works for Beyond Limited. She closes the Ancillary project and imports the project from the Beyond Limited version control server. She follows established procedures when working with the Beyond Limited team and carefully separates the changes for each company.

19.6.3 Subversion and Version Control Interaction Rules

- **Do not use the Subversion command line.** People familiar with the Subversion command line might be tempted to use it with Designer to perform simple commits or updates. Designer has many tools to manage the merging and object dependencies within an Identity Manager project. Using the Subversion command line bypasses these tools and can easily lead to a corrupted project and data loss.
- **Do not use other Subversion clients.** Tortoise, Subclipse, or any other Subversion client can cause the same problems as the Subversion command line. Do not use them on the same working copy you are using for Designer.
- **Disable Subclipse before using version control with Designer.** Subclipse is a great tool, but it does not work well with the Subversion integration in Designer for Identity Manager. It is not enough to just not use Subclipse, you must make sure it is disabled if you have it installed.

  This only affects those who run Designer inside Eclipse instead of using the Designer install, so most users will not see this problem.

Setting Preferences

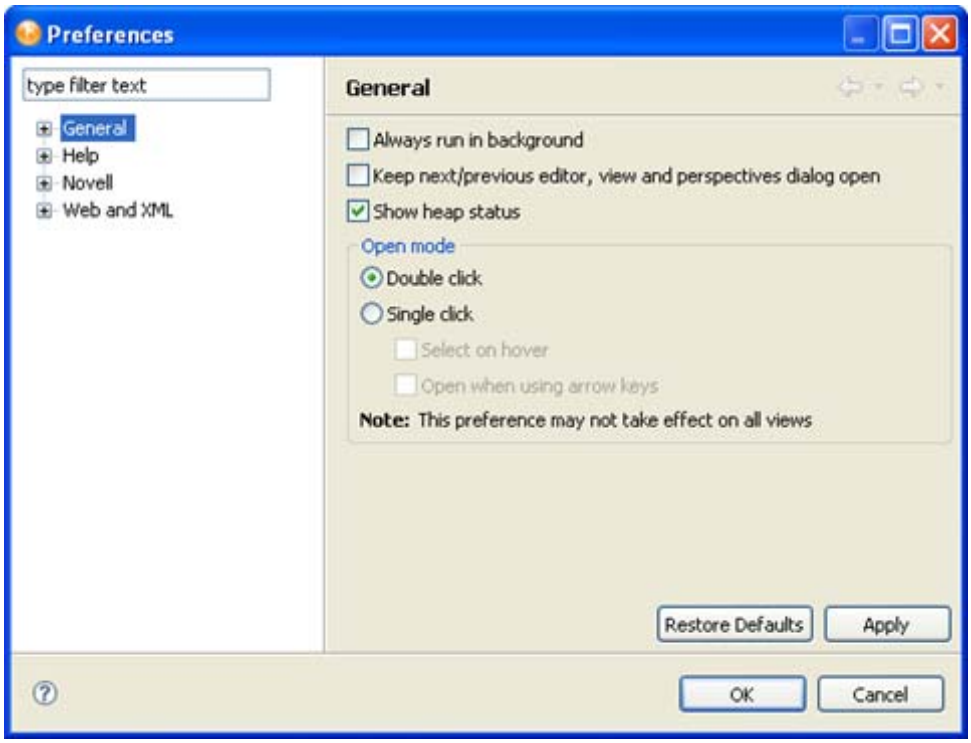
20

- Section 20.1, “Finding Preference Pages,” on page 443
- Section 20.2, “General,” on page 443
- Section 20.3, “Help,” on page 452
- Section 20.4, “Novell,” on page 452
- Section 20.5, “Web and XML,” on page 474

20.1 Finding Preference Pages

You customize Designer by setting options in Preferences.

1 From the main menu, select *Window* > *Preferences*.



2 Select a heading (for example, *Novell*) or navigate to a sub-heading.

3 Make changes, then click *Apply* or *OK*.

20.2 General

The General preferences page includes the following settings:

Table 20-1 *Preferences: General*

Setting	Description
	Searches all the preferences and shortens the tree view, depending upon what you type in the edit box.
Always run in background	Enables operations to run in the background without disturbing you.
Keep next/previous part dialog open	Keeps the editor and view dialog boxes open when an activation key is released. Normally, the dialog box closes as soon as the key combination is released.
Show heap status	Places a field in Designer's bottom right corner and displays the amount of memory being used of total memory available.
Open mode: Double click	Opens a project when you double-click it.
Single click: Select on hover	Selects the setting when the cursor hovers there.
Single click: Open when using arrow keys	Opens the setting when you select it.

Additionally, the following preferences categories appear as General sub-pages:

- Section 20.2.1, "Appearance," on page 445
- Section 20.2.2, "Capabilities," on page 446
- Section 20.2.3, "Compare/Patch," on page 447
- Section 20.2.4, "Content Types," on page 447
- Section 20.2.5, "Editors," on page 448
- Section 20.2.6, "Keys," on page 450
- Section 20.2.7, "Network Connections," on page 450
- Section 20.2.8, "Perspectives," on page 450
- Section 20.2.9, "Startup and Shutdown," on page 451
- Section 20.2.10, "Web Browser," on page 451
- Section 20.2.11, "Welcome," on page 451

20.2.1 Appearance

Figure 20-1 *Preferences: General > Appearance*

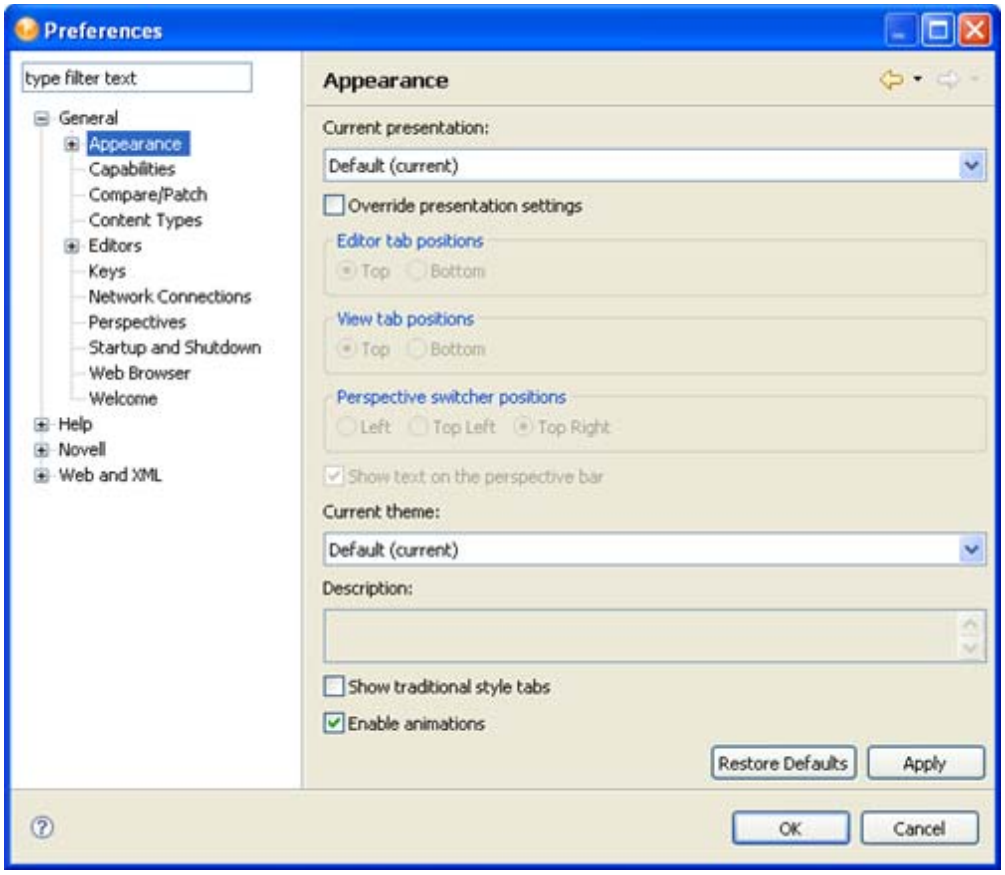


Table 20-2 *Preferences: General > Appearance*

Setting	Description
Current Presentation	Allows you to choose between Designer 4.0 presentation, the current presentation, or the Eclipse 2.1 style presentation.
Override presentation settings	Alters how the tabs and views appear in the workbench
Editor tab positions	Positions tabs on the Modeler, Novell XML Editor, or Text Editor at the top or bottom.
View tab positions	Positions view tabs (for example, the Project view tab) at the top or bottom of views.
Perspective switcher positions	Positions the Perspective Switcher at the left, top left, or top right of the workbench.
Show text on the perspective bar	Determines whether text (for example, Designer) displays next to the icons in the Perspective Switcher.

Setting	Description
Current theme	The general theme (colors and fonts) that Designer uses. Choices are Default (current), reduced pallette, and R 3.5 theme.
Show traditional style tabs	Displays square Windows-style tabs. The alternative is rounded tabs.
Enable animations	Animates views (for example, Fast Views) and editors that you minimize, maximize, or restore. Reinforces tasks in Designer.

Colors and Fonts

To change a color:

1. Under *General*, expand *Appearances*.
2. Select *Colors and Fonts*.
3. Expand an option (for example, *Basic*).
4. Select an item (for example, *Active hyperlink text color*).
5. Click the color button.

6. Select a color from the Color palette, then click *OK*.

To change a font:

1. Under *General*, expand *Appearances*.
2. Select *Colors and Fonts*.
3. Expand an option (for example, *Basic*).
4. Select an item (for example, *Banner Font*).
5. Click *Change*.
6. Select a font, style, and size, then click *OK*.

20.2.2 Capabilities

Currently, this section relates to Eclipse functionality. Designer functionality will be implemented later.

Prompt When Enabling Capabilities

Eclipse lets you turn off capabilities. However, Eclipse sometimes detects that you need the capability, depending on how you use the product, and automatically turns the capability on again. You can select *Prompt when Enabling Capabilities* to receive a message when this happens. This feature doesn't apply to Designer 4.0.

20.2.3 Compare/Patch

This Eclipse functionality customizes the behavior of the comparison editor. When you select to compare or synchronize two or more resources in the Workbench, one or more comparison editors usually open.

Table 20-3 *Preferences: General > Compare/Patch General Tab Settings*

Setting	Description
Open structure compare automatically	Makes visible an additional information area that shows differences in the underlying structure of the resources being compared. This information might not be available for all comparisons. The default is *On*.
Show additional compare information in the status line	Causes the status line to display additional context information about the comparison. The default is *Off*.
Ignore white space	Causes the comparison to ignore differences that are white space characters (for example, spaces and tabs). Also causes differences in line terminators (LF versus CRLF) to be ignored. The default is *Off*.
Automatically save dirty editors before patching	Controls whether any unsaved changes are automatically saved before a patch is applied. The default is *Off*.

For information about settings on the *Text Compare* tab, refer to the Eclipse documentation.

20.2.4 Content Types

Table 20-4 *Preferences: General > Content Types*

Pane	Description
*Content type*s	The type of content (for example, HTML or XML) that a file contains.
File associations	The file extension that is associated with a content type. For example, `.xml` is associated with a file that contains XML content. To add a file association: 1. Select a content type. 2. Click *Add*. 3. Define a new file type, then click *OK*.

20.2.5 Editors

Table 20-5 *Preferences: General > Editors*

Setting	Description
Size of recently opened files list	The number of files to add to the file menu of recently opened files, which you can easily reopen.
Show multiple editor tabs	Displays tabs for all opened projects. If you deselect this option, only one editor tab displays, and an abbreviated name displays on the tab.
Restore Editor state on startup	Displays the editor in the same state as it was when last closed, as opposed to using default settings.
Prompt to save on close even if still open elsewhere	Saves the file on close even if the same file is open in another editor.
Close editors automatically	Automatically closes the first-opened editor when you open additional editors. This option prevents displaying too many editors and cluttering the workbench.
Number of opened editors before closing	Determines how many editors can be open. For example, if you specify two and then open a third project, the first-opened project automatically closes.
When all editors are dirty or pinned	Prompts you to save unsaved components in the project that is about to automatically close, or to open an additional editor.

- "File Associations" on page 448
- "Hex Editor" on page 449
- "Structured Text Editors" on page 449
- "Text Editors" on page 449

File Associations

Enables you to associate editors (whether they are internally installed in the Designer, or an external application) with file types (extensions) so that you can edit files.

To find out which editor is associated with a file type, select the file type. For example, a `.docgen` file type is associated with Style editor, but a `.scriptpolicy` file type is associated with Policy Builder.

To associate an additional editor with a file type:

1. Select the file type.
2. In the *Associated editors* pane, click *Add*.
3. Select an additional editor, then click *OK* twice.

To add a file type:

1. In the *File types* pane, click *Add*.
2. Type the extension (for example, `.doc`) for the file type, then click *OK*.
3. In the *Associated editors* pane, click *Add*.
4. Select an editor for that file type, then click *OK* twice.

Hex Editor

Enables you to configure Designer's hex editor environment, including font, font style, and colors. You can also associate, or disassociate, the hex editor from Designer's registered file extensions, and enable hex editor logging.

Structured Text Editors

For information on structured text editors, refer to the Eclipse documenation.

Text Editors

Table 20-6 *Preferences: General > Text Editors*

Setting	Description
Undo history size	Determines the size of the undo history. The default is 200 changes.
Displayed tab width	Specifies the number of characters or spaces in a tab character. The default is 4. The maximum is 16.
Insert spaces for tabs	Inserts the number of spaces specified in *Displayed Tab Width*, instead of a tab character, when you press the tab key in the text editor.
Highlight current line	Highlights the current line.
Show print margin	Displays the print margin on the right side of the text document. A vertical line identifies the margin.
Show line numbers	Numbers each line in the editor.
Show range indicator	Displays a range indicator.
Show whitespace characters	Displays white space characters so you can see them in the text editor.
Enable drag and drop of text	Allows you to drag and drop text within the text editor.
Warn before editing a derived file	Notifies you if you attempt to edit a file generated or maintained by the system. Your changes might be overwritten.
Smart caret positioning at the line start and end	Enables the Home and End commands to move to the first and last non-whitespace character on a line.

Setting	Description
Show affordance in hover on how to make it sticky	Enables the hover over text to grab the text and place it in the clipboard.
Appearance color options	Lets you configure the display settings for the text editor. Select a particular appearance characteristic from the list to view and change the display settings for that characteristic.

For additional information on text editors, see the Eclipse documentation.

20.2.6 Keys

Enables you to view a table of all of the keyboard mappings, change those mappings, and add new mappings.

20.2.7 Network Connections

Enables you to configure a manual proxy configuration if you use a proxy server to access the Internet. The default is a direct connection to the internet.

20.2.8 Perspectives

Table 20-7 *Preferences: General > Perspectives*

Setting	Description
Open a new perspective	*In the same window*: Places a new icon in the Perspective Switcher, so that you can toggle between perspectives in the same window. *In a new window*: Opens a new perspective in a different window. You can toggle between perspective windows by selecting icons on the taskbar.
Open a new view	*Within the perspective*: Opens the view so that it is contiguous to the Modeler. *As fast view*: Opens the view and places a Fast View in the bottom left corner of the perspective.
Open the associated perspective when creating a new project	Determines how and when you switch to an associated editor when you open a perspective.
Available perspectives	Designer is the default perspective. Other available perspectives are Eclipse Debug and Resource.

20.2.9 Startup and Shutdown

Table 20-8 *Preferences: General > Startup and Shutdown*

Setting	Description
Prompt for workspace on startup	Prompts you for a workspace folder. You can have multiple workspace folders and can specify a folder on startup.
Refresh workspace on startup	Synchronizes the workspace with resources (for example, `myfile.xml`) on disk.
Confirm exit when closing last window	Displays an *Exit Designer*? prompt when you exit Designer.
Plug-ins activated on startup	Lists plug-ins that are automatically loaded and registered.

20.2.10 Web Browser

Table 20-9 *Preferences: General > Web Browser*

Setting	Description
Use internal Web browser	Enables you to use an internal Web browser.
Use external Web browser	Enables you to add and use an external browser (for example, Netscape*). If you enable this option, you must also enable *Use External Browser* in the *Help* section (also found in Preferences).
External Web browsers	Lists browsers. To add a browser: 1. Click *New*. 2. Name the new browser. 3. Scroll to and select an executable (for example, `netscp6.exe`). 4. Specify a parameter, then click *OK*.

20.2.11 Welcome

Table 20-10 *Preferences: General > Welcome*

Setting	Description
Home: Home Page Theme	Enables you to select the theme that appears when you click *Help* > *Welcome*.

Setting	Description
Home: Root Pages	Adds tabs (for example, *Overview*) on the Welcome properties page. You add functionality by customizing these tabs.

For information about the Overview and What's New tabs, refer to the Eclipse documentation.

20.3 Help

Table 20-11 *Preferences: Help*

Setting	Description
Specify how help information is displayed: Use external browser	If an embedded Web browser is supported on your system, the Help view uses that browser to display help contents. To force help to use an external browser, enable this option. Specify an external browser in Preferences: *General* > *Web Browser*.
Open window context help	Determines whether the window context help opens in a dynamic Help view or in a pop-up window.
Open dialog context help	Determines whether the dialog box context help opens in a dynamic help section of the Help view or in a pop-up window.
Open help view documents	Determines whether the documents selected in the Help view open in place or in the editor area.

20.3.1 Content

Designer lets you include external information in the help system.

Table 20-12 *Preferences: Help > Content Settings*

Setting	Description
Include help content from a remote infocenter	Enables including external information in the help system.
Location	Specifies the hostname, path, and port to the external information.

20.4 Novell

The following Preferences categories appear as Novell sub-pages:

- Section 20.4.1, "Designer," on page 453
- Section 20.4.2, "Identity Manager," on page 456
- Section 20.4.3, "Provisioning," on page 474

20.4.1 Designer

The following preferences categories appear as Designer sub-pages:

- "DS Trace" on page 453
- "Language" on page 453
- "Project Checker" on page 454
- "Schema" on page 455
- "Trace" on page 455

DS Trace

This setting lets you configure DS Trace settings.

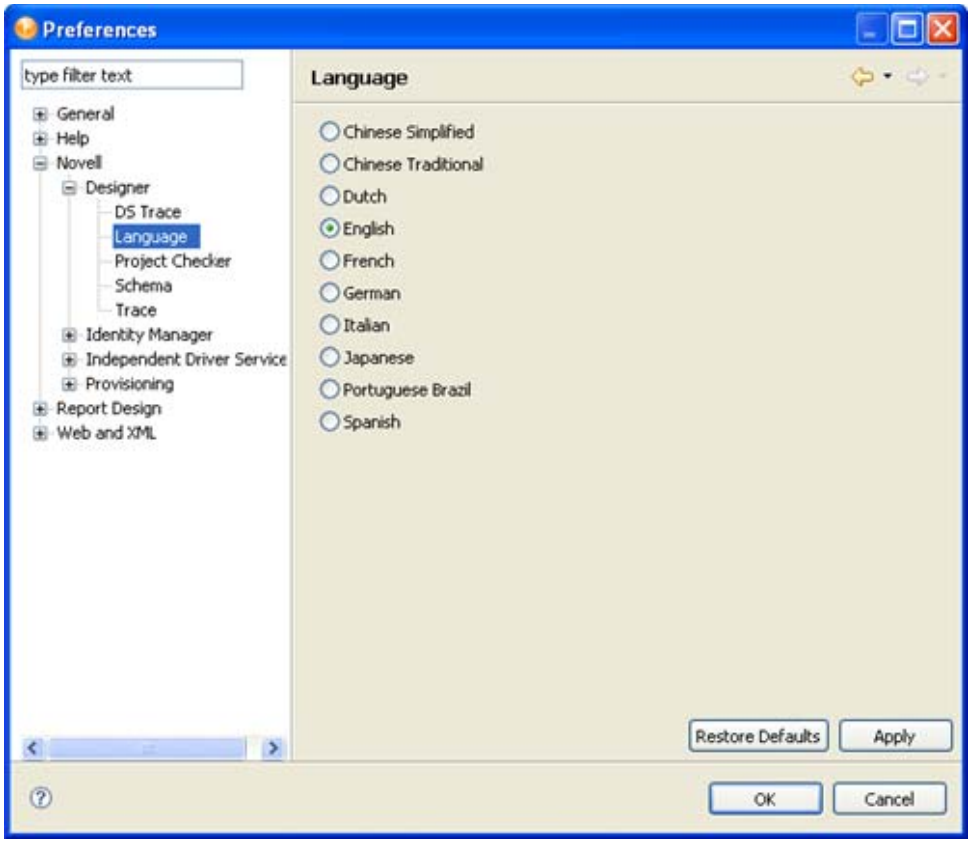
Table 20-13 *DS Trace Preferences*

Setting	Description
Live DS Trace Display	Specifies the size of the DS Trace window buffer, in lines (or entries). When the number of DS Trace entries exceeds the Window Size, DS Trace drops the oldest entry for each new entry it captures.
Auto-scroll display	Enables auto-scrolling of the live DS Trace window so that the latest log entries are always on screen. When deselected, you must manually scroll down the list of log entries.

Language

When you installed Designer, you selected a language to display Designer's UI. This setting enables you to change the language.

Figure 20-2 *Preferences: Novell > Designer > Languages Tab*



1. Select a language, then click *OK*.

   You must restart Designer for the language change to take effect.

2. Restart Designer.

NOTE: *Restore Defaults* reads the `config.ini` file, detects the previous language setting, and then defaults to that setting. When the changed property is written back to the `.ini` file, all comments are removed from the file. To preserve these comments, Designer copies the original `config.ini` to `config.ini.bak` and uses the backup to determine the default setting.

Project Checker

This settings lets you configure the Project Checker.

Table 20-14 *Preferences: Novell > Designer > Project Checker*

Setting	Description
Limit Visible Items to	Allows you to limit the number of items displayed in the Project Checker. The default value is 100.
Prompt me to save the editor before running Project Checker	Allows you to receive a prompt asking you to save your project before running the Project Checker. By default, this is enabled.

Schema

Allows you to manage the Identity Vault and connected system's schema.

Table 20-15 *Preferences: Novell > Designer > Schema*

Setting	Description
Warn when LDAP names are different from eDirectory names during .ldif import/export	Allows you to turn off this warning prompt, which appears during the import or export of the schema.
Warn when exporting base classes to .ldif	Allows you to turn off this warning prompt, which appears during the export of the schema.
Show the information message for the Manage Application Schema context menu	Allows you to turn off the information message that appears when managing the application schema.

Trace

The Trace view is useful in the following situations:

- To trace internal errors and messages, so that you can find out why something might not work as expected.
- To provide information for Novell Support, engineers, or other consulting resources.

All Designer-specific trace messages go to the Trace view if this view is open. Otherwise, no trace messages are sent.

Warnings and error messages are sent to the `.log` file, found in the run-time workspace `metadata` directory. Use the Error view to view this information.

Table 20-16 *Parameters: Novell > Designer > Trace*

Setting	Description
Enable tracing	Writes events to the Trace view. By default, tracing is off. To increase performance, disable tracing when you don't need it.
Include stack traces	Provides separate traces. Dumps the entire stack where an internal exception occurs, so that you can see in the code where the internal exception is failing.
Include XML processor traces	Provides separate traces that detail all of the processing of XML documents. This trace can become quite verbose.
Show plug-In names in the trace	In the Trace view, displays names of plug-ins where tracing has occurred. This is useful if you are tracing more than one plug-in.
Show view when tracing	Automatically brings up the Trace view if a trace is trying to be logged. By default, this setting is On.

Setting	Description
Trace buffer size	Increases the buffer to show more characters. As the buffer increases in size gets higher, performance might degrade, depending on your system.
Plug-Ins to Trace	Lists all Designer plug-ins (in their simple name form). Select plug-ins that you want to trace.
Select All	Enables tracing in all Designer plug-ins.
Deselect All	Disables tracing in all Designer plug-ins.

20.4.2 Identity Manager

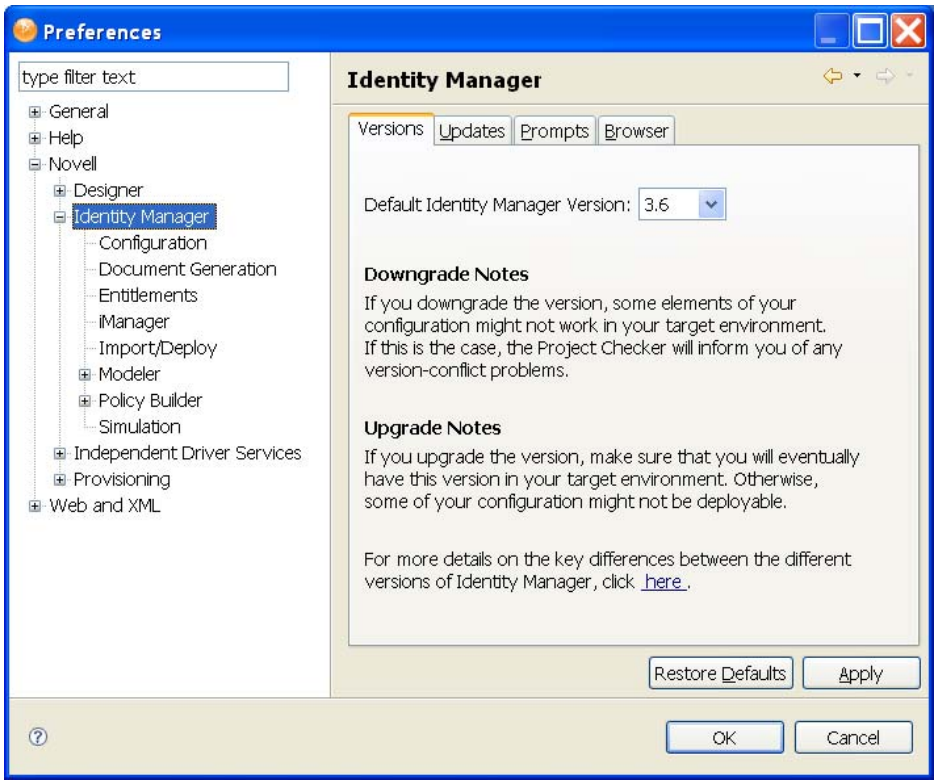
- "Identity Manager" on page 456
- "Configuration" on page 460
- "Document Generation" on page 465
- "Entitlements" on page 466
- "iManager" on page 467
- "Import/Deploy" on page 467
- "Modeler" on page 469
- "Policy Builder" on page 471
- "Simulation" on page 473

Identity Manager

- "Versions" on page 457
- "Updates" on page 458
- "Prompts" on page 459
- "Browser" on page 460

Versions

Figure 20-3 *Preferences: Novell > Identity Manager Versions tab*



Specifies the Identity Manager version running on a server.

Updates

Figure 20-4 *Preferences: Novell > Identity Manager > Updates*

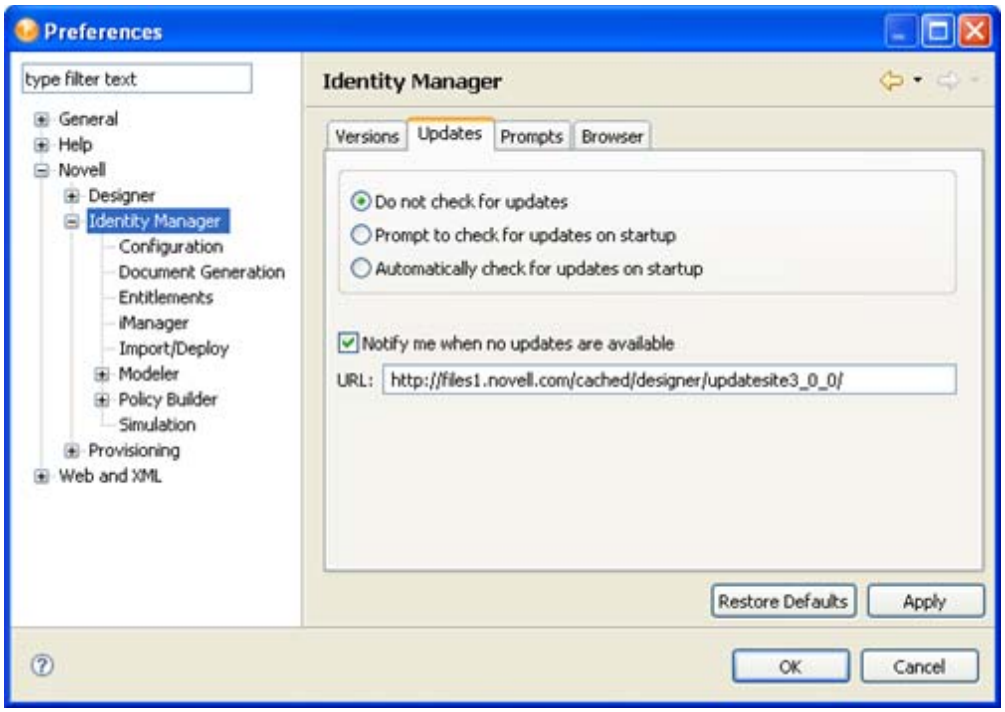


Table 20-17 *Preferences: Novell > Identity Manager > Updates*

Settings	Description
Do not check for updates	Prevents Designer from checking for updates on startup. Hides the Designer Updates dialog box.
Prompt to check for updates on startup	Displays a prompt each time you run Designer. You can disable this prompt.
Automatically check for updates on startup	Always checks for updates. If you disable the prompts that appear on startup, select this option.
Notify me when no updates are available	Displays a *No New Updates* message when you select to check for updates.

Prompts

Figure 20-5 *Preferences: Novell > Identity Manager > Prompts*

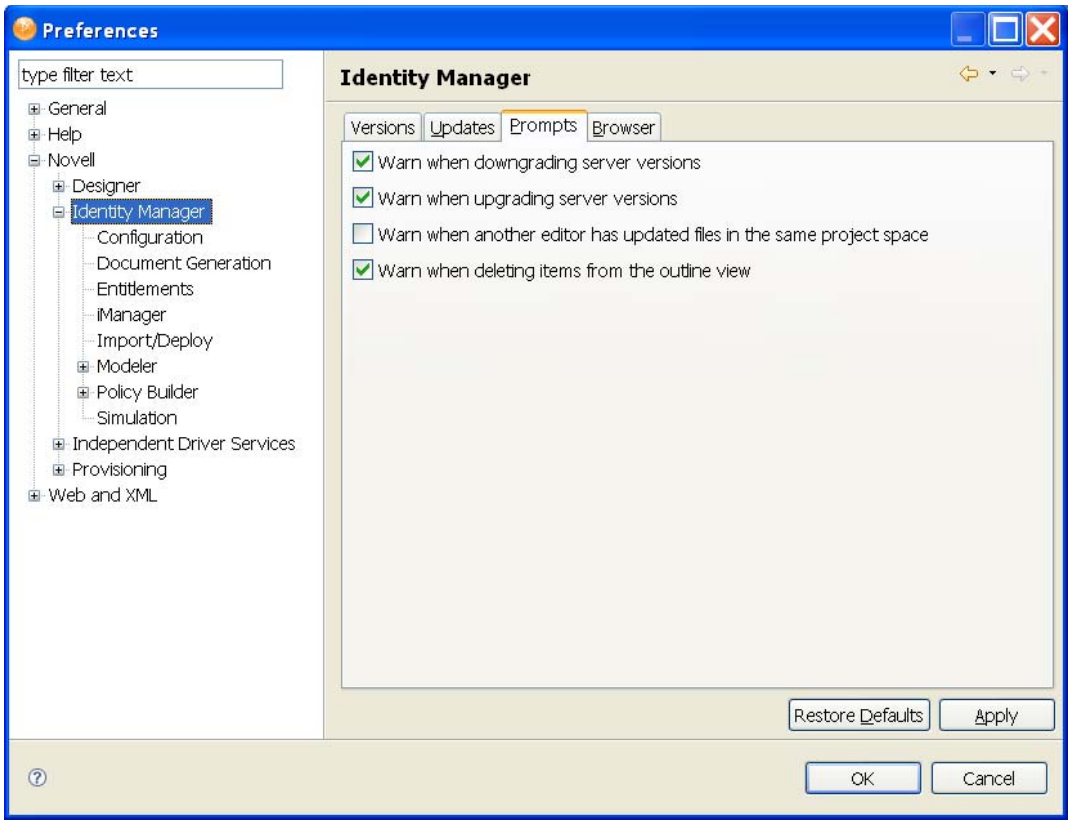
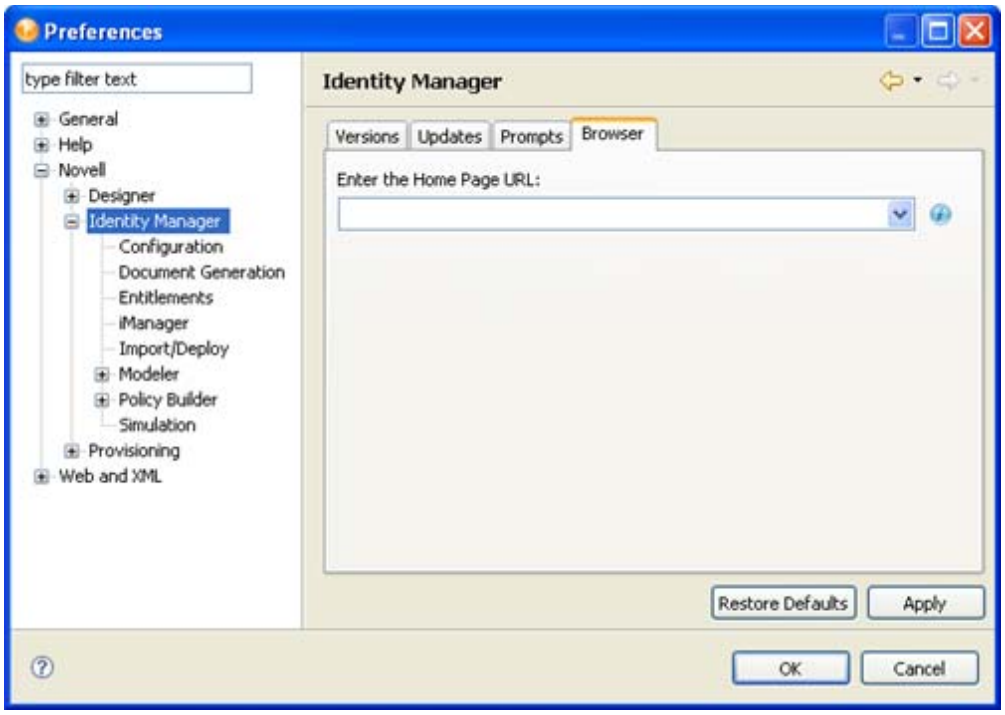


Table 20-18 *Preferences: Novell > Identity Manager > Prompts*

Setting	Description
Warn when downgrading server versions	Prompts you when you select an earlier server version for a project. If you downgrade, some elements of your configuration might not work in your target environment.
Warn when upgrading server versions	Prompts you when you select a later server version for a project. If you upgrade, some of your configuration might not be deployable unless you have this later server version in your environment.
Warn when another editor has updated files in the same project space	Warns you that your project might be erased from your workspace. The prompt occurs when overwriting a file in the file system for notification templates and policies.
Warn when deleting items from the outline view	Confirms that you want to delete the selected items.

Browser

Figure 20-6 *Preferences: Novell > Identity Manager > Browser*



You can use Designer to open a Web browser. After you enter the URL, Designer stores it. To change the URL, type a new one in *Preferences*, then click *OK*.

Configuration

- “General” on page 460
- “eDir-to-eDir SSL/TLS” on page 462
- “Prompts” on page 464

Each driver has a startup parameter. If it is disabled, the driver never starts until you change the setting. By default, Identity Manager drivers are disabled when you create them in the Modeler or start Designer. You must start them manually.

For more information, see Section 4.5, “Configuring Driver Sets,” on page 78.

General

These general settings specify how drivers start up and how their global configuration values (GCVs) act on specified target servers. The default state uses *Disabled* and *Merge GCVs*.

Figure 20-7 *Preferences: Novell > Identity Manager > Configuration General Tab*

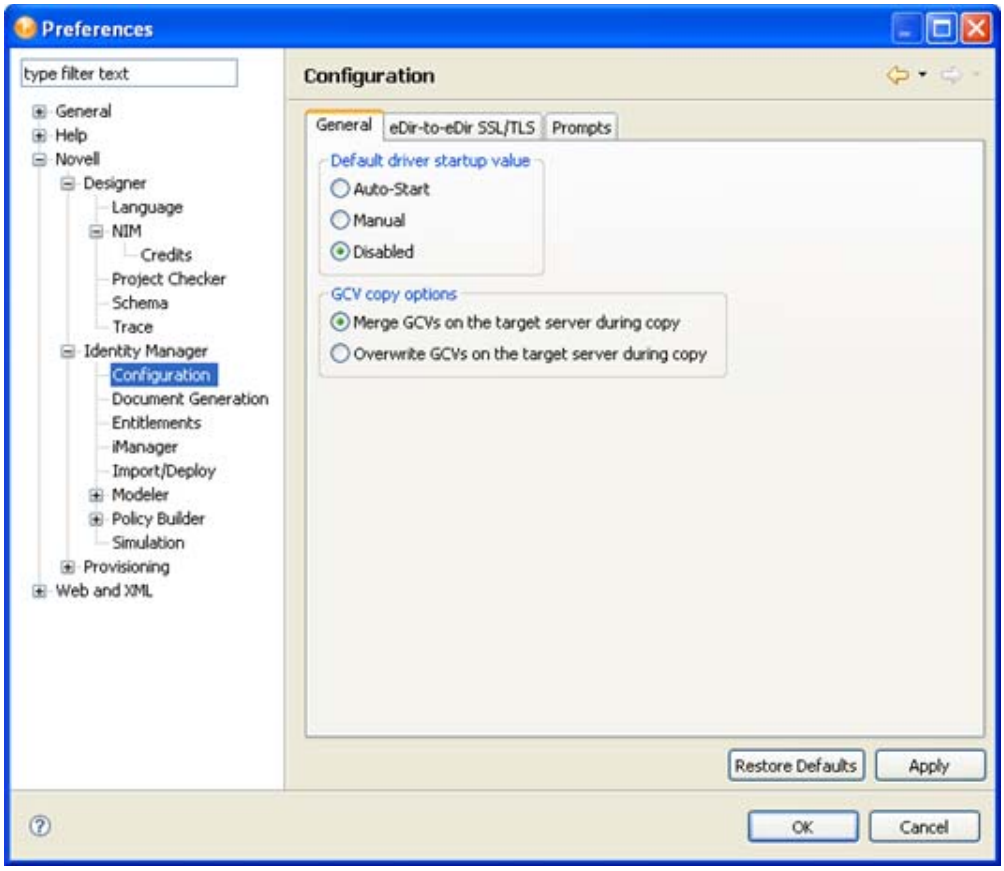


Table 20-19 *Preferences: Novell > Identity Manager > Configuration > General Tab Settings*

Setting	Description
Auto-Start	The driver automatically starts after you create it or whenever you start or load Designer.
Manual	You must start the driver manually.
Disabled	The driver never starts.
Merge GCVs on the target server during copy	Copies the GCVs from one driver/driver set to multiple targets of the same type. For example, you might configure GCVs on one driver and then copy them to multiple drivers. You also have the option of overwriting the target GCVs or merging your source GCVs with the existing target driver GCVs, if they exist.
Overwrite GCVs on the target server during copy	Overwrites existing GCVs when they are copied to the server.

eDir-to-eDir SSL/TLS

This setting configures how two eDirectory drivers communicate with each other over a secure channel. For more information, see Section 17.7, “Configuring TLS for eDir-to-eDir Drivers,” on page 392.

Figure 20-8 *Preferences: Novell > Identity Manager > Configuration eDir-to-eDir SSL/TLS Tab*

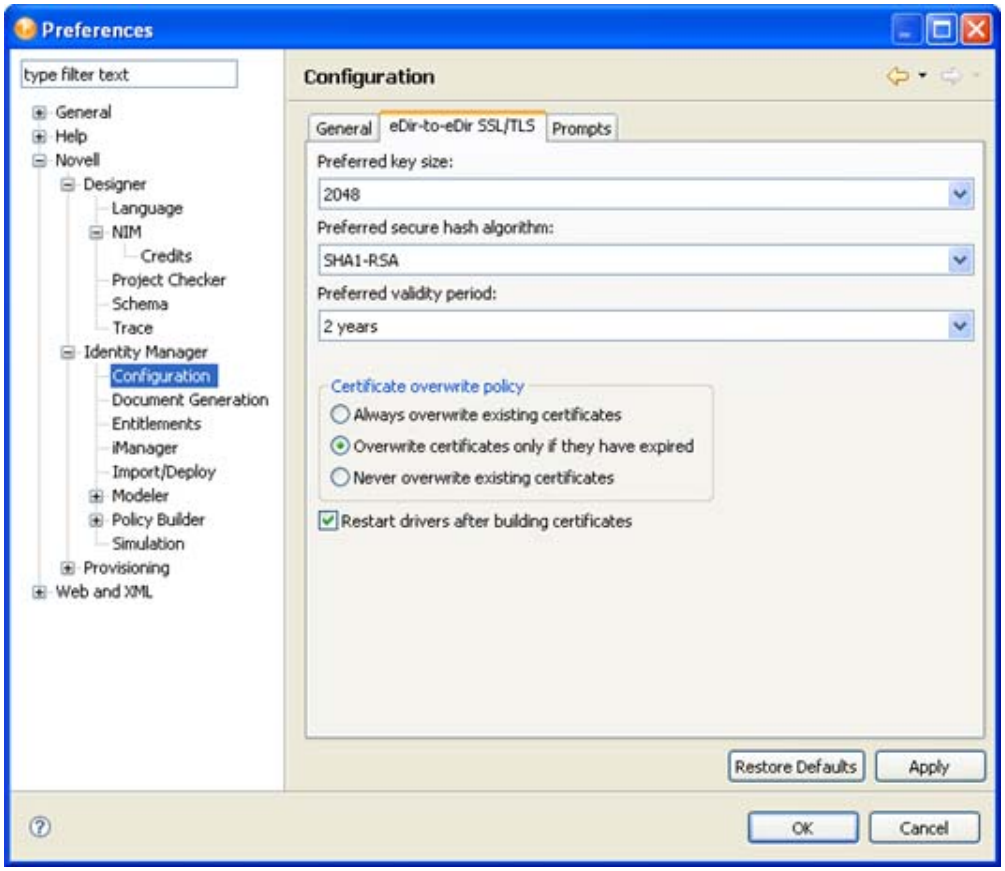


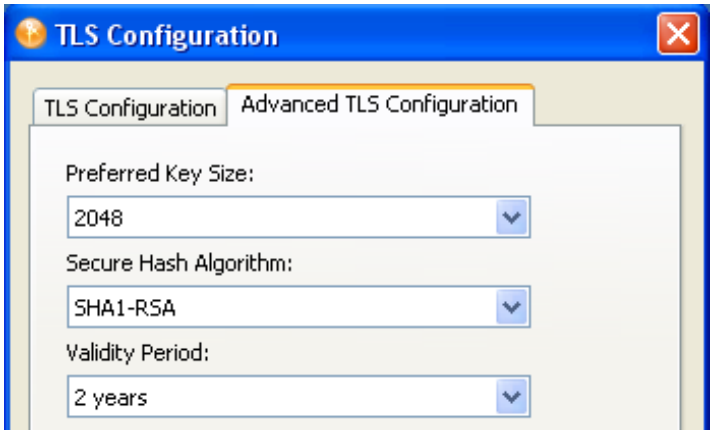
Table 20-20 *Preferences: Novell > Identity Manager > Configuration > eDir-to-eDir SSL/TLS Tab Settings*

Setting	Description
Preferred key size	Specifies the preferred key size that is generated when drivers are encrypted and stored in eDirectory: 512, 768, 1024, or 2048 bytes.
Preferred secure hash algorithm	Specifies the preferred hash algorithm to use when encrypting drivers: SHA1-RSA, MD2-RSA, or MD5-RSA.
Preferred validity period	Specifies the validity period for a driver certificate, ranging from 6 months to 10 years.

Setting	Description
Always overwrite existing certificates	Specifies that existing driver certificates are overwritten with each deployment. If you select this option, Designer deletes existing certificates and creates new ones. The new certificates are then good for another two years (assuming the default value is two years, as defined in the *Preferred Validity Period* field.) If you select *Live > Create eDir-to-eDir Certificates*, Designer deletes old certificates and creates new ones.
Overwrite certificates only if they have expired	Specifies that only expired driver certificates are overwritten with each deployment. This is the default setting. The default expiration length is two years. If a certificate expires, SSL/TLS stops working. If a certificate is expired, Designer deletes it and creates a new one.
Never overwrite existing certificates	Never overwrites driver certificates.
Restart drivers after building certificates	Restarts drivers after certificates have been updated or created.

When you create certificates, Designer reads the preferences, including Preferred Key Size, Preferred Secure Hash Algorithm, and Preferred Validity Period. These options are also available through *Secure Connection Settings > Advanced TLS Configuration*.

Figure 20-9 *The Advanced TLS Configuration Dialog Box*



NOTE: Designer reads these preferences after you first set them. If you subsequently change the preferences by using the driver's configuration page, those changes override the settings in Preferences.

After you change default settings and click *OK*, that configuration information is recorded. When you deploy the driver, Designer creates the certificates, or deletes and creates new certificates with a new time stamp.

Prompts

These settings specify how users are prompted to manage driver certificates on the target server. All are selected in the default state.

Figure 20-10 *Preferences: Novell > Identity Manager > Configuration Prompts Tab*

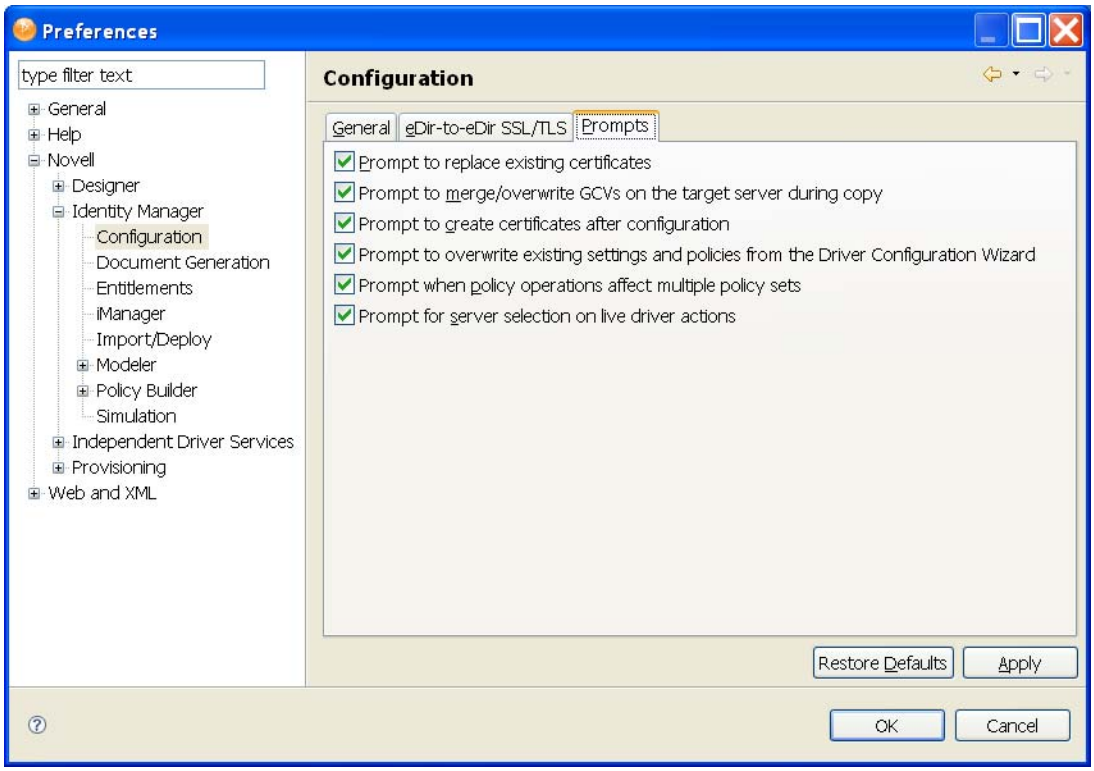


Table 20-21 *Preferences: Novell > Identity Manager > Configuration > Prompts Tab Settings*

Setting	Description
Prompt to replace existing certificates	Prompts the user to provide new certificates.
Prompt to merge/overwrite GCVs on target server during copy	Prompts the user to merge or overwrite when copying GCVs to the target server.
Prompt to create certificates after configuration	Prompts the user to create certificates after configuring a secure connection.
Prompt to overwrite existing settings and policies from the Driver Configuration Wizard	In the Driver Configuration Wizard, prompts the user whether to reset (overwrite) all driver settings and policies.
Prompt when policy operations affect multiple policy sets	Turns on and off a warning dialog box associated with policy operations. The dialog box appears when you move policies in a pre-3.5 environment and the move operation affects multiple policy sets.
Prompt for server selection on live driver actions	Any time you perform a live action on a driver (such as starting or stoping the driver) it prompts you to specify the server associated with the driver.

Document Generation

The Document Generator comes with the following settings:

Figure 20-11 *Preferences: Novell > Identity Manager > Document Generation Tab*

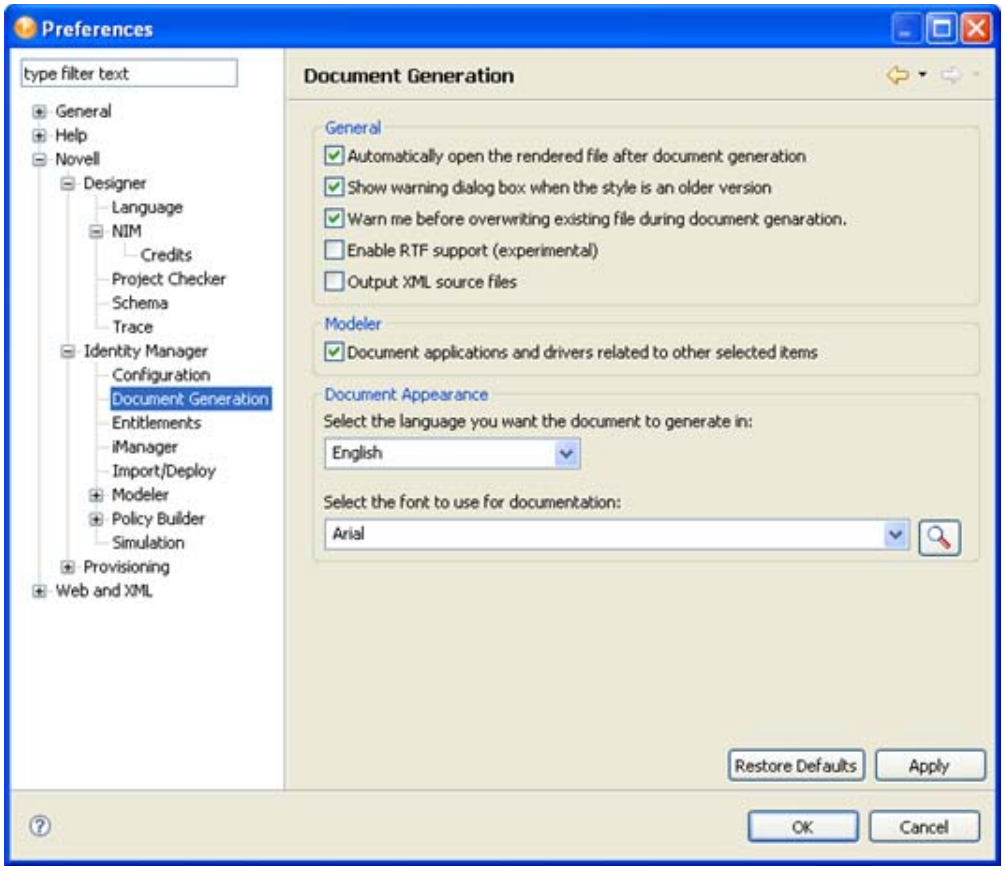


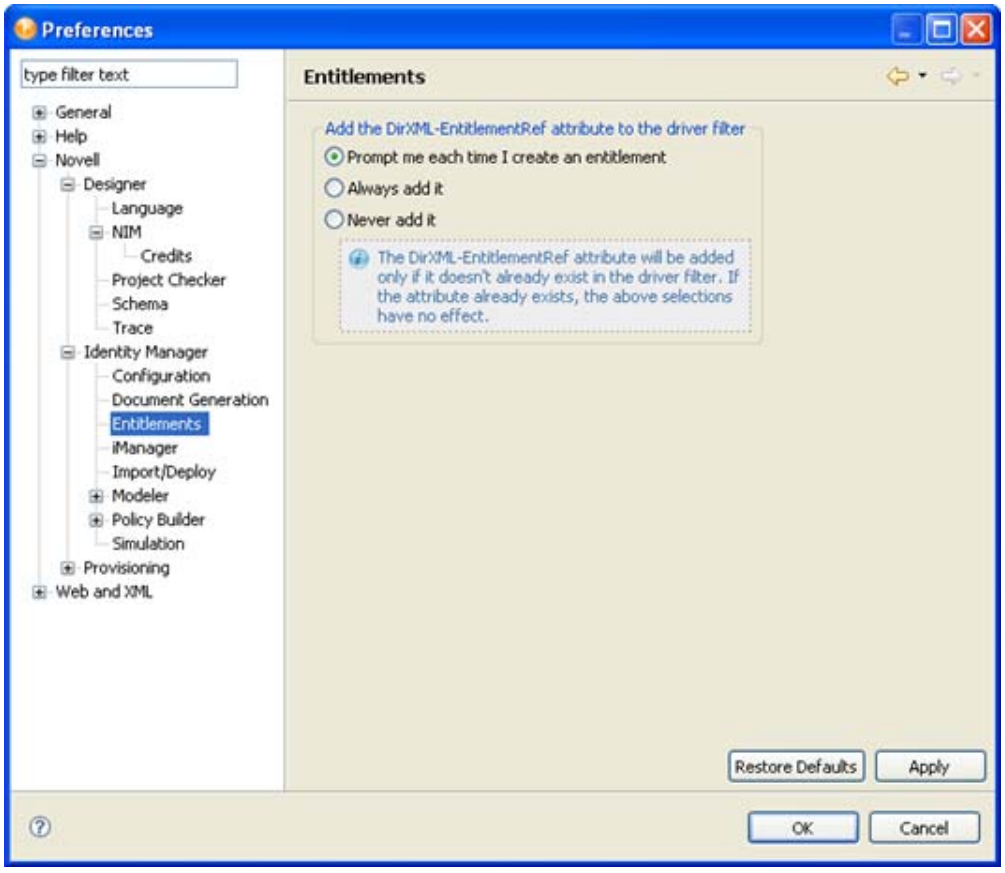
Table 20-22 *Preferences: Novell > Identity Manager > Document Generation Tab Settings*

Setting	Description
Automatically open the rendered file after document generation.	If you have a PDF reader installed on your workstation, the rendered file automatically opens in the reader. If you have enabled the RTF format and have a TRTF reader installed, the rendered file automatically opens in the reader. The default is On.
Show warning dialog box when the style is an older version.	Displays a warning when generating documents on out-of-version styles. The default is On.
Warn me before overwriting existing file during document generation	Displays a warning when overwriting previously generated files.
Enable RTF support (experimental).	Allows you to save documents to RTF format. This is a pre-release version. The default is Off.
Output XML source files	Generates XML files as part of the document generation process.

Setting	Description
Document applications and drivers related to other selected items.	With this option selected, parent objects and direct child objects are included to give context to the document. Deselecting this option excludes direct children of the selected item. The default is On.
Document Language	Allows you to select a language other then English in which to generate documents. Languages include Chinese Simplified, Chinese Traditional, Dutch, English, French, German, Italian, Japanese, Portuguese Brazil, and Spanish. The default is English.
Font settings	Allows you to select the font you want to use for document generation. This selection adds double-byte font support. The default is Arial font.

Entitlements

Figure 20-12 *Preferences: Novell > Identity Manager > Entitlements Tab*



Controls whether or how often you receive a prompt whenever you add the DirXML-EntitlementRef attribute to a driver filter. The default is *Prompt me*, but because this attribute is added only if it doesn't already exist on the driver filter, you can select *Always add it* to not see the pop-up window.

iManager

The iManager preferences page includes the following settings:

Setting	Description
iManager URL	The IP address and port for the iManager server.
Show Novell iManager Information Dialog	Prompts you for the URL to the iManager server after you select *Tools* > *iManager*. If the URL is missing or incorrect, iManager is unable to launch.

Import/Deploy

The Import/Deploy preferences window contains three tabs: *Behaviors*, *Prompts*, and *Trace*. The following tables describe their options.

Table 20-23 *Preferences: Novell > Identity Manager > Behaviors (Import Settings)*

Setting	Description
Perform prompt checking when running a driver configuration file	Displays the *Do you wish to perform all mandatory and required prompt checking when running this Driver configuration file?* prompt. If you select *Yes* to the prompt, you must then enter information in required fields while configuring the driver. If you select *No*, you temporarily disable this setting and can skip required fields.
Include application schema when importing drivers	Imports the eDirectory application schema when you select this option. You might not want to import all the associated data. The default is Off. See Section 11.4.3, “Importing a Schema,” on page 256.

Table 20-24 *Preferences: Novell > Identity Manager > Behaviors (Deploy Settings)*

Setting	Description
Replace driver set/server associations when deploying a driver set	If you want driver set and server associations when deploying, select this option. The default is Off.
Always deploy both drivers of an eDir-to-eDir connection	With this option selected, you are prompted to deploy both sides on the connection. With both drivers deployed, Deploy is integrated with the creation of eDir-to-eDir certificates, if the certificates are created in Designer. Deploy adheres to the settings set in *Preferences* > *Designer for IDM* > *Configuration* > *eDir-to-eDir.* The default is On. This is the recommended setting.
Restart running drivers after deploying the driver	Restarts the driver after it is deployed. The default is On.

Table 20-25 *Preferences: Novell > Identity Manager > Behaviors (Summary Dialog)*

Setting	Description
Show the summary dialog prior to performing an import	Allows you to view what's being imported in a summary screen. The default is On.
Show the summary dialog prior to performing a deployment	Allows you to view what's being deployed in a summary screen. The default is On.
Filter passwords out of summary and compare dialogs	Check this box if you want to filter passwords out of summary and compare dialogs.

Table 20-26 *Preferences: Novell > Identity Manager > Behaviors (Export Settings)*

Setting	Description
Copy cross driver policy references into exported configuration files	Selected by default, this option saves you the trouble of manually inputting cross-driver policy references.

Table 20-27 *Preferences: Novell > Identity Manager > Prompts Tab Settings*

Setting	Description
Show dialog to export cross driver policy references to configuration files	Selected by default. If you do not want to see a dialog box about these references, deselect the option.
Show a warning dialog when overwriting a driver set/server association	Warns that the driver set being deployed has a different server association than the server that you are about to deploy to. The association in the deployed driver set overwrites the existing server association.
Show the dialog box to deploy both drivers of an eDir-to-eDir connection	This is the default, and it is also the recommended setting. With this option selected, you are prompted to deploy both sides of the connection.
Show the dialog box to restart drivers after a deployment	Selected by default. If you do not want to see a dialog box about these references, deselect the option.

Table 20-28 *Preferences: Novell > Identity Manager > Trace Tab Settings*

Setting	Description
Trace import and deploy event information	Deselected by default. If you need to troubleshoot an import or a deploy, select this option, then open the Trace view to inspect the import or deploy.
Generate debug messages for the Driver configuration prompt dialog box	Deselected by default. If you need to generate debug messages, select this option.

Setting	Description
Show verbose debug messages	Deselected by default. If you need to generate verbose debug messages, select this option.
Time import and deploy operations	Deselected by default. If you need to time how long it takes to import or deploy an object, select this option.

Modeler

The Modeler preferences window contains seven tabs: *Behaviors*, *Display*, *Guidance*, *Layouts*, *Pages*, *Prompts*, and *Themes*. The following tables describe their options.

Additionally, the following preferences categories appear as Modeler sub-pages:

- “Dataflow Page” on page 471
- “Palette Page” on page 471

Table 20-29 *Preferences: Novell > Identity Manager > Modeler > Behaviors Tab Settings*

Setting	Description
Auto-create servers when connecting a driver to a different driver set	Automatically creates a server for a driver set when you connect a driver to a different driver set.
Launch the driver Properties dialog box	Launches the driver’s Properties page.
Show the driver’s Policy Flow view	Displays the driver’s Policy Flow diagram in the Outline view.

Table 20-30 *Preferences: Novell > Identity Manager > Modeler > Display Tab Settings*

Setting	Description
Show labels by Applications and Identity Vaults (Arhitect. mode)	Shows labels below applications (in both modes) and above Identity Vaults (Architect mode only).
Show driver icons in Developer mode	Displays a driver icon on the line that represents a driver in the Modeler.
Show password icons in Developer mode	Displays a password sync icon below a driver icon in the Modeler.
Auto-expand Identity Vaults to fit contents	Causes Identity Vaults to expand to accommodate objects that you place in them.
Auto-shrink Identity Vaults to fit contents	Causes Identity Vaults to shrink when you remove objects from them.
Auto-size Identity Vaults to fit their titles	Enables vaults to expand horizontally, to accommodate long titles. Otherwise, the titles concatenate after approximately 20 characters.

Setting	Description
Grid Width	Increases or decreases cells in the Modeler's grid. To access the grid, select the Modeler, then click *View > Grid*.

Table 20-31 *Preferences: Novell > Identity Manager > Modeler > Guidance Tab Settings*

Setting	Description
If an Identity Vault doesn't already exist, one will be created when you drop the application	Creates an Identity Vault when you drag or drop an application from the palette into the Modeler.
eDir-to-eDir connection tip, when you've connected the same eDir app to two driver sets	Prompts you to connect a line directly between the end driver sets when you set up an eDir-to-eDir relationship.
Setting dataflows in architect mode will default all policy and schema settings	Sets policy and schema settings to defaults when you set data flows in architect mode. To edit the settings, use the Developer mode.
Saving Dataflow to disk will first force a project save	Requires you to save a project before you can save a dataflow to disk.

Table 20-32 *Preferences: Novell > Identity Manager > Modeler > Layouts Tab Settings*

Setting	Description
Default Layout for Applications on Import	Specifies the default layout for application objects when you import a project into Designer.

To arrange an existing project in a particular layout:

1. In the Modeler, right-click a driver set.
2. Select *Arrange Applications*.
3. Select a layout.

Table 20-33 *Preferences: Novell > Identity Manager > Modeler > Pages Tab Settings*

Setting	Description
Check the additional Modeler pages you want visible	Determines whether the Architect, Dataflow, and Table pages display as tabs at the bottom of the Modeler. The Developer mode is always enabled.

Table 20-34 *Preferences: Novell > Identity Manager > Modeler > Prompts Tab Settings*

Setting	Description
Show the Driver Config Wizard at connection time	Launches the Driver Configuration Wizard when you drag or drop an application in the Modeler.

Setting	Description
Confirm when a driver is being deleted	Provides a Yes/No prompt for you to choose whether you want to delete the driver and its policies.

Table 20-35 *Preferences: Novell > Identity Manager > Modeler > Themes Tab Settings*

Setting	Description
Developer	Specifies the theme for Developer mode. Themes define the colors used for background, text, line, domain group background, and domain group title in the Modeler.
Architect	Specifies the theme for Architect mode. Themes define the colors used for background, text, line, domain group background, and domain group title in the Modeler.

Dataflow Page

Specifies the number of columns per page that the Dataflow editor saves in the HTML reports.

To view or use the Dataflow editor, select the *Dataflow* tab in the Modeler.

Figure 20-13 *The Dataflow Tab*

Palette Page

The Palette tab includes the following settings:

Setting	Description
Arrange applications in folders	Displays folders (for example, Database) in the palette and places applications in appropriate folders.
Arrange applications in an alphabetical list	Places all applications into one folder in the palette, and lists the applications alphabetically.

Policy Builder

The Policy Builder preferences page includes the following settings:

Additionally, the following preferences categories appear as Policy Builder sub-pages:

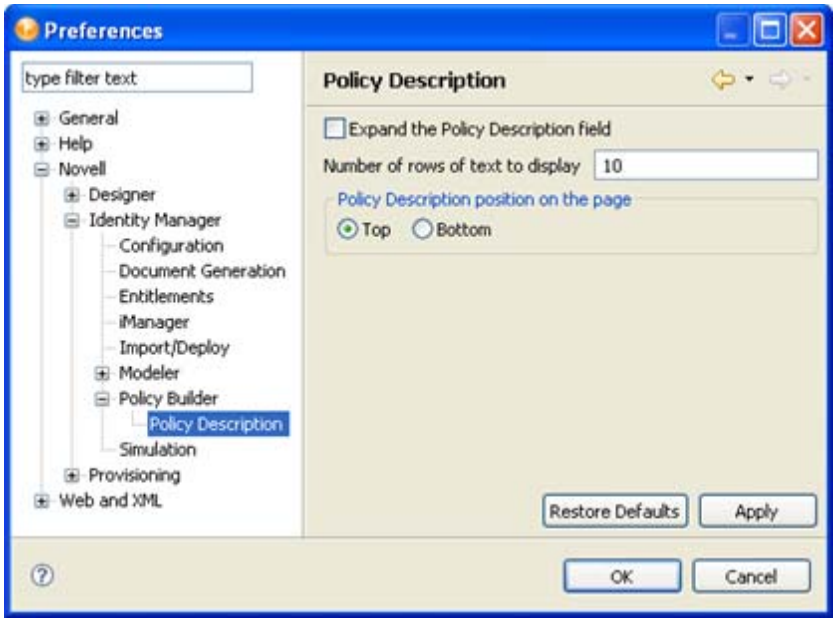
- "Policy Description" on page 472

Setting	Description
Localize actions, conditions and tokens	Translates the names of policy actions, conditions and tokens into the selected Designer language. When not selected, policy actions, conditions and tokens display in English.
Include project name in title	Includes project name in the title.
Expand all rules when the Policy Builder is loaded	Automatically expands rules in the Rules pane when you open the Policy Builder.
Show version/author/last changed information	Adds additional fields in the Rule Inline editor (available when you double-click a rule.) Designer adds the information from these fields to the policy.

Policy Description

The Policy Description allows you to add comments about the policy.

Figure 20-14 *Preferences: Novell > Identity Manager > Policy Builder > Policy Description*



The Policy Description preferences page includes the following settings:

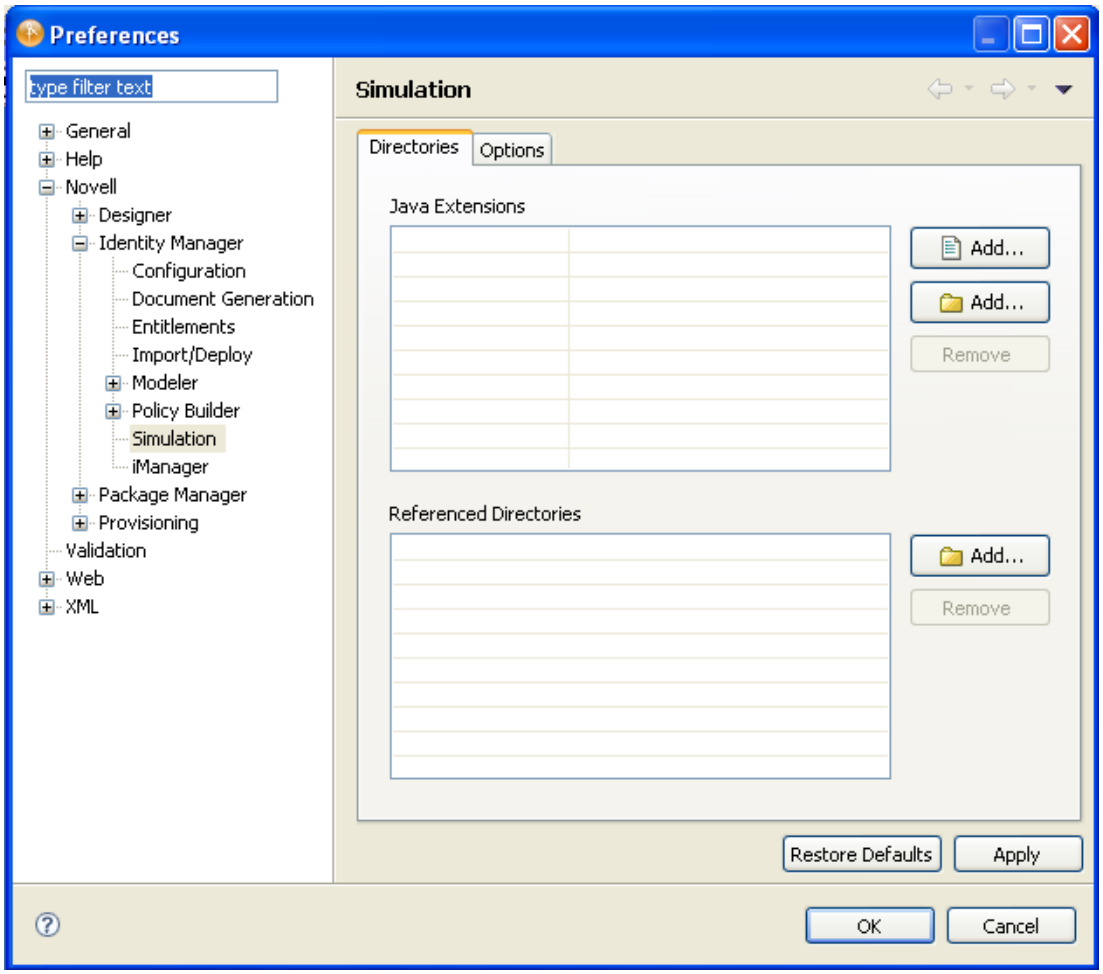
Setting	Description
Expand the Policy Description field	Automatically expands the *Policy Description* field. You can hide the field by selecting the check box.
Number of rows of text to display	Determines how many rows to display in the *Policy Description* field. The default is 10.

Setting	Description
Policy Description position on the page	Places the *Policy Description* field above or below the Rules pane.

Simulation

The Simulation preferences page allows you to set the preferences for simulating policies and/or drivers.

Figure 20-15 *Preferences: Novell > Identity Manager > Simulation*



The Simulation preferences page includes the following settings:

Setting	Description
Directories: Java Extensions	Enables you to simulate policies that contain references to external Java extensions. Specify the `.jar` file or the directory where the `.jar` file is located to add it to the class path. You can specify multiple Java extensions.

Setting	Description
Directories: Referenced Directories	Enables you to choose directories referenced by the external jars added to the Simulator's classpath in the *Directories: Java Extensions* option. By default, it is the current working directory. Click *Add* to add the selected directory to the Simulator's classpath. You can specify a different directory referenced by the jars added in the *Directories: Java Extensions* option. While simulating a policy or driver, configuation or referenced files present in the directory will be available at runtime.
Options: Clear the policy simulation log file prior to performing a simulation	Automatically clears the log file. If you don't enable this setting, Designer displays a *Clear Log* button that you can use. If you do many simulations in succession, you might want to disable this option. The log file then captures and displays the events of all the tests, until you click *Clear Log*.
Options: Show the information prompt when a query is generated	Displays a prompt when the Simulator generates a query. It simulates what the engine would do when a query is required to process the policy.
Options: Notify user when converting the Input Document schema	Notifies a user when the Policy Simulator must convert the Application schema to the ID Vault schema, or vice versa. This is typically necessary when changing the input document's simulation point.

20.4.3 Provisioning

The Provisioning preferences page configures provisioning settings for the Identity Manager User Application. For information on provisioning settings, see Setting Provisioning View Preferences (http://www.novell.com/documentation/idm40/dgpro/index.html?page=/documentation/idm40/dgpro/data/palprovprefs.html) in the *User Application: Design Guide*.

20.5 Web and XML

The Web and XML preferences let you specify how Designer should handle editing and creation of CSS, HTML, and JavaScript* files.

- Section 20.5.1, "CSS Files," on page 475
- Section 20.5.2, "HTML Files," on page 478
- Section 20.5.3, "JavaScript Files," on page 482
- Section 20.5.4, "Task Tags," on page 486
- Section 20.5.5, "XML Catalog," on page 486
- Section 20.5.6, "XML Files," on page 488

20.5.1 CSS Files

The CSS Files preferences allow you to specify how Designer displays and manages CSS files.

Figure 20-16 *Preferences: Web and XML > CSS Files*

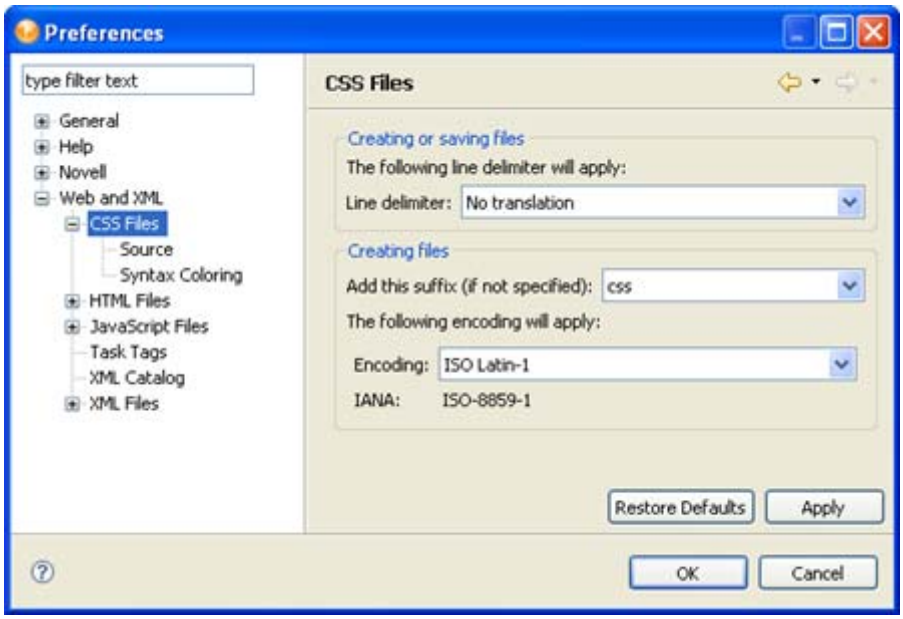


Table 20-36 *Preferences: Web and XML > CSS Files Settings*

Preference	Description
Creating or saving files: Line Delimiter	Sets the type of line delimiter applied to newly created files. The options are: ◆ *Windows* ◆ *Unix* ◆ *Mac* ◆ *No translation*
Creating files: Add this suffix	The file suffix the editor adds when creating files. The default is CSS.
Creating files: Encoding	The encoding used for new files.

- "CSS Source" on page 476
- "Syntax Coloring" on page 477

CSS Source

Figure 20-17 *Preferences: Web and XML > CSS Files > Source*

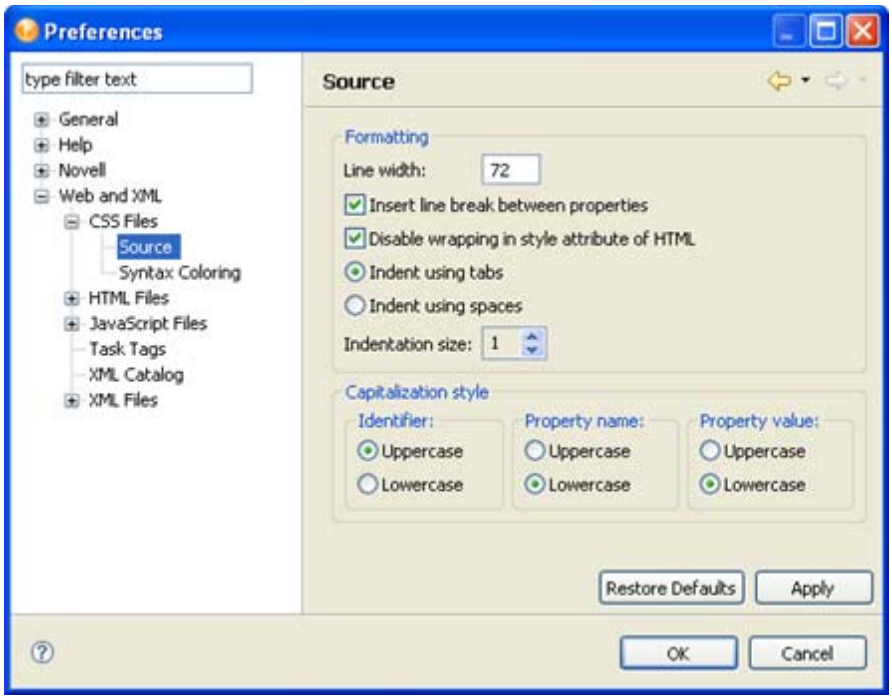


Table 20-37 *Preferences: Web and XML > CSS Files > Source Settings*

Setting	Description
Formatting: Line width	Specifies the number of characters in a line.
Formatting: Insert line break between properties	Specifies whether the editor should insert a line between the CSS properties.
Formatting: Disable wrapping in style attribute of HTML	Specifies whether the HTML editor (used in the e-mail notification template editor) should allow wrapping of the value of a style attribute.
Formatting: Indent using tabs or spaces	Specifies how the first line of text indents.
Formatting: Indentation size	Specifies the size of the indent.
Formatting: Capitalization style	Specifies the default case for identifiers, property names, and property values.

Syntax Coloring

Figure 20-18 *Preferences: Web and XML > CSS Files > Syntax Coloring*

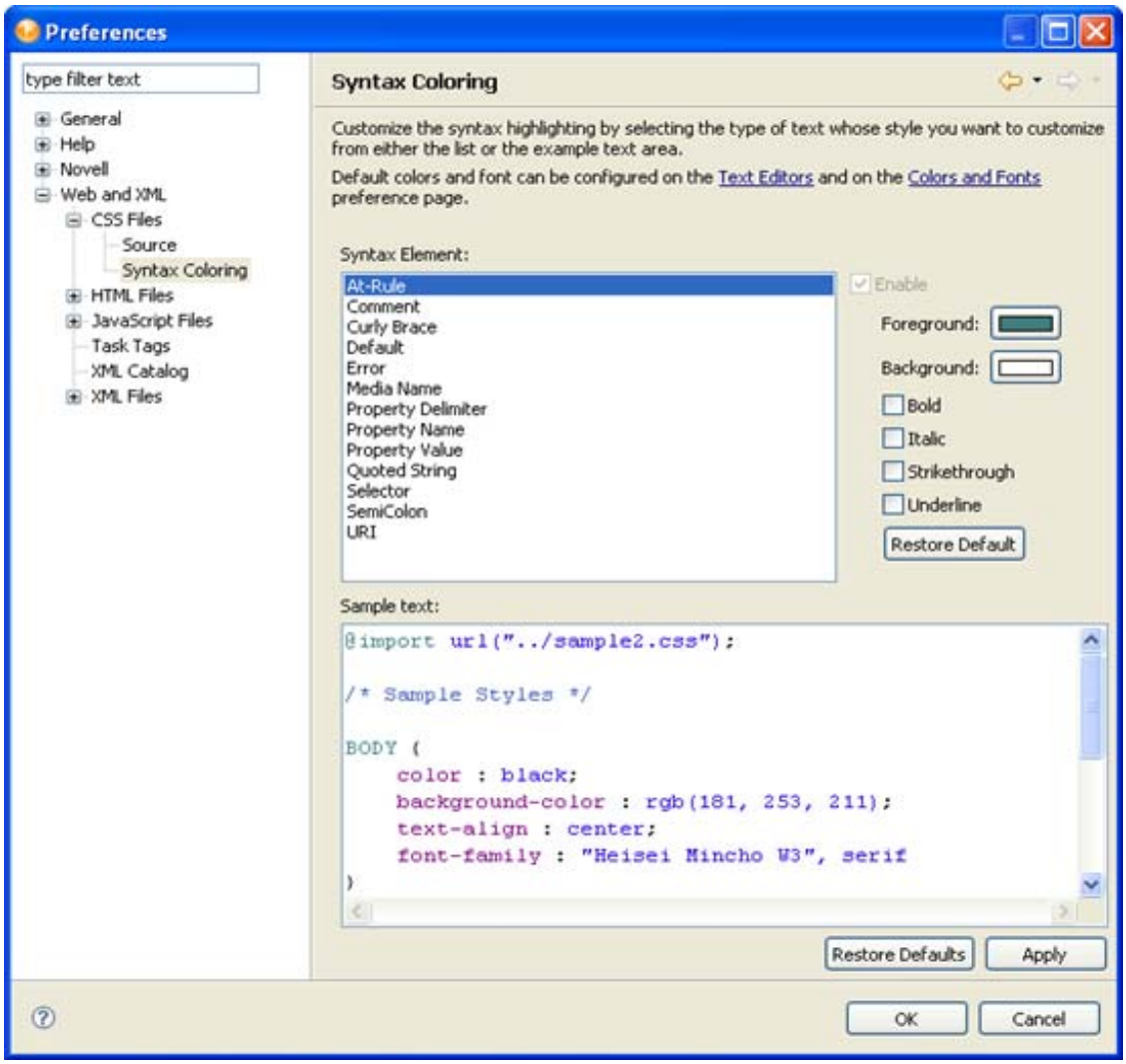


Table 20-38 *Preferences: Web and XML > CSS Files > Syntax Coloring Settings*

Setting	Description
Syntax Element	Choose the content type for which you want to define a style.
Foreground/Background/Bold/Italic/Strikethrough/ Underline	Specifies the syntax highlighting and formatting for individual CSS elements.
Sample Text	Displays sample CSS with the selected syntax coloring options.

20.5.2 HTML Files

The HTML Files preferences allow you to specify how Designer displays and manages HTML files and content.

Figure 20-19 *Preferences: Web and XML > HTML Files*

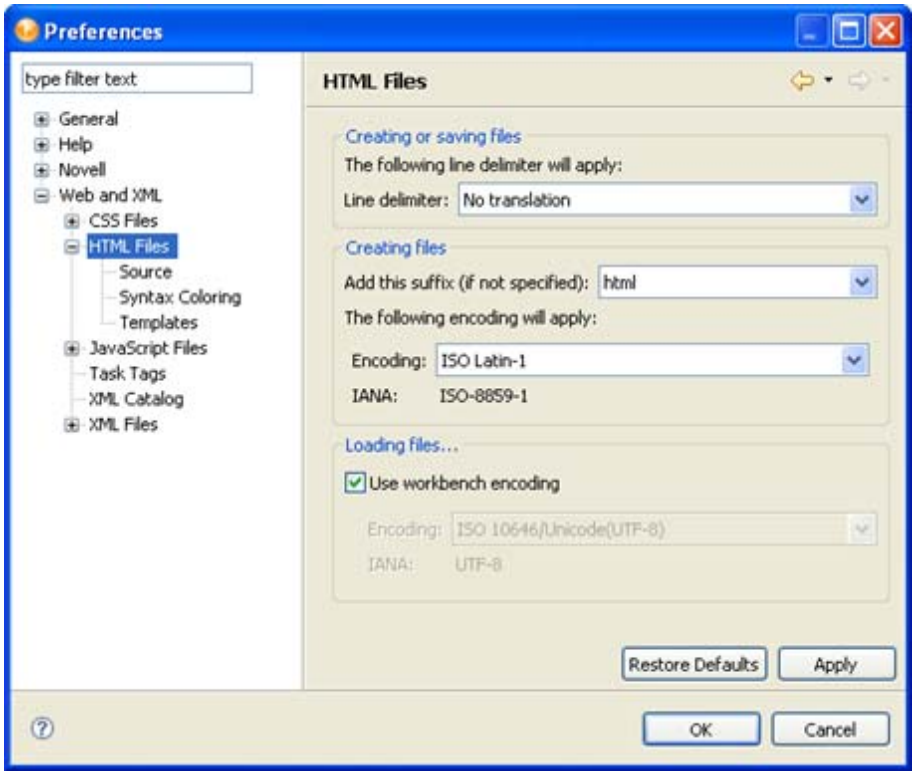


Table 20-39 *Preferences: Web and XML > HTML Files Settings*

Preference	Description
Creating or saving files: *Line Delimiter*	Choices are: ◆ *Windows* ◆ *Unix* ◆ *Mac* ◆ *No translation*
Creating files: Add this suffix	Specifies the file suffix the editor should add when creating a new file. The default is `html`.
Creating files: Encoding	Specifies the editor's encoding for new files.
Loading files	Choose the encoding for files opened in the editor. Click *Use workbench encoding* to accept the default UTF-8, or select one from the list.

- "HTML Source" on page 480
- "HTML Styles" on page 481
- "HTML Templates" on page 482

HTML Source

Figure 20-20 *Preferences: Web and XML > HTML Files > Source*

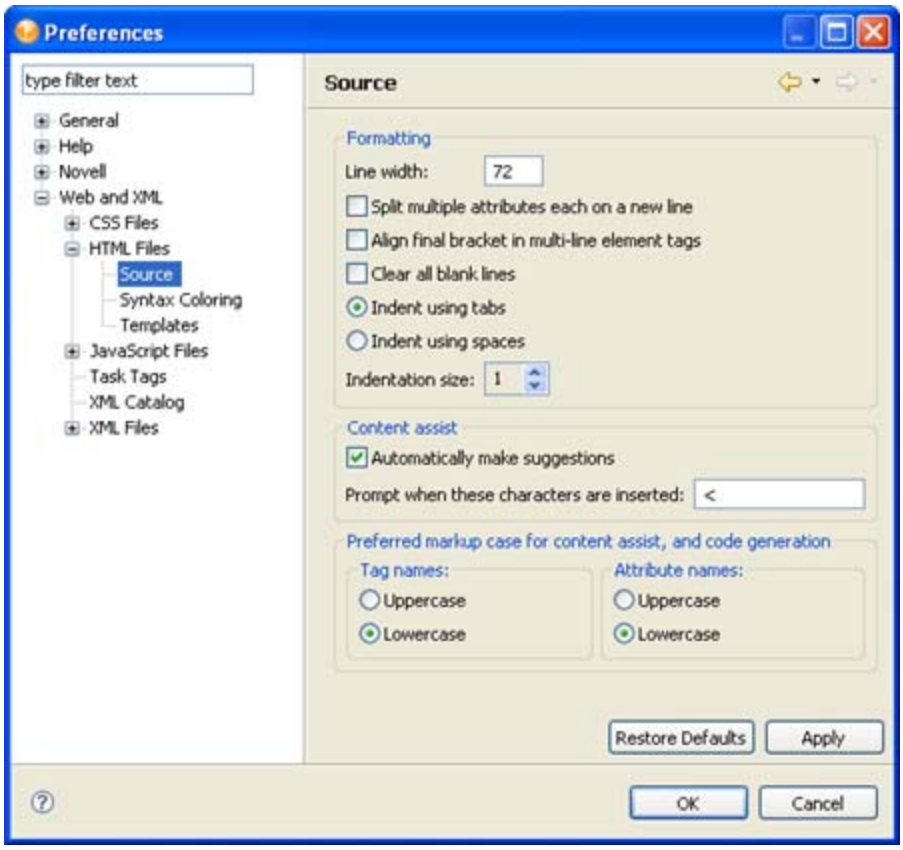


Table 20-40 *Preferences: Web and XML > HTML Files > Source Settings*

Setting	Description
Formatting: Line Width	Specifies the number of characters for each line.
Formatting: Split multiple attributes each on a new line	Specifies what the editor should do with multiple attributes.
Formatting: Align final bracket in multi-line element tags	Specifies what the editor should do with final brackets
Formatting: Clear all blank lines	Specifies what the editor should do with blank lines
Formatting: Indent using tabs or spaces Indentation size	Specifies whether the indent should be using tabs or spaces and also the indentation size.
Content assist: Automatically make suggestions	Specifies whether to do automatic code completion.
Content assist: Prompt when these characters are inserted	Specifies the characters that initiate the content assist.

Setting	Description
Preferred markup: Tag Names/Attribute Names	Specifies if the editor's suggestions should be in uppercase or lowercase.

HTML Styles

Figure 20-21 *Preferences: Web and XML > HTML Files > Syntax Coloring*

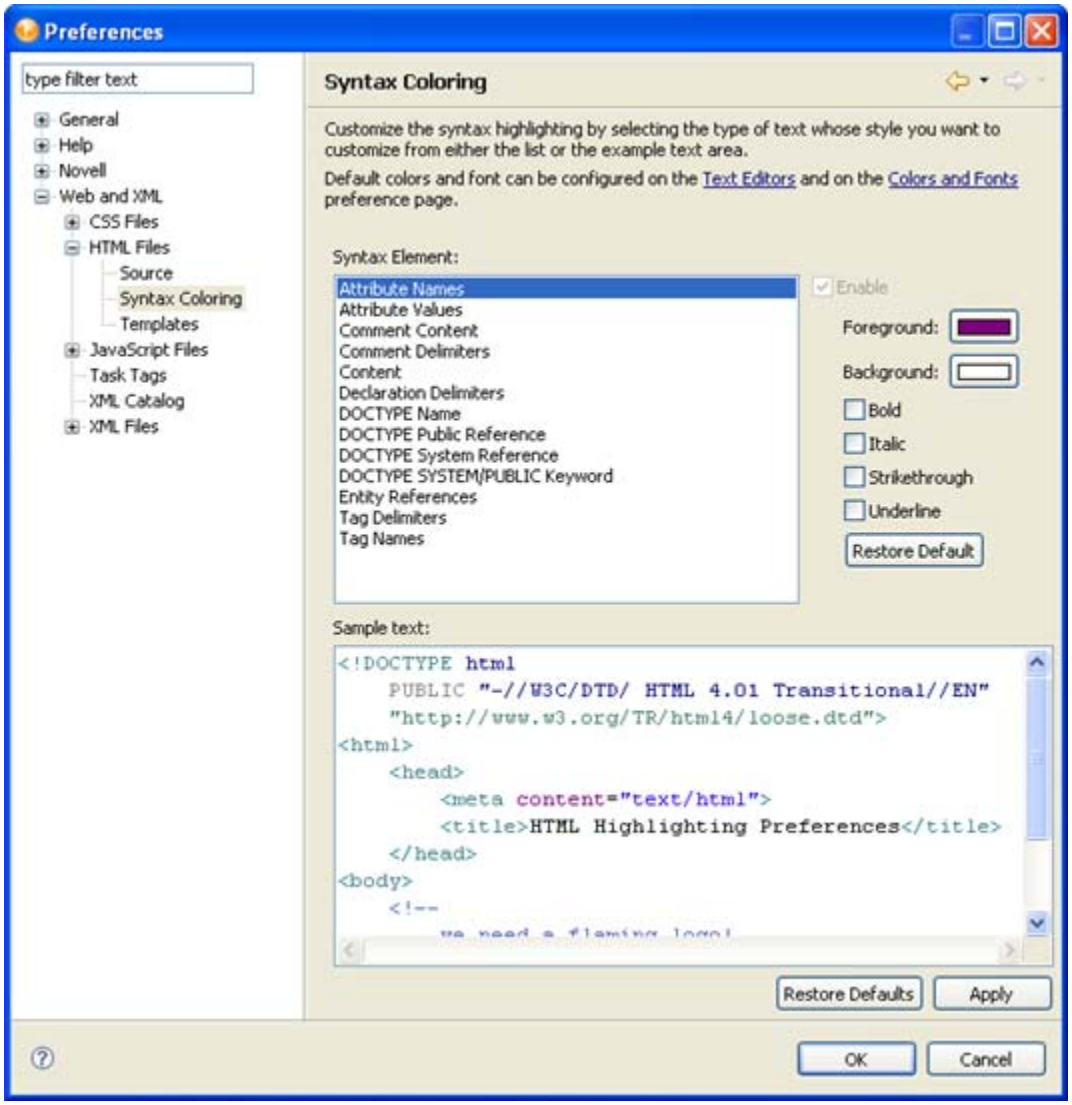


Table 20-41 *Preferences: Web and XML > HTML Files > Syntax Coloring Settings*

Setting	Description
Syntax Element	Choose the content type for which you want to define a style.
Foreground/Background/Bold/Italic/Strikethrough/ Underline	Specifies the syntax highlighting and formatting for individual CSS elements.

Setting	Description
Sample Text	Displays sample CSS with the selected syntax coloring options.

HTML Templates

Figure 20-22 *Preferences: Web and XML > HTML Files > Templates*

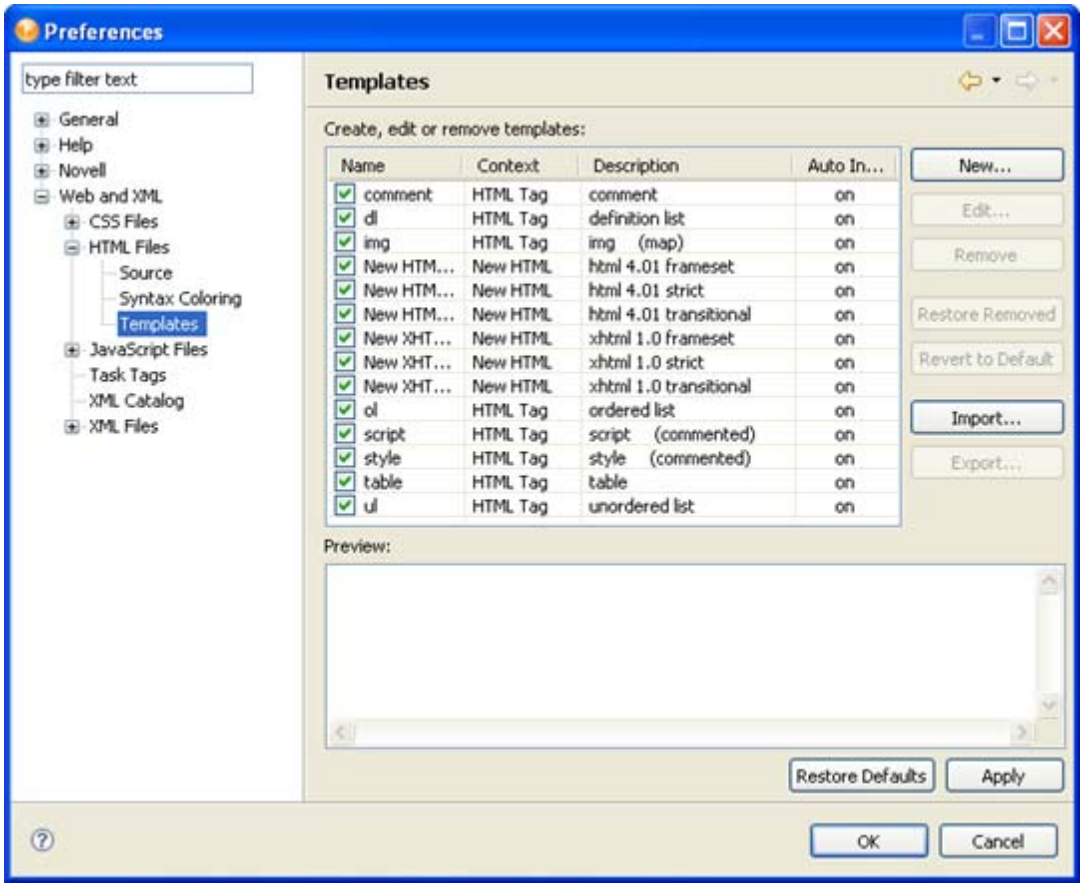


Table 20-42 *Preferences: Web and XML > HTML Files > Templates Settings*

Setting	Description
Templates	The templates are used in the code completion in source editor. Use this preference to add, remove or edit templates.

20.5.3 JavaScript Files

Changes how JavaScript files are created.

Figure 20-23 *Preferences: Web and XML > JavaScript Files*

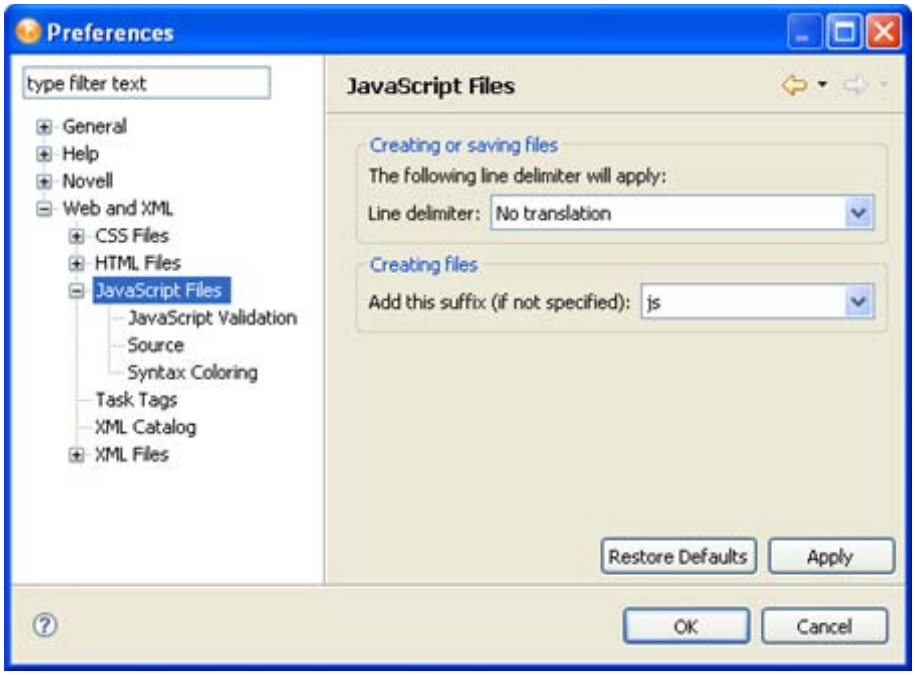


Table 20-43 *Preferences: Web and XML > JavaScript Files Settings*

Setting	Description
Creating or saving files: Line delimiter	Sets what type of line delimiter is applied to the file. The options are: ◆ *No translation* ◆ *UNIX* ◆ *Mac* ◆ *Windows*
Creating files: Add this suffix (if not specified)	Sets the suffix for the file. The default value is `js`. It can be set to any value.

- "JavaScript Validation" on page 483
- "Source" on page 484
- "Syntax Coloring" on page 485

JavaScript Validation

Allows the editor to validate the JavaScript as it is entered. Select *Automatically validate scripts* to automatically validate the scripts. If it is not selected, the JavaScript is not validated.

Figure 20-24 *Preferences: Web and XML > JavaScript Files > JavaScript Validation*

Source

Changes the formatting for the source JavaScript files.

Figure 20-25 *Preferences: Web and XML > JavaScript Files > Source*

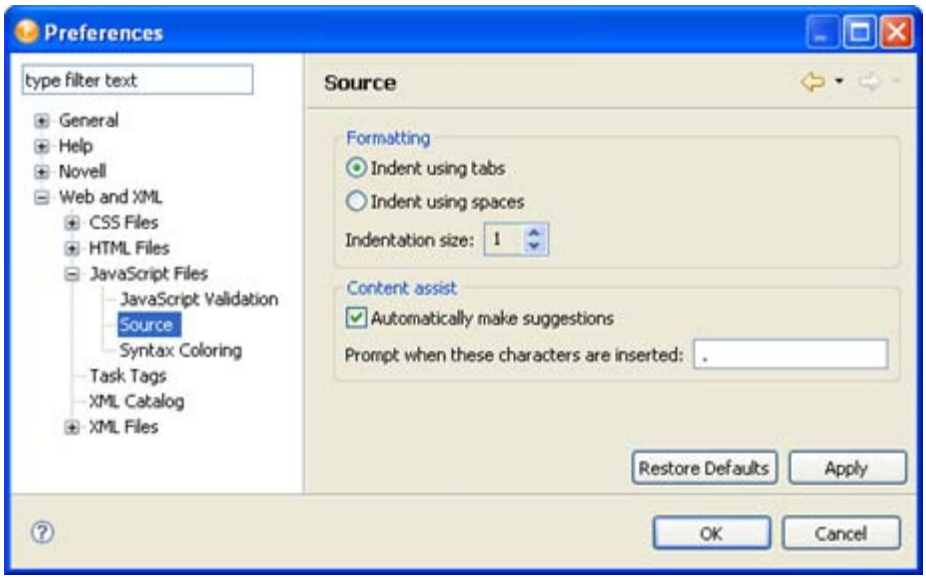


Table 20-44 *Preferences: Web and XML > JavaScript Files > Source Settings*

Setting	Description
Formatting	Sets the formatting for the editor. The options are: ◆ *Indent using tabs* ◆ *Indent using spaces* ◆ *Indentation size*: Changes the indentation size by setting a numeric value.
Content assist	Helps with prompts when creating the files. *Automatically make suggestions*: Can be enabled or disabled if the check box is selected. *Prompt when these characters are inserted*: Allows you to receive prompts when the specified characters are entered.

Syntax Coloring

Customizes the syntax highlighing in the ECMAScript editor. The default colors and fonts can be configured on the Text editors and on the Colors and Fonts preference page.

Figure 20-26 *Preferences: Web and XML > JavaScript Files > Syntax Coloring*

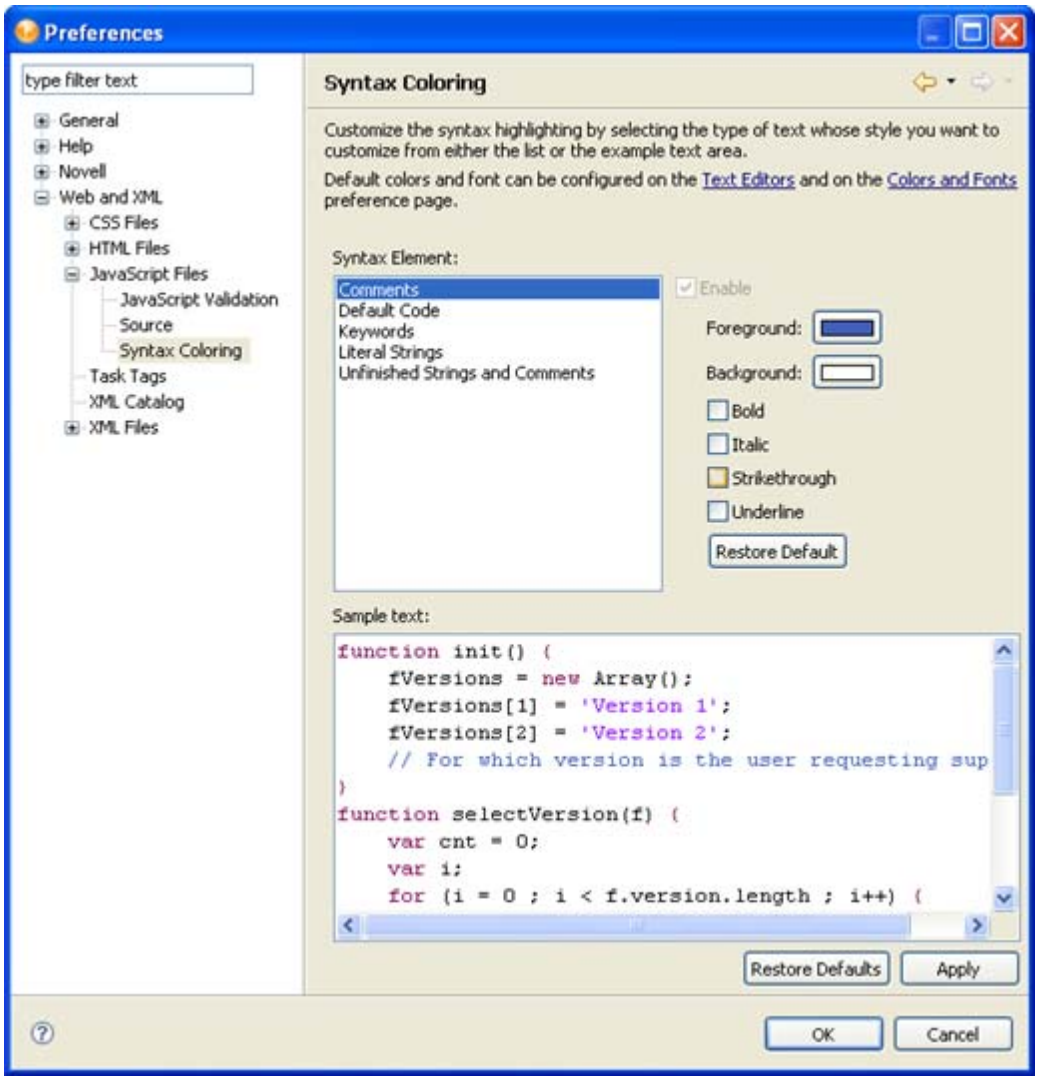
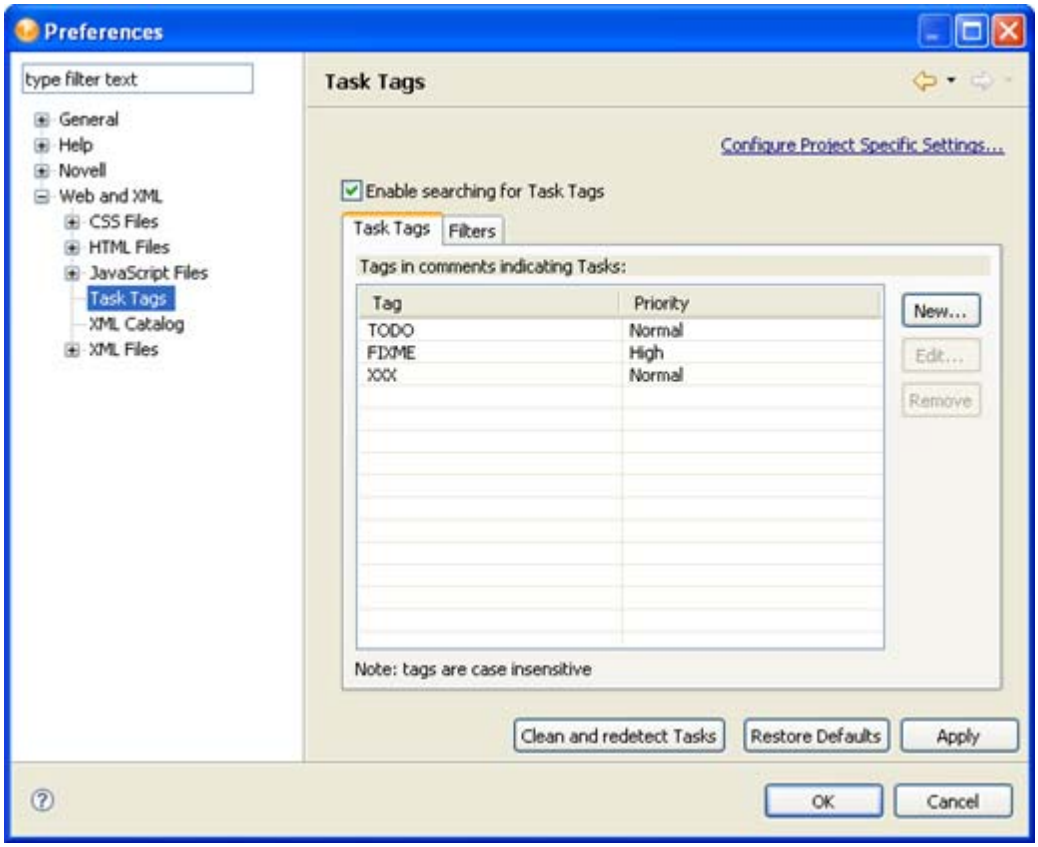


Table 20-45 *Preferences: Web and XML > JavaScript Files > Syntax Coloring Settings*

Setting	Description
Syntax Element	Choose the content type for which you want to define a style.
Foreground/Background/Bold/Italic/Strikethrough/ Underline	Specifies the syntax highlighting and formatting for individual CSS elements.
Sample Text	Displays sample CSS with the selected syntax coloring options.

20.5.4 Task Tags

Figure 20-27 *Preferences: Web and XML > Task Tags*



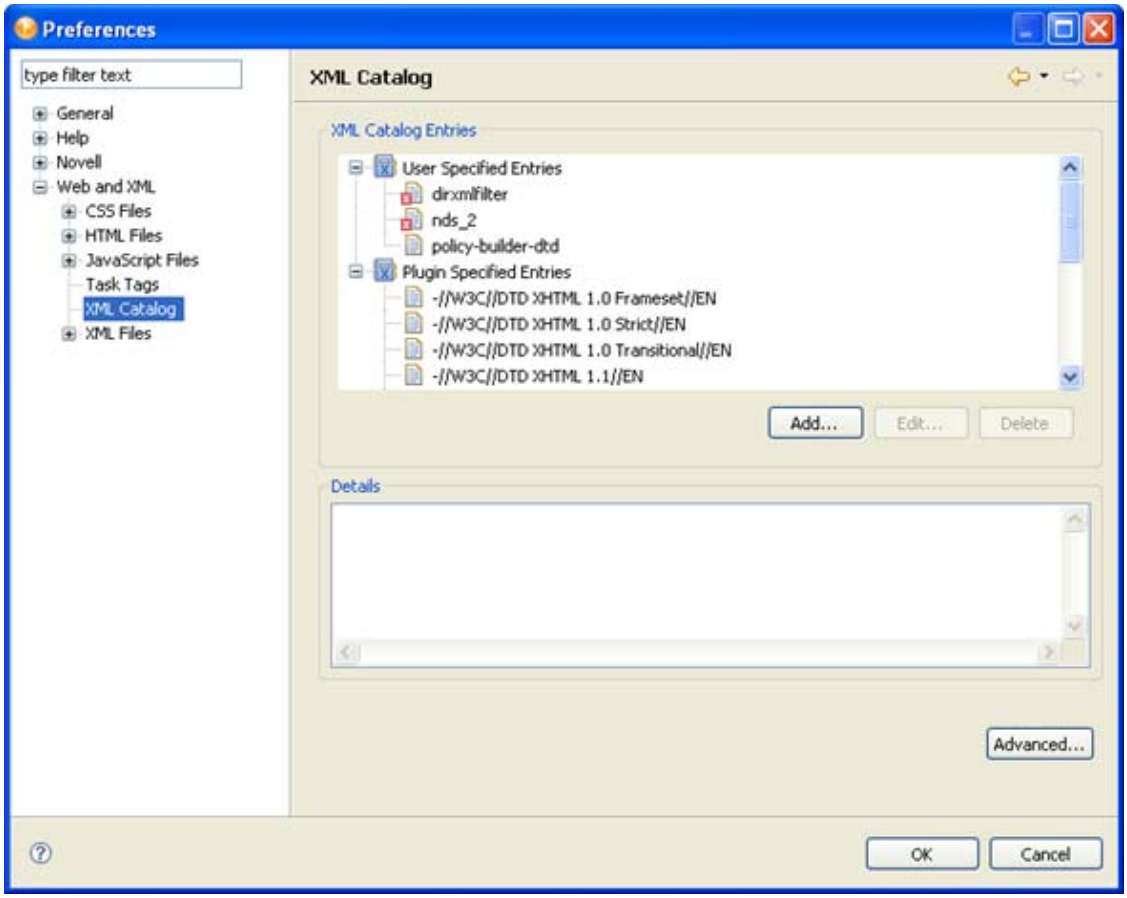
The Task Tag preferences allow you to specify whether Designer should search for task tags and which task tags to display in the Task view. If *Enable searching for Task Tags* is not selected, Designer does not search for task tags. If you do not use the Task view, make sure this option is not selected because the search can affect Designer's performance.

The *TODO* and *FIXME* task tags are defined by default, but you can specify additional tags by selecting *New* and completing the Task Tag dialog box. Use the *Filters* tab to specify the file types that Designer should search for the selected task tags.

20.5.5 XML Catalog

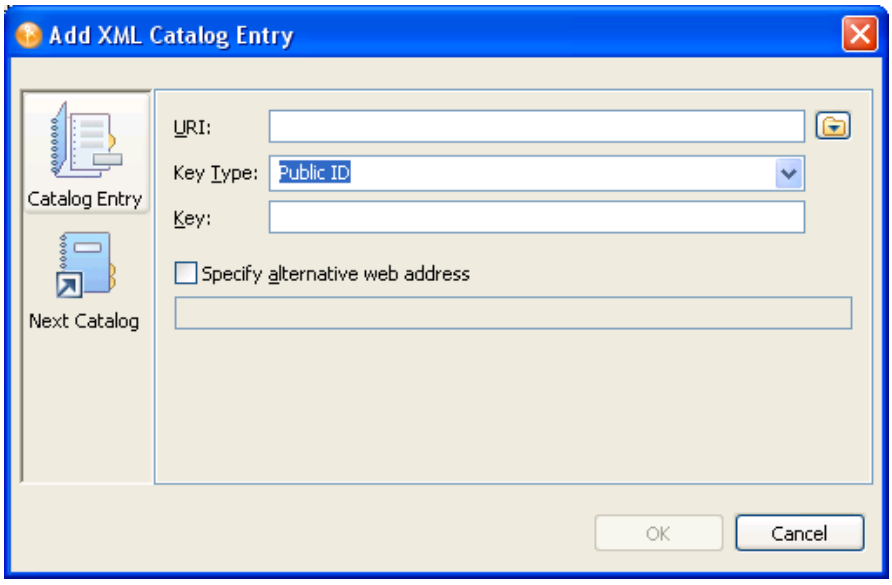
The XML Catalog preferences allow you to manage the WST XML catalog implementation. You can add, edit, or delete user-specified catalogs. You cannot use this preference to manage the plug-in specified entries.The XML editor uses the WST XML catalog implementation to resolve XML schema and DTD references for associating URLs, system, and public identifiers with URLs.

Figure 20-28 *Preferences: Web and XML > JavaScript > XML Catalog*



To add a user-specified entry:

1. Click Add.



2 Fill in the fields as follows:

Field	Description
Location	Specify a location on disk or a URL of the schema or DTD. Use the Search button to search Designer's workspace or the file system.
Key Type	Specify the key type. Values are public identifiers for DTDs or URIs for XML schemas.
Key	Specify a unique key
Specify alternative web address	Optionally, specify an alternative Web address for locating the schema or DTD.

3 Click *OK* to save.

20.5.6 XML Files

You can set the following general XML File preferences:

Figure 20-29 *Preferences: Web and XML > XML Files*

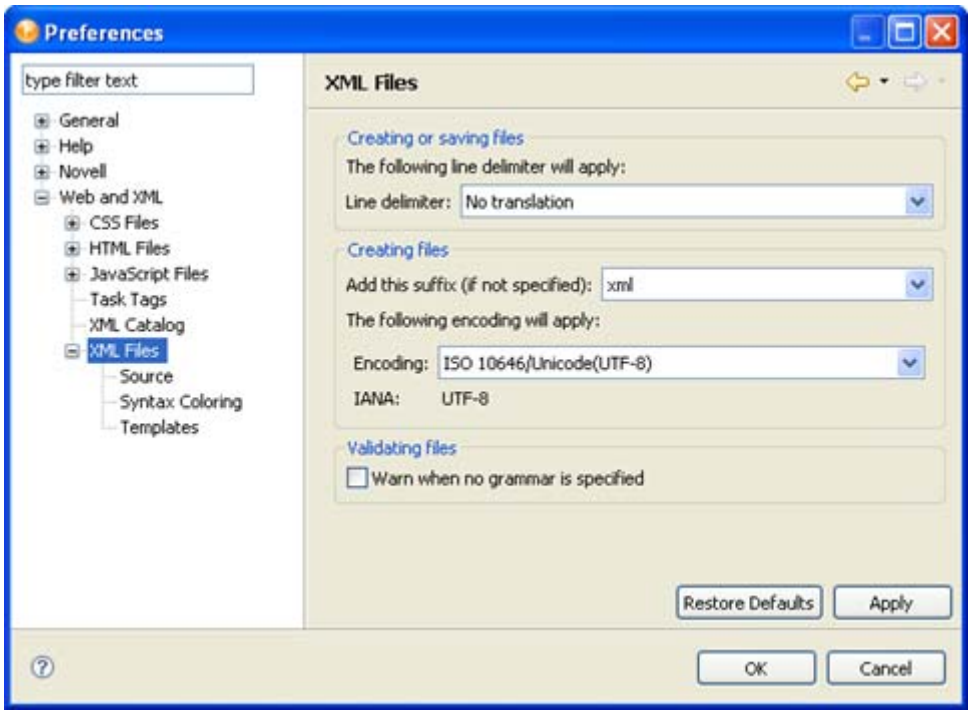


Table 20-46 *Preferences: Web and XML > XML Files*

Category	Preference	Description
Creating or saving files	*Line delimiter*	Choices are: ◆ *Windows* ◆ *Unix* ◆ *Mac* ◆ *No translation*
Creating files	*Add this suffix*	The file suffix. The default is XML.
	Encoding	The encoding used by the editor
Validating files	*Warn when no grammar is specified*	Specifies whether to display a warning when no grammar (such as XML Schema or DTD) is associated with the XML document.

- "XML Source" on page 489
- "Syntax Coloring" on page 491
- "XML Templates" on page 492

XML Source

You can set the following XML Source preferences:

Figure 20-30 *Preferences: Web and XML > XML Files > Source*

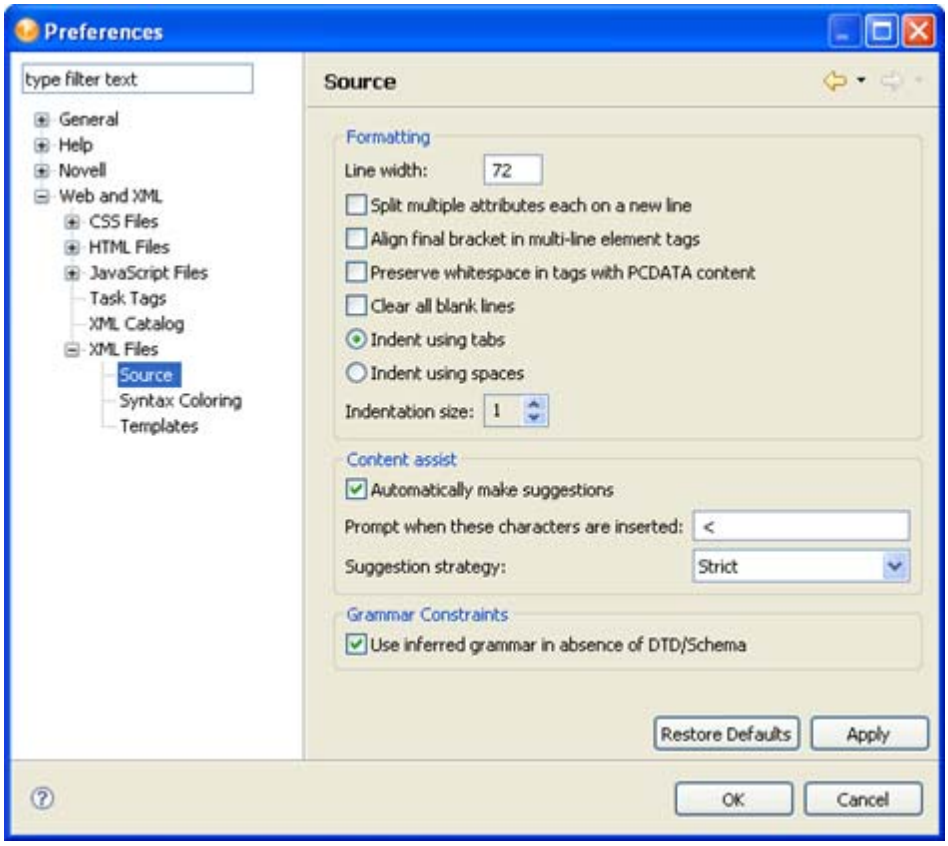


Table 20-47 *Preferences: Web and XML > XML Files > Source Settings*

Category	Preference	Description
Formatting	*Line width*	Specifies the number of characters in a line. Default is 72.
	Split multiple attributes each on a new line	Specifies how attributes are formatted (whether to show each attribute on a separate line).
	Align final bracket in multi-line element tags	Allows you to allign the final bracket “>” in multi-line element tags.
	Preserve whitespace in tags with PCDATA content	Specifies whether to preserve any whitespaces that are in tags containing PCDATA content.
	Clear all blank lines	Specifies whether blank lines are removed when formatting.
	Indent using tabs/ or spaces	Specifies whether to use tabs or spaces as indentation and indentation size.
	Indentation size	
Content Assist	*Automatically make suggestions*	Specifies whether to do automatic code completion.

Category	Preference	Description
	Prompt when these characters are inserted	The list of characters that initiate code completion.
	Suggestion strategy	Specifies whether to use Lax or Strict grammar when making suggestions
Grammar constraints	*Use inferred grammar in absence of DTD/Schema*	Specifies whether to display code completion suggestions based on existing content of the XML document.

Syntax Coloring

The XML syntax coloring lets you specify the syntax highlighting (foreground and background color) and the text formatting for individual XML constructs.

Figure 20-31 *Preferences: Web and XML > XML Files > Syntax Coloring*

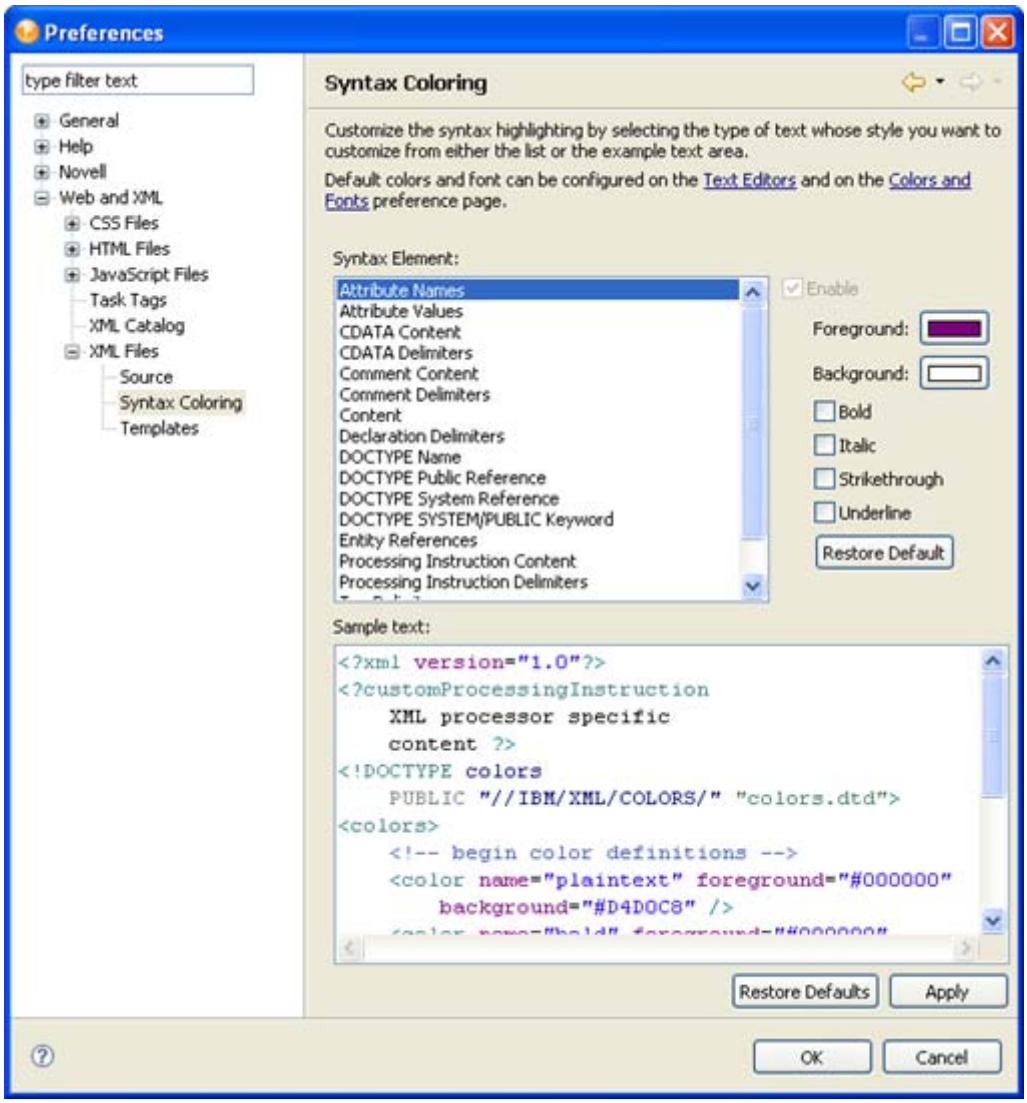


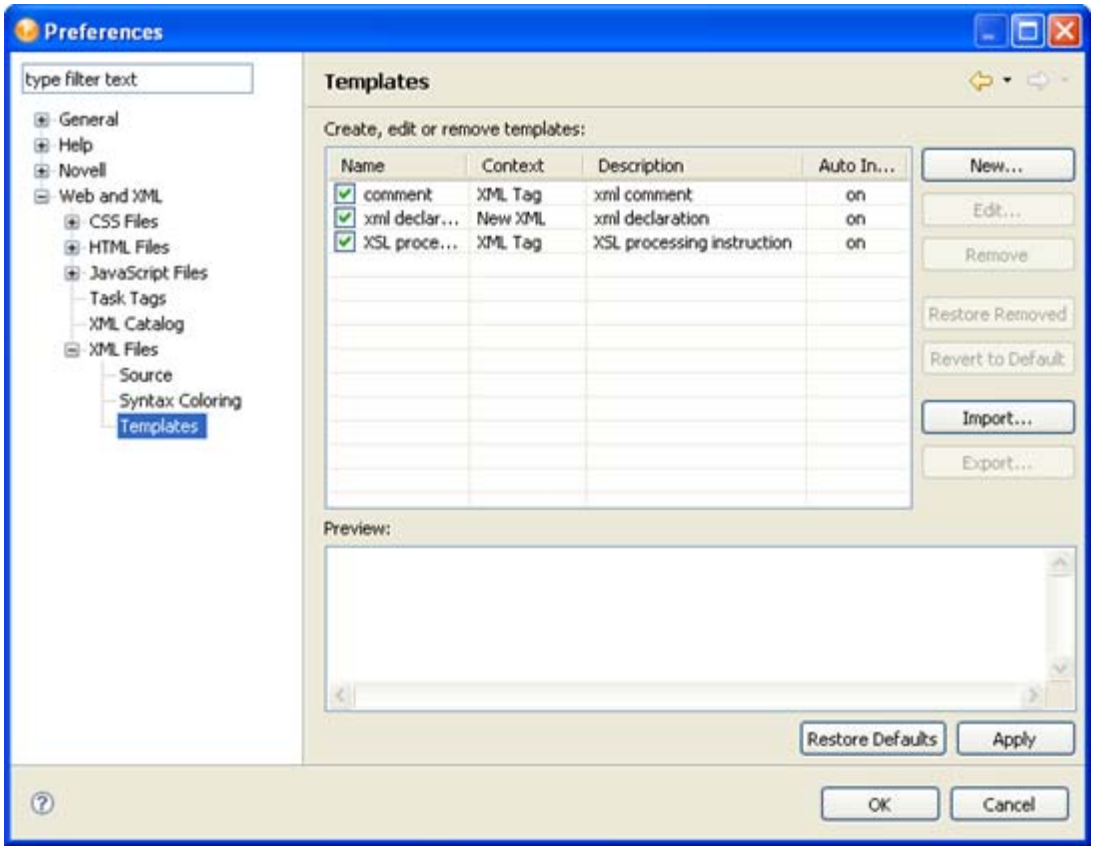
Table 20-48 *Preferences: Web and XML > XML Files > Syntax Coloring Settings*

Setting	Description
Syntax Element	Choose the content type for which you want to define a style.
Foreground/Background/Bold/Italic/Strikethrough/ Underline	Specifies the syntax highlighting and formatting for individual CSS elements.
Sample Text	Displays sample CSS with the selected syntax coloring options.

XML Templates

Use the XML Templates preference page to define XML templates. The templates are used in the code completion in the XML Source Editor. For example, selecting the `XSL Processing Instruction` template in the code completion inserts `<?xml-stylesheet type="text/xsl" href=""?>` in the source editor and places the cursor in the `href` value.

Figure 20-32 *XML Templates Preference*



Troubleshooting Designer 21

- Section 21.1, “Running the Project Checker,” on page 493
- Section 21.2, “Viewing the Error Log,” on page 493
- Section 21.3, “Turning On Trace Messages,” on page 496
- Section 21.4, “Checking Loaded Plug-Ins,” on page 498
- Section 21.5, “Deploying Identity Manager Objects,” on page 498
- Section 21.6, “No F1 Help in Maximized Editors,” on page 512
- Section 21.7, “Freeing Heap Memory,” on page 512
- Section 21.8, “Reporting Bugs and Giving Feedback,” on page 512

21.1 Running the Project Checker

Designer provides a Project Checker tool to check your project. The project can be checked at any time, but you should run the Project Checker before deploying your project. The Project Checker checks for proper design, contexts, server associations, policies, missing user data, and dependency problems that would cause the deployment of project into the Identity Vault to fail. It only checks the objects in Designer; it does not check the current objects in the Identity Vault.

21.2 Viewing the Error Log

The Error Log view can help you troubleshoot. If something isn’t working, messages written to the error log might help you. The log is named `.log`. It is a hidden file.

To view the error log, you can use menus or browse the file system.

- Section 21.2.1, “Browsing the File System,” on page 493
- Section 21.2.2, “Using Menus,” on page 494
- Section 21.2.3, “Event Details,” on page 494
- Section 21.2.4, “Customizing Filter Settings,” on page 495

21.2.1 Browsing the File System

1. Browse to your Designer workspace.

   In Windows, the log file is typically in subfolders in the `/eclipse/workspace/.metadata` directory.

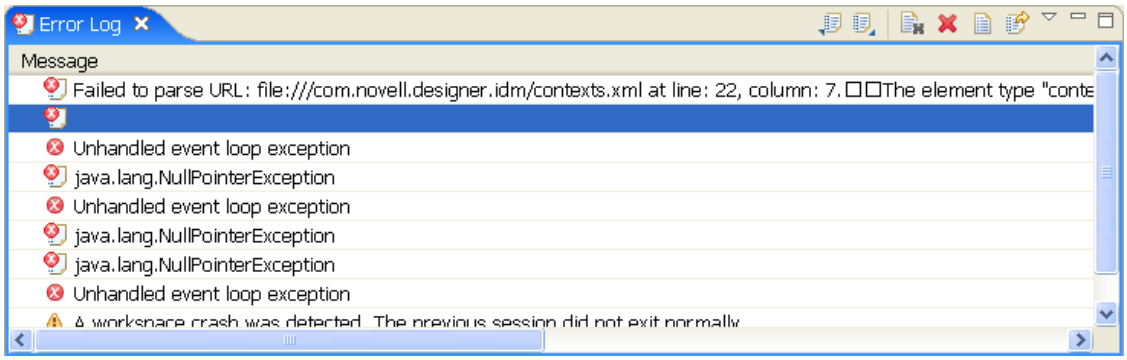
   In Linux, the log file is typically in the `Home` directory, in the `/eclipse/workspace/.metadata` directory.
2. Open the log file.

21.2.2 Using Menus

1. Select *Window* > *Show View* > *Other* > *PDE Runtime* > *Error Log*.
2. Click *OK*.

If you view the log through the application, a list of messages displays.

Figure 21-1 *The Error Log*



For a description of the icons located in the upper right corner of the Error Log view, see Error Log View (http://www.novell.com/documentation/designer30/designer_intro/data/projerrorlogview.html) in the *Understanding Designer for Identity Manager* .

The following options are available when you right-click inside the Error Log view:

Table 21-1 *Right-Click Options in the Error Log View*

Operation	Description
Copy	Enables you to copy event details to the clipboard.
Clear Log Viewer	Clears all the entries in the Error Log viewer.
Delete Log	Deletes all items in the Error Log.
Open Log	Opens an error log entry.
Restore Log	Enables you to restore log entries that have been previously cleared.
Export Log	Enables you to export the Error Log to a location on the file system.
Import Log	Enables you to import a file from the file system to the Error Log.
Event Details	Opens the Event Details window.

To sort messages in the Error Log view, click the appropriate header bar.

21.2.3 Event Details

To view event details:

1. Double-click an error log message.

or

Right-click an error log message, then click *Event Details*.

The following functionalities are available in the Event Details window:

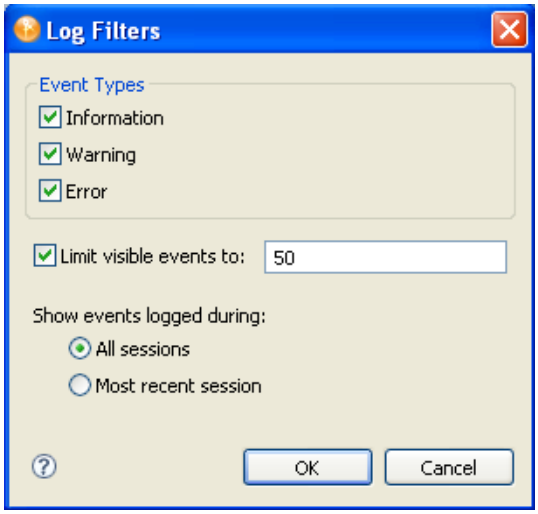
Table 21-2 *Event Details Window*

Operation	Description
Date	Displays the date and time the error occurred.
Severity	States the severity of the error.
Message	Displays the message of the error.
View Details of Previous Event	Up and down arrows that enable you to scroll through the event details of each event in the error log.
Copy	Enables you to copy event details to the clipboard.
Exception Stack Trace	Displays Exception Stack Trace (if available).
Session Data	Provides relevant session data.

21.2.4 Customizing Filter Settings

To access the Log Filters window:

1. On the Error Log view toolbar, click the *Menu* icon.
2. Click *Filters*.



The following functionalities are available in the Log Filters window:

Table 21-3 *Log Filters Window*

Operation	Description
Event Types	Set what type of information you want displayed in the error log. The error log can be configured to display any combination of Information, Warnings, and Errors.
Limit Visible Events	Set a limit on how many events you want displayed in the error log at one time.
Show Events Logged During:	Specify whether to show events logged during all sessions, or your most recent session.

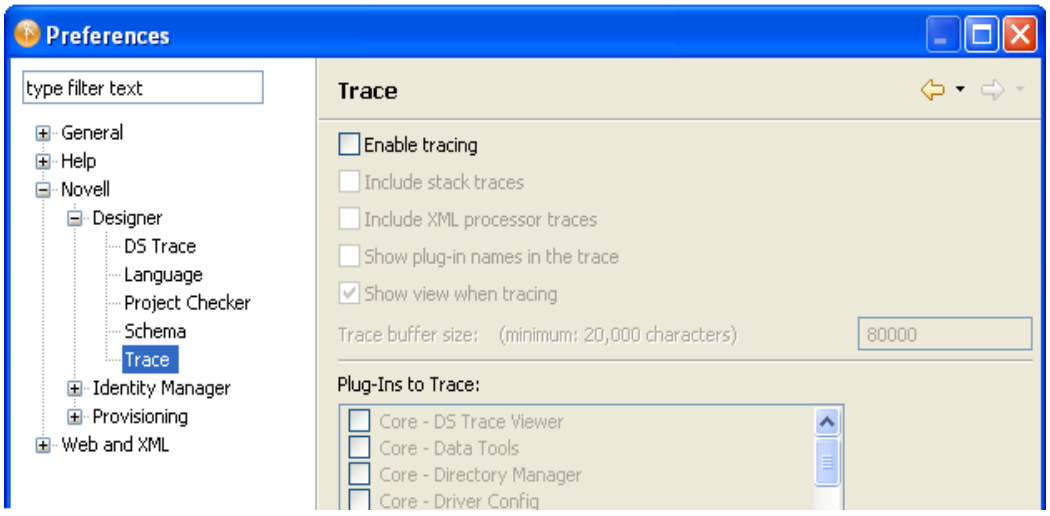
21.3 Turning On Trace Messages

You might want to send trace messages to the error log so that the messages are captured in a file. You can then easily e-mail the trace message to Novell Support or others.

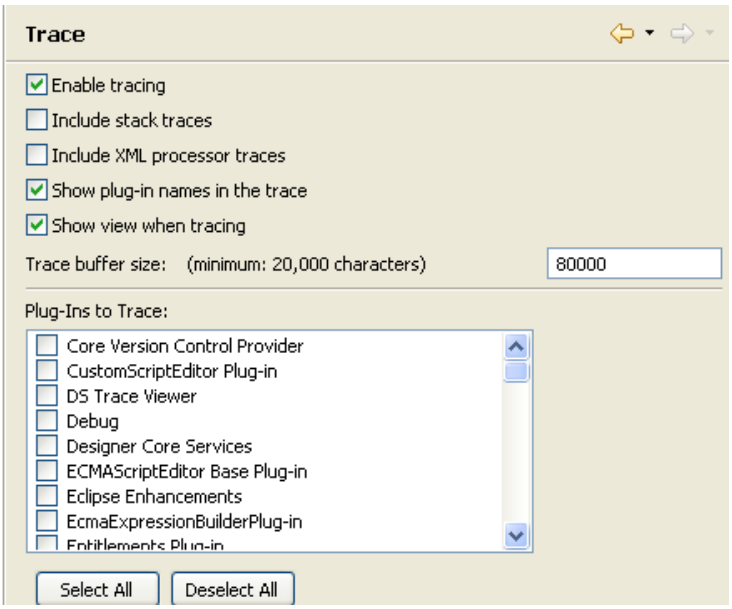
Programmers sometimes place hidden messages in their code so that if you are having problems, you can turn on the trace functionality and get additional insight. Even if you don't understand the hidden messages, they can help Novell Support diagnose the problem.

To get trace messages:

1. Click *Window* > *Preferences* to display the Preferences dialog box.
2. Click *Novell* > *Designer* > *Trace*.



3. Select *Enable tracing*, then select the options that you want to include or show.

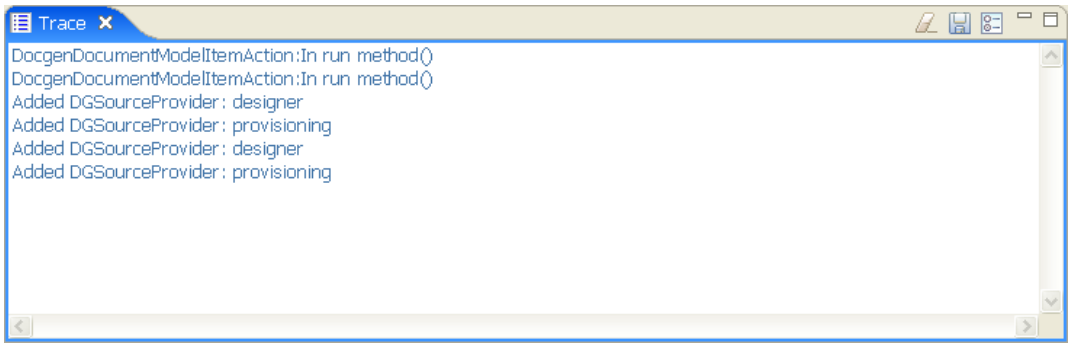


4 Select the plug-ins that you want to trace, then click *OK*.

To view the results of traces:

1 Select *Window* > *Show View* > *Trace*.

2 View data in the Trace view.



3 You can also turn on trace options from the Trace view by clicking the Preferences icon in the Trace view.

The following options are available when you right-click inside the Trace view:

Table 21-4 *Right-Click Options in the Trace View*

Option	Function
Undo	Undo a previously executed action.
Cut, Copy and Paste	Cut, copy and paste items in the Trace view by selecting the item, then clicking the desired action.
Delete	Enables you to delete items in the Trace view.

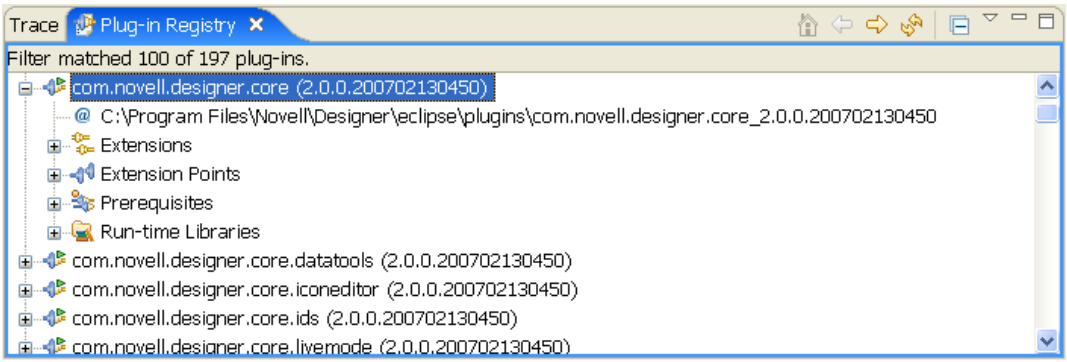
Option	Function
Select All	Enables you to simultaneously select all trace messages in the Trace view.

An icon toolbar is located in the upper right corner of the Trace view. For information on the icons in this toolbar, see Trace View (http://www.novell.com/documentation/designer30/designer_intro/data/projerrorlogview.html) in the *Understanding Designer for Identity Manager*.

21.4 Checking Loaded Plug-Ins

A problem can occur if a plug-in fails to load. To see which plug-ins are loaded:

1. Select *Window* > *Show View* > *Other*.
2. Open the PDE Runtime folder.
3. Click *Plug-in Registry* > *OK*.



The Plug-in Registry page lists the Designer plug-ins, which have a green triangle in the plug-in icon.

4. Use the *Home* icon to bring you to the top of the plug-in list.
5. Select a plug-in, then use the right-arrow icon to drill into the plug-in and use the left arrow icon to return.
6. Use the *Refresh* icon to refresh the Plug-In Registry view.
7. Use the Plug-In Registry view toolbar to select *Show Active Plug-Ins Only*.

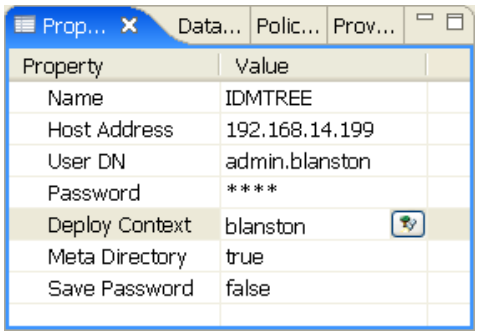
21.5 Deploying Identity Manager Objects

When you see an error message in Designer, the error message corresponds to the place where Designer could not complete the task, and indicates the best place to start troubleshooting. This section discusses the common problems you face when deploying Identity Manager objects into an eDirectory tree. To see error messages and possible solutions, see Section 21.5.3, “Error Messages and Solutions,” on page 501.

21.5.1 Deployment Considerations

- Ensure that the Metadirectory server meets the system requirements necessary to run Identity Manager. See Metadirectory Server (http://www.novell.com/documentation/idm36/idm_install/data/bemoll6.html) in the *Identity Manager 3.6.1 Installation Guide* for requirements.
- Ensure that the Metadirectory server you are deploying to has Identity Manager installed and holds a real copy of the objects to which you want to synchronize. The server running eDirectory must have a Master Read-Write or a Filtered Read-Write replica.
- Ensure that the Java software installed on the server is running correctly, because Identity Manager is dependent on Java. If Java is corrupted, you might be able to deploy to a Metadirectory server but not run the Identity Manager drivers.
- To deploy an Identity Manager-based project or an object in a project, you must have access to the eDirectory tree that is associated with the Identity Vault you are designing. Select the Identity Vault you want to deploy, then look in the Properties view below the Project/Outline view.

Figure 21-2 *The Properties View*



In the Properties view, ensure that the Identity Vault's Name, Host Address, User DN, Password, Deploy Context's Distinguished Name (DN), and Metadirectory information is complete and accurate. (You can click the *Browse* button to find the Deploy Context's DN on an existing tree if the other information is accurate and Designer can attach to the eDirectory tree.) You need this information to deploy anything, even a policy, into an existing eDirectory tree running the Metadirectory engine.

- Use the Deploy feature only after you have thoroughly tested the rules and policies that make up your drivers. To test a policy, use the Policy Simulator (right-click a policy and select *Simulate*, then click *Start* to see the simulation results of the policy that is being tested). For policy design, see the Policy Builder Help topics within the Designer utility.

  You can use the Import feature to import a driver, a channel, or a policy. You can then modify the object or objects, run the Policy Simulator to ensure that the object is working correctly, then deploy the object back into the test tree for further analysis. You can also run the Compare feature to see the differences between your modified driver and the driver that is currently running on an Identity Vault server.

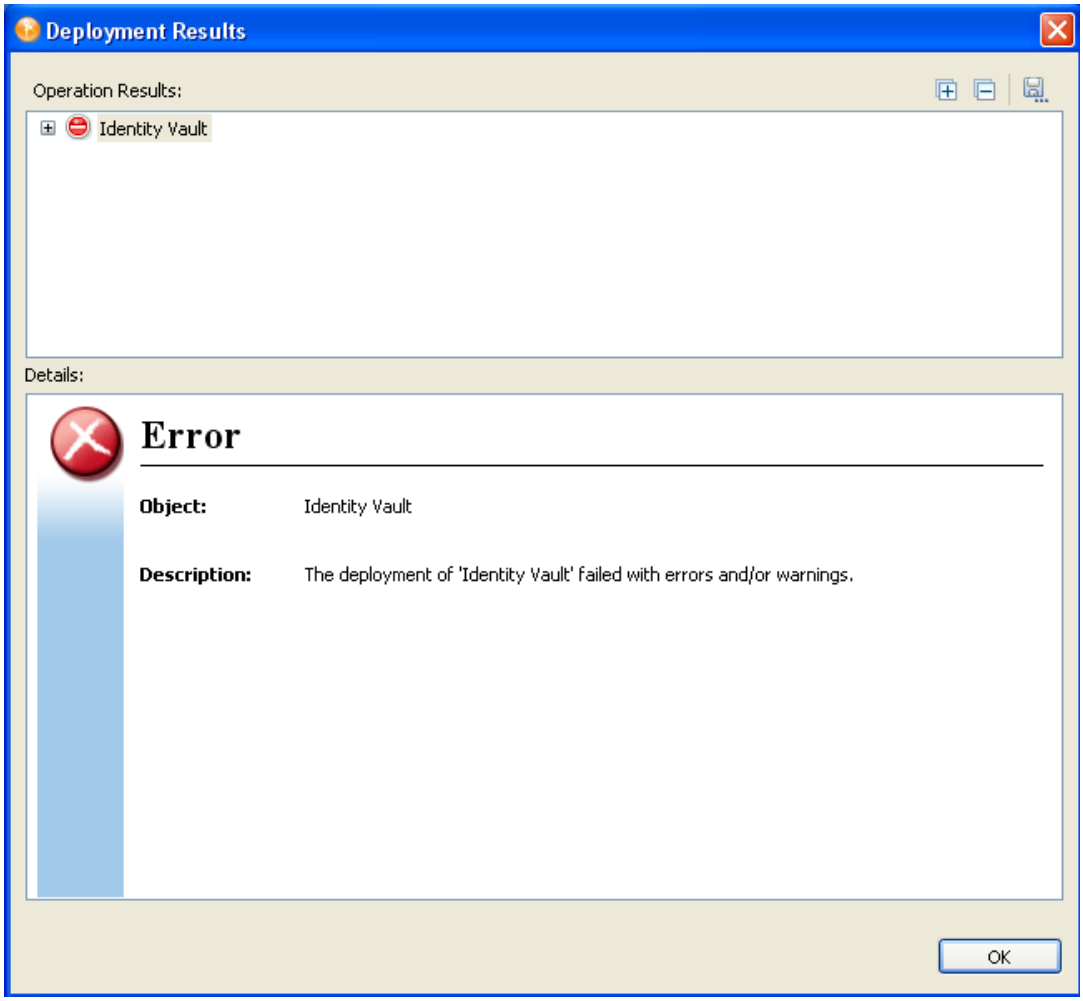
- In the Outline view in the Project Group view, right-click the driver object in question (you can also double-click the driver object). Use the Properties window to make most changes to drivers. Properties are unique to each driver.

  A simple driver problem is specifying the incorrect context (Distinguished DN) for an eDirectory tree. For example, the context of a user object in eDirectory is shown with the slash notation (for example, *Blanston\Sales\Users*) on the Properties of the Identity Manager driver or when you import the driver. However, different drivers can use formats other than the slash notation. For example, Active Directory and LDAP drivers use comma-delimited format (*OU=Users,OU=Sales,O=Blanston*). See the driver guides for further details on the drivers you are deploying.

21.5.2 An Example Deploy Error

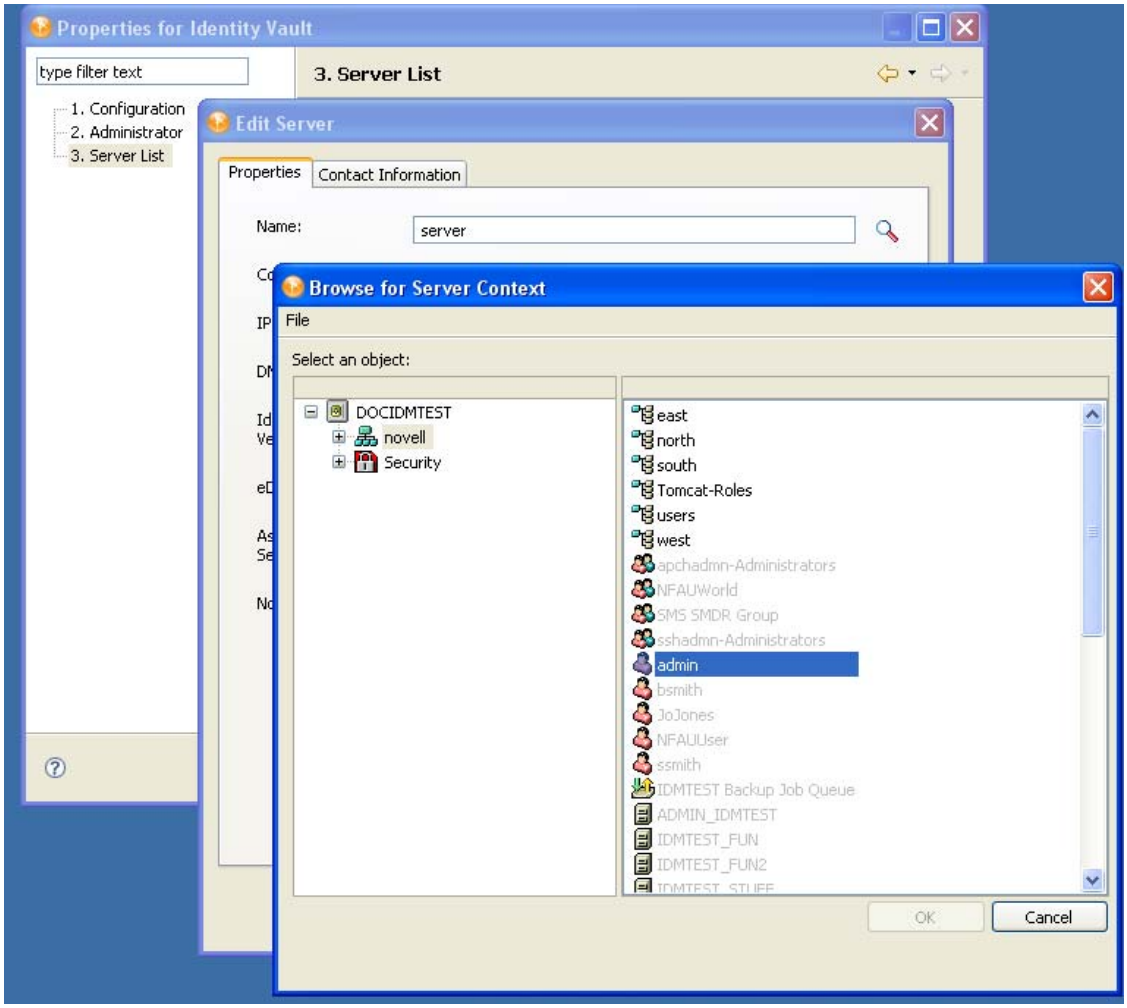
When deploying an Identity Vault for the first time, there are several common sources for errors, from incorrectly typing information to not completing the driver set templates.

Figure 21-3 *Default Server Container Message*



Right-click the Identity Vault in the Modeler view, select *Properties* > *Server List*, then click the *Edit* button to edit the server information.

Figure 21-4 *Correcting a Server Name Problem*



21.5.3 Error Messages and Solutions

When you see an error message in Designer, the error message corresponds to the place where Designer could not complete the task and indicates the best place to start troubleshooting. This section discusses the error messages you might see when deploying Identity Manager objects into an eDirectory tree, followed by their cause and possible solutions.

- “Identity Vault Configuration Errors” on page 502
- “Driver Configuration Errors” on page 502
- “Internal Designer Errors” on page 503
- “eDirectory Access Errors” on page 504
- “eDirectory Object/Attribute Creation Errors” on page 505
- “Warnings” on page 507

Identity Vault Configuration Errors

```
Cannot connect to host [Identity Vault Host]; verify the address is correct
and that the server is running.
```

Possible Cause: The address listed in the Identity Vault properties is incorrect or the server is not running.

Solution: Verify that the server address is correct and that the server is up and running.

```
[User] could not be authenticated to [Identity Vault Host]. Cannot proceed.
```

Possible Cause: The username or password listed in the Identity Vault properties is incorrect.

Solution: Verify the username specified in the Identity Vault properties and reenter the user's password.

Driver Configuration Errors

```
The driver configuration file [Driver Config File] is not a valid XML
document: [Error Message].
```

Cause: The Driver Configuration file being imported from the file system does not contain a valid XML document.

Solution: Fix the Driver Configuration file format.

```
The XML contained the file named [Driver Config File] is not a driver
configuration file. The file cannot be imported.
```

Cause: The Driver Configuration file being imported from the file system is a valid XML document but is not a valid driver configuration file.

Solution: Import a driver configuration file.

```
The following 'XML DOM Exception' was thrown.
[ExceptionInfo]
```

Cause: The Driver Configuration XML document is incorrectly formatted. This is probably an internal error because driver configuration files are dynamically generated by Designer for deployment.

Solution: Turn on trace for Designer. To do this, select *Window* > *Preferences* > *Designer for IDM Trace* > *Enable Tracing*. In the Trace window, select the check box for *Include XML Processor Traces*. Attempt to deploy again, then send the trace file to Novell Support.

```
The following 'Number Format Exception' was thrown.
[ExceptionInfo]
```

Cause: An integer value in the driver configuration file being deployed is invalid. All integer fields in Designer should validate the content when it is set.

Solution: Turn on trace for Designer. To do this, select *Window* > *Preferences* > *Designer for IDM Trace* > *Enable Tracing*. In the Trace window, select the check box for *Include XML Processor Traces*. Deploy again and analyze the generated driver configuration file to see if all integer attribute values are correct. Identify the incorrect parameter in Designer, correct the setting, and redeploy.

```
The specified driver configuration file does not contain a valid driver
configuration.
```

Cause: Designer attempted to process a dynamically generated driver configuration file with an invalid format.

Solution: Turn on XML tracing for the Import/Deploy plug-in. To do this, select *Window > Preferences > Designer for IDM Trace > Enable Tracing*. In the Trace window, select the check box for *Include XML Processor Traces*. Deploy again, then send the trace to Novell Support. Otherwise, edit and correct the configuration file being imported.

```
Tree population is not supported from a Driver Set configuration. Tree
population components will be ignored.
```

Cause: The driver configuration file being processed has a `<ds-object>` element under a `<driver-set-configuration>` element, which is not permitted.

Solution: If this is a dynamically generated configuration file, contact Novell Support; otherwise, move the `<ds-object>` element under a `<driver-configuration>` element.

```
The following Driver Set based global variables could not be resolved:
[Global Variable List]
These variables exist in both the source and target Driver Sets. The two
definitions, however, have different types.
```

Cause: The driver configuration file being processed has global variable definitions that could not be resolved.

Solution: If this is a dynamically generated configuration file, contact Novell Support. If it is a driver configuration file on disk, check the global variable definitions.

```
The driver configuration file being processed does not contain a valid driver
configuration.
```

Cause: The driver configuration file being processed does not contain a `<driver-configuration>` element.

Solution: If this is a dynamically created configuration file, turn on XML tracing for the Import/Deploy plug-in. To do this, select *Window > Preferences > Designer for IDM Trace > Enable Tracing*. In the Trace window, select the check box for *Include XML Processor Traces*. Deploy again, then send the trace to Novell Support. Otherwise, edit and correct the configuration file that is being imported.

```
The specified driver configuration file was only intended to be imported from
a ConsoleOne command line.
```

Cause: The driver configuration file being processed is not a valid document.

Internal Designer Errors

```
An internal error has occurred in the Designer Data Model: The policy named
[Policy Name] does not know its container.
```

Cause: The policy being deployed is not contained in a Channel or Driver object. This is an abnormal error, indicating that the Designer model has become corrupted.

Solution: Contact Novell Support.

eDirectory Access Errors

```
The following 'Component Creation Exception' occurred while trying to access
eDirectory.
[Exception Info]
```

Cause: A value contained in the driver configuration file being deployed could not be successfully created in eDirectory. This is probably an internal error because driver configuration files are dynamically generated by Designer for deployment. However, if the Driver in Designer was created by importing a driver configuration file from the file system and that configuration file contained a Tree Population Segment, a value within a <ds-object> element might be invalid.

Solution: Turn on trace for Designer. To do this, select *Window* > *Preferences* > *Designer for IDM Trace* > *Enable Tracing*. In the Trace window, select the check box for *Include XML Processor Traces*. Deploy again and analyze the generated driver configuration file to see if any <ds-object> elements exist. If they do, verify that all attribute values are correct. If no <ds-object> elements exist or if all values seem to be correct, contact Novell Support.

```
The following 'IO Exception' occurred while trying to access eDirectory.
[ExecptionInfo]
```

Cause: This is a Java exception indicating that Designer could not perform the requested input or output operation.

Solution: Contact Novell Support.

```
DSAccessException:
[ExceptionInfo]
```

Cause: Designer could not connect to the target deployment server.

Solution: Verify that the server information specified in the Identity Vault properties page is correct and that the eDirectory server is up and running.

```
The following 'Namespace Exception' occurred while trying to access
eDirectory. ({0})
```

Cause: This is a namespace exception indicating that there is a problem with the eDirectory schema, such as a missing attribute or class.

Solution: Verify that the eDirectory schema being imported from or deployed to is correct. If the driver being deployed contains Tree Population segments, verify that the objects being created are valid for the target eDirectory schema.

```
An exception occurred during the deployment. Cannot perform the operation.
```

Cause: An unknown exception was encountered.

Solution: Contact Novell Support.

```
The following 'Snapin Exception' occurred while trying to access eDirectory.
[ExceptionInfo]
```

Cause: Snap-in exceptions can be thrown in certain methods to report exceptions or errors during import/deploy. Subclasses of a snap-in exception include:

- NotAContainerException: There was a call to get the children of an eDirectory object that is not a container.

- ObjectNotFoundException: The object being resolved cannot be found in eDirectory.
- SPIException: Unable to connect to the eDirectory tree.

Solution: The exception might include the name of the object that caused the exception. Verify that the eDirectory tree being imported or deployed to is up and running and that it has Identity Manager installed.

`The following exception occurred but was not handled. ({0})`

Cause: An unexpected error occurred while resolving an object in eDirectory.

Solution: Contact Novell Support.

eDirectory Object/Attribute Creation Errors

`The driver could not be created.`

Cause: Designer attempted to create a driver in eDirectory, but the process failed.

Solution: Verify that the target eDirectory server has Identity Manager installed.

`A [ObjectClass] object named [ObjectName] could not be created.`

Cause: Designer attempted to create a Publisher, Subscriber, or Policy object in eDirectory, but the process failed.

Solution: Verify that the target eDirectory server has Identity Manager installed.

`The driver password could not be saved.`

Cause: Designer attempted to set the Driver password in eDirectory, but the request failed.

Solution: Verify that the target eDirectory server has Identity Manager installed.

`The password named ''{0}'' could not be saved.`

Cause: Designer attempted to set a named password in eDirectory, but the request failed.

Solution: Turn on stack tracing for the Import/Deploy plug-in to get details of the exception. To do this, select *Window* > *Preferences* > *Designer for IDM Trace* > *Enable Tracing*. In the Trace window, select the check box for *Include Stack Traces*.

`The value for the attribute named [Attribute Name] could not be stored on the object named [Object name].`

Cause: Designer attempted to add an attribute to an object in eDirectory, but the request failed. The error message should contain information about the attribute and object.

Solution: Verify that the attribute and value are valid for the given eDirectory object type.

`The value for the attribute named ''{0}'' could not be updated using the XSLT on the object named ''{1}''.`

Cause: Unable to export shim configuration information.

Solution: Contact Novell Support.

```
An exception was thrown updating the value of the [Attribute Name] attribute
on the [Item Type] object named [Object Name].
[Exception Info]
```

Cause: Unable to deploy the Identity Manager object and attributes to eDirectory. The error message should contain details of the exception.

Solution: Contact Novell Support.

`A [Object Class] object could not be created. The name is missing.`

Cause: An eDirectory object could not be created for the given object class because a name was not provided.

Solution: Contact Novell Support.

`The policy named [Policy Name] contains a cycle in its next transformation list.`

Cause: This is a warning message generated when Designer encounters a circular loop in the policy chain.

Solution: Remove the policy loop by correcting the next policy in the Policy Set view.

`The policies named [Policy name] contain cycles in their next transformation lists.`

Cause: This is a warning message generated when Designer encounters a circular loop in the policy chain.

Solution: Remove the policy loop by correcting the next policy in the Policy Set view.

`Driver [Driver name] could not be restarted for the deployed changes to be in effect.`

Cause: Designer was unable to restart a driver after a deployment.

Solution: Turn on DSTrace screen in eDirectory to identity the error preventing the driver from starting.

`Driver '[Driver Name]' is disabled and could not be restarted for the deployed changes to be in effect.`

Cause: Designer was unable to restart a driver after a deployment because its Driver Start option is set to *Disabled*.

Solution: Change the Driver Start option to *Manual* or *Auto-start* under the driver properties and then deploy the driver.

`Driver '[Driver Name]' could not be stopped for the deployed changes to be in effect.`

Cause: Designer was unable to stop a running driver after a deployment.

Solution: Turn on DSTrace screen in eDirectory to identify the error preventing the driver from stopping.

`An invalid request to set up security on an exported driver was made, no Driver objects were provided. The request cannot be processed.`

Cause: The code to set up the security equivalence for a deployed driver was passed an invalid parameter.

Solution: Contact Novell Support.

Warnings

```
The version of Identity Manager running on the server named '[Server Name]'
does not support all the features of Designer. Although you can import a
configuration from that server, changes may not work if the configuration is
deployed back to it.
```

Cause: An import or deploy action was made to an eDirectory server running an unsupported version.

Solution: The server must be upgraded for deployments.

```
An internal error has occurred. The parameters passed into the importer were
invalid.
```

Cause: The code that performs the import was passed an invalid parameter.

Solution: Contact Novell Support.

```
The '[Attribute Name]' attribute of '[Object Name]' refers to a policy that
does not exist or cannot be accessed.
```

Cause: The driver configuration file being processed contains a DN attribute that cannot be resolved in eDirectory.

Solution: Verify or correct the DN attribute value on the specified object in eDirectory.

```
An external reference to '[Object Name]' was not handled.
```

Cause: The driver configuration file being processed contains a DN attribute that cannot be resolved in eDirectory.

Solution: Contact Novell Support.

```
The XML for the policy named '[Object Name]' contained in the [Policy Type]
named '[Policy Name]' does not contain valid XML for a policy. '[Root Node]' is
not recognized as the root node for policy XML.
The policy is being ignored.
```

Cause: The policy being imported does not contain a valid XML document.

Solution: Correct the content of the policy in eDirectory.

```
A [Item Type] can only be imported into a [Item Type].
A [Item Type] can only be imported into a [Item Type] or [Item Type].
```

Cause: An attempt was made to import an Identity Manager object into an invalid parent object. For example, policies might not be imported into a Driver Set. The code should prevent this from happening, but this error identifies scenarios that were not caught.

Solution: Contact Novell Support.

```
An unhandled import request was encountered in DeployImporter_Import method
[Object DN].
```

Cause: An attempt was made to import an unknown object or attribute from eDirectory. The code should prevent this from happening, but this error identifies scenarios that were not caught.

Solution: Contact Novell Support.

```
Could not access the driver configuration file named '[File Name]'.
```

Cause: Designer could not open or parse the given driver configuration file.

Solution: Contact Novell Support.

```
The driver filter could not be read from the driver named '[Driver Name].
```

Cause: Designer could not import the Driver filter.

Solution: Turn on the DSTrace in eDirectory to determine the error, then contact Novell Support.

```
An error was encountered processing the driver configuration file. The
variable named [Variable Name] is defined more than once.
```

Cause: The driver configuration file has a variable that is being defined multiple times.

Solution: If you are importing a driver configuration file from a file, edit the file and remove multiple declarations for the specified variable. If this is a dynamically generated configuration file (import/deploy to eDirectory), turn on XML tracing for import/deploy to get a trace of the generated configuration file, then contact Novell Support. To turn on trace for Designer, select *Window* > *Preferences* > *Identity Manager* > *Application* > *Trace* > *Enable Tracing*. In the Trace window, select the check box for *Include XML Processor Traces*.

```
An error was encountered processing the driver configuration file. The
declaration of the Node variable named [Variable Name] is invalid. The
[Attribute name] attribute is missing.
```

Cause: The driver configuration file being processed has an invalid variable declaration.

Solution: If you are importing a driver configuration file from a file, edit the driver configuration file and correct the variable declaration. If this is a dynamically generated configuration file (import/deploy to eDirectory), turn on XML tracing for import/deploy to get a trace of the generated configuration file, then contact Novell Support. To turn on trace for Designer, select *Window* > *Preferences* > *Identity Manager* > *Application* > *Trace* > *Enable Tracing*. In the Trace window, select the check box for *Include XML Processor Traces*.

```
An error was encountered processing the driver configuration file. Flexible
prompting requires a 'use-when-value' when a 'use-when-var' is specified.
```

Cause: The driver configuration file being processed has an error.

Solution: If you are importing a driver configuration file from a file, edit the driver configuration file and add a use-when-value for the specified use-when-var. If this is a dynamically generated configuration file (import/deploy to eDirectory), turn on XML tracing for import/deploy to get a trace of the generated configuration file, then contact Novell Support. To turn on trace for Designer, select *Window* > *Preferences* > *Identity Manager* > *Application* > *Trace* > *Enable Tracing*. In the Trace window, select the check box for *Include XML Processor Traces*.

```
An error was encountered processing the driver configuration file. Flexible
prompting requires a 'use-when-var' when a 'use-when-value' is specified.
```

Cause: The driver configuration file being processed has an error.

Solution: If you are importing a driver configuration file from a file, edit the file and add a use-when-var for the specified use-when-value. If this is a dynamically generated configuration file (import/deploy to eDirectory), turn on XML tracing for import/deploy to get a trace of the generated configuration file, then contact Novell Support. To turn on trace for Designer, select *Window* > *Preferences* > *Identity Manager* > *Application* > *Trace* > *Enable Tracing*. In the Trace window, select the check box for *Include XML Processor Traces*.

```
The variable named [Variable Name] has been referred to but not defined in the
driver configuration file being processed.
```

Cause: The driver configuration file has a variable that is being referenced but has not been defined.

Solution: If you are importing a driver configuration file from a file, edit the driver configuration file and add a declaration for the specified variable. If this is a dynamically generated configuration file (import/deploy to eDirectory), turn on XML tracing for import/deploy to get a trace of the generated configuration file, then contact Novell Support. To turn on trace for Designer, select *Window* > *Preferences* > *Identity Manager* > *Application* > *Trace* > *Enable Tracing*. In the Trace window, select the check box for *Include XML Processor Traces*.

```
An error was encountered processing the driver configuration file. Built-in
variables cannot be used as a flexible prompting control variable. The
reference to the variable named '[Variable Name]' is invalid.
```

Cause: The driver configuration file being processed contains an invalid reference to a variable.

Solution: If this is a dynamically created configuration file generated during an import/deploy action, contact Novell Support. If this is a driver configuration file being imported from disk, edit and correct the configuration file for the variable specified.

```
An error was encountered processing the driver configuration file. There was a
non-checkbox reference to the checkbox variable named '[Check Box Variable
name]'.
```

Cause: The driver configuration file being processed contains an invalid reference to a check box variable.

Solution: If this is a dynamically created configuration file that is generated during an import/deploy action, contact Novell Support. If this is a driver configuration file being imported from disk, edit and correct the configuration file for the check box variable specified.

```
An error was encountered processing the driver configuration file. An
unhandled import prompt was encountered.
```

Cause: The driver configuration file being processed contains an invalid prompt type.

Solution: If this is a dynamically created configuration file that is generated during an import/deploy action, contact Novell Support. If this is a driver configuration file being imported from disk, edit and correct the configuration file.

```
The eDirectory tree corresponding to the Identity Vault named '[Identity Vault
Name]' cannot be accessed. Directory browsing cannot be performed.
```

Cause: Designer attempted to access eDirectory through an eDirectory browse button in the Driver Configuration Wizard, but the connection could not be created.

Solution: Cancel out of the Driver Configuration Wizard, set up the connection parameters in Identity Vault, and run the Driver Configuration Wizard again.

```
The partition could not be created on the ''{0}'' object. The problem may be
that it has not replicated to the master yet. You can try creating the
partition manually later.
```

Cause: Designer attempted to create a partition when deploying a driver set and the partition operation failed.

Solution: Turn on the eDirectory tracing options for partitioning to determine why the eDirectory partitioning operation failed.

```
The Driver Set was created but did not replicate to all the servers in the
replica ring. The deployment cannot proceed.
```

Cause: Designer cannot deploy per-server attributes until the driver set has replicated to the eDirectory server.

Solution: Turn on the eDirectory tracing options for replication and determine why eDirectory replication is not occurring.

```
There are no servers associated with the Driver Set named ''{0}''. There must
be at least one server associated with any Driver Set being deployed or the
Driver Set containing any objects being deployed.
```

Cause: Designer cannot deploy an Identity Vault or driver set with an empty server list.

Solution: Edit the properties of the Identity Vault and the driver set to add a server to the server lists.

```
The Identity Vault name '[Identity Vault Name]'' does not contain any Driver
Set objects to deploy.
```

Cause: You cannot deploy an Identity Vault that does not contain at least one driver set.

Solution: Add a driver set to the Identity Vault.

```
'[User Name]' could not be authenticated to '[Host Name]'. Cannot proceed.
```

Cause: Designer could not authenticate to the eDirectory tree.

Solution: Verify that the hostname, user, and password for the Identity Vault are correct in the Identity Vault properties.

```
The Identity Vault named '[Identity Vault Name]' does not contain the
eDirectory tree to access. Cannot proceed.
```

Cause: The Identity Vault does not contain a host address or DNS name for authentication.

Solution: Specify the host address or DNS name for the Identity Vault in the Properties view or Properties page.

```
Deploy_Util_NoIdentityVault=The {2} named ''{1}'' is not contained in an {0}.
Cannot proceed.
The Identity Vault named '[Identity Vault name]' does not contain the DN of the
user to authenticate to the target eDirectory tree with. Cannot proceed.
```

Cause: The Identity Vault does not contain a user for authentication.

Solution: Specify the user for the Identity Vault in the Properties view or Properties page.

```
The server list on the parent Driver Set for the following eDirectory Driver is
empty. We were unable to import the connected eDirectory Driver:
```

Cause: Designer uses the per-server Shim Auth Server attribute of an eDirectory driver to identify the tree and connected eDirectory driver to import. Because the server list is empty, the connected eDirectory driver cannot be imported.

Solution: Fix the server list on the driver set for the eDirectory driver and the Drivers Shim Auth Server attribute in eDirectory, or import the connected eDirectory driver separately.

```
The Shim Auth Server parameter for the eDirectory Driver '[Driver Name]' on
server '[Server Name]' is empty. We were unable to import the connected
eDirectory Driver.
```

Cause: Designer uses the Shim Auth Server parameter of an eDirectory driver to identify the tree and connected eDirectory driver to import. If this parameter is empty, the connected eDirectory driver cannot be imported.

Solution: Fix the Shim Auth Server parameter on the eDirectory driver or import the connected eDirectory driver separately.

```
Unable to save Driver Configuration to file '[File Name]'.
```

Cause: Designer was unable to save an exported driver configuration file.

Solution: Try to save the file to a different directory or filename.

```
Unable to clear contents of Driver Configuration file '[File Name]'.
```

Cause: Designer was unable to clear the contents of a driver configuration file that is being overwritten.

Solution: Delete the configuration file being overwritten.

```
Setting up the Security Equals and Excluded objects may only be performed on a
Driver object.
```

Cause: An invalid object was selected in the Modeler or Outline view.

Solution: Select a single Driver object to set up security equivalences or excluded users.

```
The selected Driver ''{0}'' has not been deployed or cannot be found in the
eDirectory ''{1}''.
```

Cause: Designer cannot resolve to the Driver object in eDirectory to set up the security equivalences or excluded user list.

Solution: Deploy the driver to eDirectory before setting up the security equivalences or excluded users.

```
The eDirectory tree corresponding to the Identity Vault named '[Tree Name]'
cannot be accessed. Setting up the Driver Security Equivalence/Excluded Users
cannot be performed.
```

Cause: Designer cannot connect or authenticate to the eDirectory tree to set up a driver's security equivalences or excluded user list.

Solution: Verify that the eDirectory parameters specified on the Identity Vault are correct and that the eDirectory server is running.

```
The Identity Vault named '[Identity Vault Name]' has no deployment DN
specified. It is not deployable.
```

Cause: A deployment context is not specified on the Identity Vault or driver set being deployed.

Solution: Add a deploy DN (context) to the properties of the Identity Vault or Driver Set object in Designer.

21.6 No F1 Help in Maximized Editors

Context-sensitive help is available when you press F1. However, if you maximize an editor (for example, the Modeler), help topics do not display when you press F1. To view the help, minimize the editor.

21.7 Freeing Heap Memory

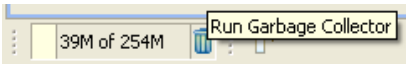
A status field at the bottom of Designer displays heap memory used and heap memory available for an application or other item in Designer.

Figure 21-5 *The Heap Memory Display*

The information varies, depending on which item (for an example, an application) you click in the Modeler, Outline view, or other editors.

To free unused heap memory at any time, click the *Run Garbage Collector* icon.

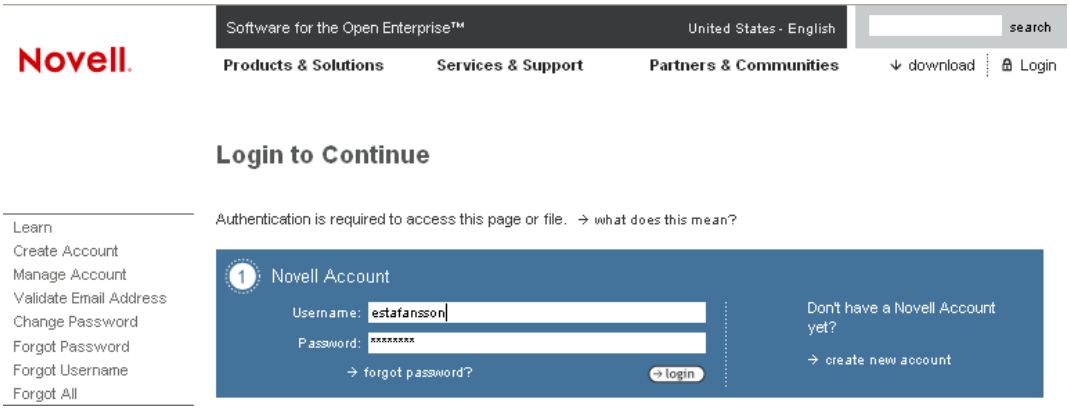
Figure 21-6 *The Run Garbage Collector Icon*



21.8 Reporting Bugs and Giving Feedback

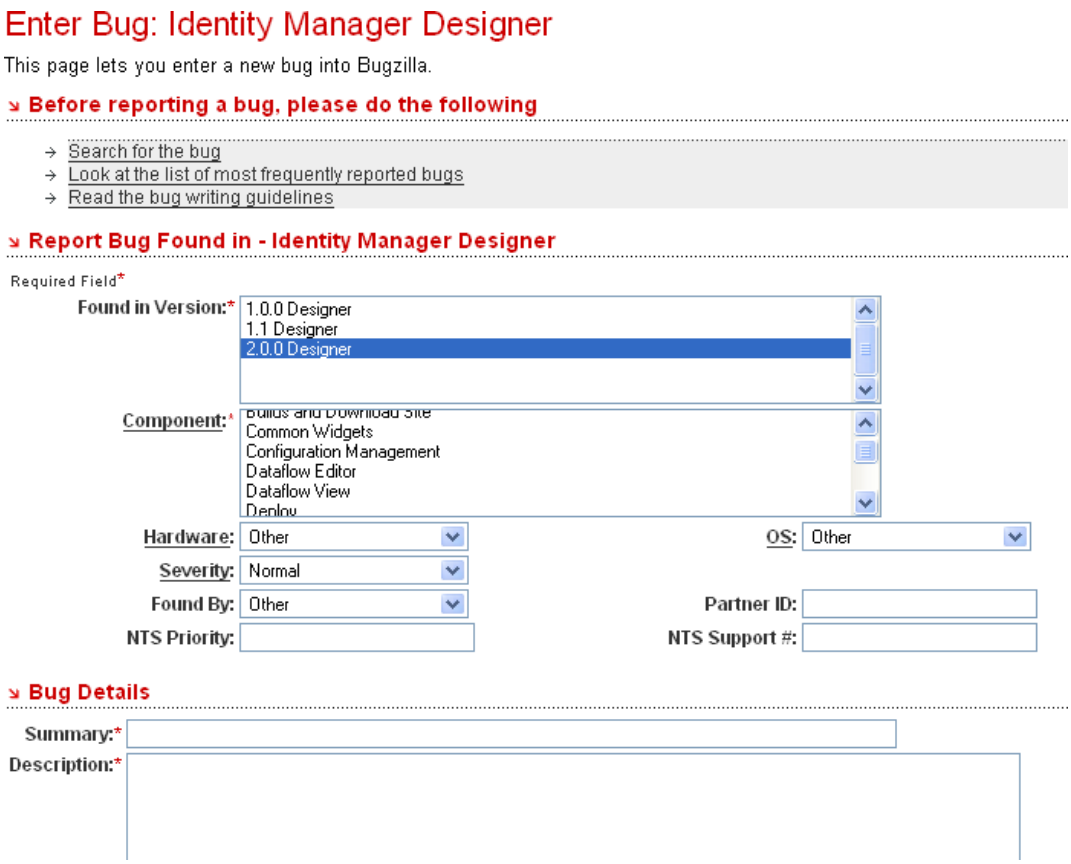
Gathering bugs and getting your ideas are keys to improving the performance of Designer and making Designer a tool of choice for you. To send us your feedback, select *Help* > *Report a Bug or Feedback*. We encourage you to try it!

1. Select *Help* > *Report a Bug or Give Feedback*.
2. Log in to Bugzilla.



If you don’t have an account, you can easily create one.

3. Select the component in the product that you are reporting on.



The Designer 4.0 product is preselected. If you don't know which component you are reporting on, select your best guess (for example, Modeler).

4 In the *Summary* field, summarize the problem or your request for an enhancement.

5 In the *Description* field, describe the bug or enhancement.

If you are reporting a bug, provide clear steps on how to reproduce the problem.

Modeler Operations

A

Modeler operations are available when you right-click inside the Modeler. The list of operations depends upon whether you right-click the Modeler space or one of the objects.

- Section A.1, “Modeler Space Operations,” on page 515
- Section A.2, “Identity Vault Operations,” on page 516
- Section A.3, “Driver Set Operations,” on page 519
- Section A.4, “Driver Operations,” on page 521
- Section A.5, “Application Operations,” on page 525
- Section A.6, “Submenus,” on page 528
- Section A.7, “Keyboard Support,” on page 529

A.1 Modeler Space Operations

The following figure illustrates Modeler options that are available when you right-click empty Modeler space.

Figure A-1 *Modeler-Space Operations*

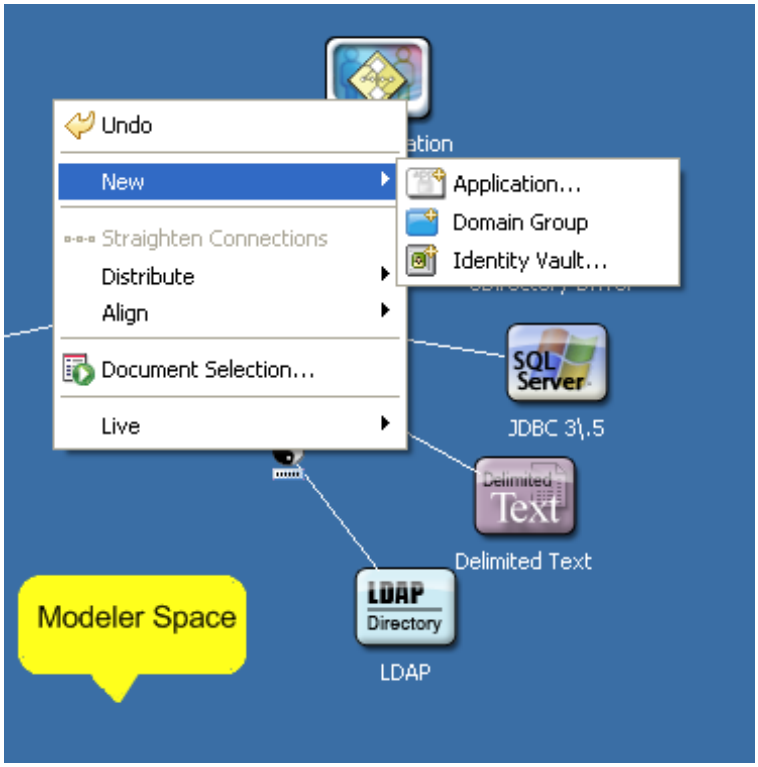


Table A-1 *Modeler-Space Operations*

Right-Click Operation	Description
Undo Change Location	Returns an item to its previous status.
New > Application	Selects an application from a list and places the application to the Modeler.
New > Domain Group	Places a Domain Group in the Modeler.
New > Identity Vault	Launches a dialog box that specifies a server and creates an Identity Vault.
Straighten Connections	Straightens lines for selected items. For example, you can straighten a line to a driver, all lines in a driver set, everything in a Domain Group, or an entire project. If a line is not within a few degrees of being horizontal or vertical, this option is dimmed.
Distribute	Evenly distributes applications vertically or horizontally. Press Ctrl, select the items that you want to distribute, then select a pattern.
Align	Aligns applications according to a pattern that you select. Press Ctrl, select the items, then select a pattern (for example, Align Bottom).
Document Selection	Launches the Document Generation Wizard, which documents your project.
Live > Import	Imports and Identity Vault.

A.2 Identity Vault Operations

The following figure illustrates Modeler operations that are available when you right-click an Identity Vault.

Figure A-2 *Identity Vault Operations*

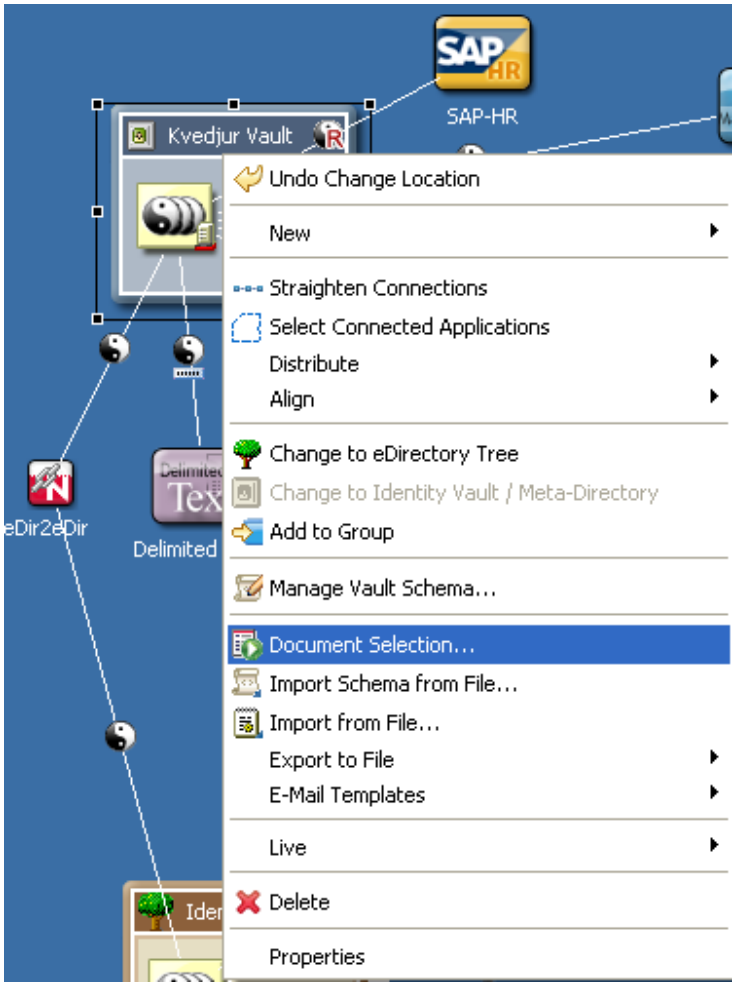


Table A-2 *Identity Vault Operations*

Operation	Description
New > Driver Set	Adds a Driver Set object to an Identity Vault.
New > Library	Launches the New Library Wizard.
New > Server	Launches the Add Server Wizard.
Straighten Connections	Straightens lines for selected items. For example, you can straighten a line to a driver, all lines in a driver set, everything in a Domain Group, or an entire project. If a line is not within a few degrees of being horizontal or vertical, this option is dimmed.
Select Connected Applications	Selects all applications that are connected to the driver set or Identity Vault. This is convenient if you have several applications connected to a driver set. You can quickly move them all or delete them all without browsing to and selecting each one.

Operation	Description
Distribute	Evenly distributes applications vertically or horizontally. Press Ctrl, select the items that you want to distribute, then select a pattern.
Align	Aligns applications according to a pattern that you select. Press Ctrl, select the items, then select a pattern (for example, Align Bottom). See Table A-6 on page 528.
Change to eDirectory Tree	Changes an Identity Vault to an eDirectory™ tree. In Architect mode, this option displays a tree instead of a vault. This is just for diagramming purposes; there is no functional difference.
Cha*nge to Identity Vault/Metadirectory*	Changes an eDirectory tree into an Identity Vault. In Developer mode, this option displays a vault instead of a tree. This is just for diagramming purposes; there is no functional difference.
Add to Group	Creates a Domain Group, and adds the selected items to it. The selected items are removed from any group to which they were previously associated.
Manage Vault Schema	Launches the Schema Manage tool, from which you can manipulate schema settings for the selected Identity Vault or directory.
Document Selection	Launches the Document Generation Wizard, which documents the selected Identity Vault.
Import Schema from File	Enables you to browse to a file and import a schema into a `.sch` or `.ldif` file.
Import from Configuration File	Allows you to browse to and import a driver configuration file.
Export to File > Configuration	Exports the Identity Vault to a `.xml` file. iManager can consume this format, and Designer can re-import it. For more information, see Section 15.10, “Exporting To a File,” on page 357.
Export to File > Schema	Exports the schema to a `.sch` or `.ldif` file.
E-mail Templates > E-Mail Server Properties	Configures an e-mail server to send e-mail notifications. Edits templates used to notify users concerning password events. For more information, see Section 10.5, “Configuring the E-Mail Server,” on page 223.
E-Mail Templates > Edit Templates	Launches the E-mail Templates dialog box, from which you can edit the e-mail templates associated with the selected Identity Vault. For more information, see Chapter 10, “Setting Up E-Mail Notification Templates,” on page 215.
E-Mail Templates > Update Templates	Adds localized templates to the Default Notification Collection.
Live > Import	Enables you to connect to a server, browse to and select objects, and import the objects into the Identity Vault.
Live > Deploy	Prepares a deployment summary and then deploys selected objects and attributes.
Live > Compare	Compares selected Identity Vaults. Enables you to reconcile or update Identity Vaults. See Section 15.7, “Using the Compare Feature When Deploying,” on page 348.

Operation	Description
Live > Schema > Import	Imports the schema from an existing Identity Vault.
Live > Schema > Deploy	Deploys the modified or imported schema.
Live > iManager	Enables you to connect to a server and launch iManager.
Live > Manage Directory	Launches the eDirectory Object Manager, which allows you to view, and edit attributes for, the selected Identity Vault. For more information, see Section 17.6, “Managing Directory Objects,” on page 389.
Live > Status for All Drivers	Lists drivers that are stopped or running.
Live > Start All Drivers	Starts all drivers associated with the selected object.
Live > Stop All Drivers	Stops all drivers associated with the selected object.
Live > Restart All Drivers	Restarts all drivers associated with the selected object.
Delete	Deletes the Identity Vault.
Properties	Displays the Identity Vault’s properties pages.

A.3 Driver Set Operations

The following figure illustrates Modeler operations that are available when you right-click a driver set.

Figure A-3 *Driver Set Operations*

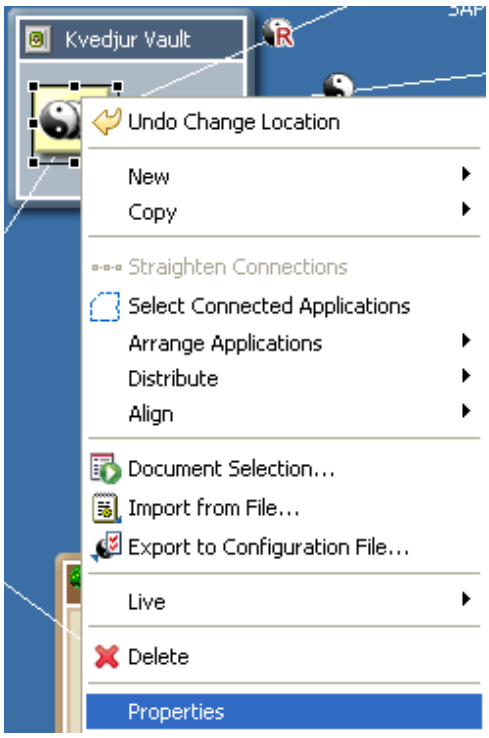


Table A-3 *Driver Set Operations*

Operation	Description
New > Driver	Launches the Driver Configuration Wizard to add a driver to the driver set.
New > Job	Launches the Job Scheduler Wizard to create a job.
New > Library	Launches the New Library Wizard.
Copy > Driver Set Settings	Enables you to browse to a driver set and copy its settings. A pasted copy overwrites data in the target driver set.
Copy > Global Configuration Values	Enables you to copy Global Configuration Values (GCVs) from one driver set to one or more other driver sets. This option enables you to configure GCVs in one place and then apply GCV settings to selected targets.
Straighten Connections	Straightens all lines in the driver set. If a line is not within a few degrees of being horizontal or vertical, this option is dimmed.
Select Connected Applications	Selects all applications that are connected to the driver set. You can quickly move or delete the applications without browsing to and selecting each one.
Arrange Applications	Arranges application icons around their associated driver set icon. A check mark indicates the current layout for the driver set. After the layout is set, any applications that you connect are automatically snapped into that layout. For more information, see Table A-7 on page 528.
Distribute	Evenly distributes applications vertically or horizontally. Press Ctrl, select the items that you want to distribute, then select a pattern.
Align	Aligns applications according to a pattern that you select. Press Ctrl, select the items, then select a pattern (for example, Align Bottom). See Table A-6 on page 528.
Document Selection	Launches the Document Generation Wizard, which documents the selected driver set.
Import from Configuration File	Reads in exports made from iManager or Designer. For more information, see Section 11.3, “Importing a Driver Configuration File,” on page 245.
Export to Configuration File	Exports the driver set to a `.xml` file. iManager can consume this format, and Designer can re-import it. For more information, see Section 15.10, “Exporting To a File,” on page 357.
Live > Import	Enables you to connect to a server, browse to and select objects, and import the objects into the driver set.
Live > Deploy	Prepares a deployment summary and then deploys selected objects and attributes.
Live > Compare	Compares selected driver sets. Enables you to reconcile or update driver sets. See Section 15.7, “Using the Compare Feature When Deploying,” on page 348.

Operation	Description
Live > Driver Set Configuration > Import Attributes	Imports attributes from an existing driver set.
Live > Driver Set Configuration > Deploy Attributes	Deploys the modified or imported attributes.
Live > Driver Set Configuration > Compare Attributes	Compares attributes in Designer to the connected Metadirectory server.
Live > Status for All Drivers	Lists drivers that are stopped or running.
Live > Start All Drivers	Starts all drivers associated with the selected object.
Live > Stop All Drivers	Stops all drivers associated with the selected object.
Live > Restart All Drivers	Restarts all drivers associated with the selected object.
Delete	Deletes the driver set.
Properties	Enables you to configure Identity Vaults, driver sets, drivers, and applications.

A.4 Driver Operations

The following figure illustrates Modeler operations that are available when you right-click a driver.

Figure A-4 *Driver Operations*

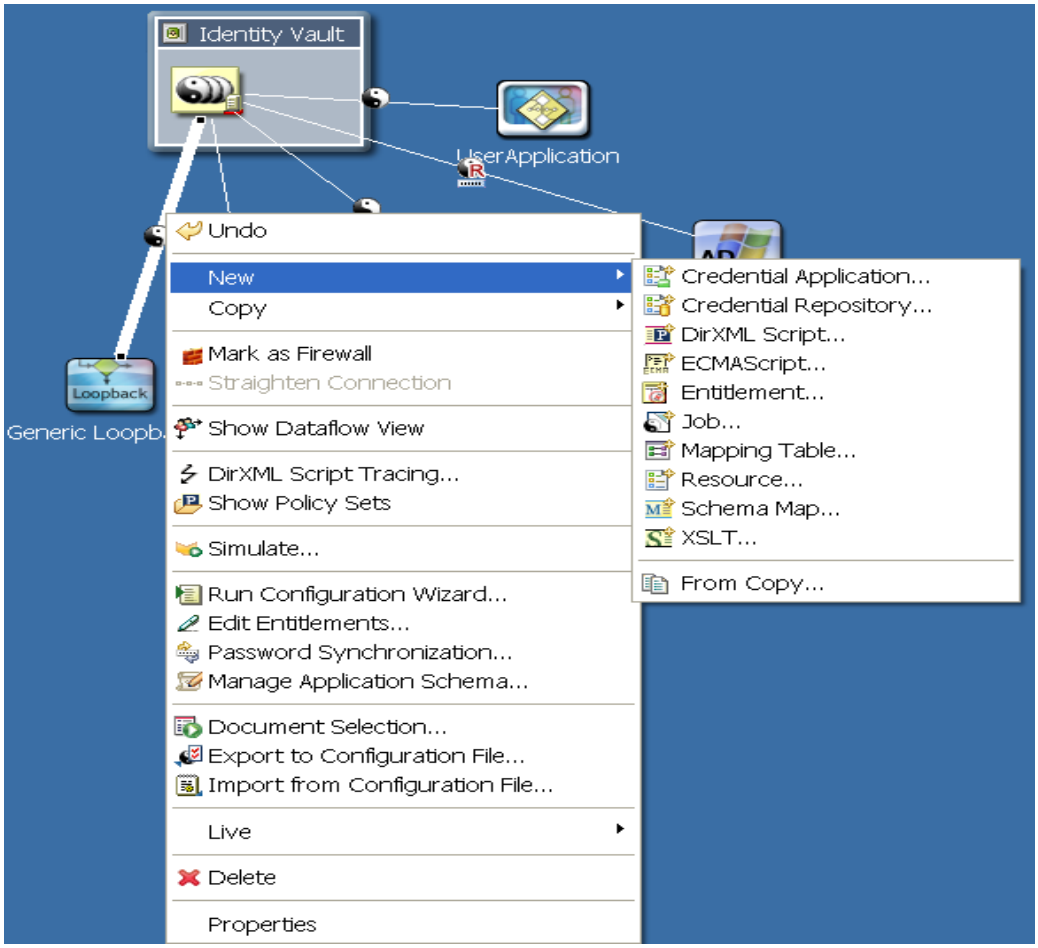


Table A-4 *Driver Operations*

Operation	Description
Undo Change Location	Returns an item to its previous status.
New > Credential Application	Creates a new Application object, which stores static single sign-on parameters for a specific application. For more information, see *Novell Credential Provisioning for Identity Manager 3.6 (http://www.novell.com/documentation/idm36/policy_credprov/data/bookinfo.html)*.
New > Credential Repository	Creates a new Repository Object, which stores static configuration information for an authentication credential repository such as either Novell SecretStore or Novell SecureLogin. For more information, see the overview section in the *Novell Credential Provisioning for Identity Manager 3.6 (http://www.novell.com/documentation/idm36/policy_credprov/data/bookinfo.html)*.

Operation	Description
New > DirXMLScript	Launches the Policy Builder, creates a policy, and creates a new DirXML® Script. DirXML Script is the primary method of implementing policies in the Novell Identity Manager Metadirectory engine.
New > ECMAScript	Creates an ECMAScript object and opens the ECMAScript editor.
New > Entitlement	Launches the Entitlement Wizard and adds an entitlement to the selected driver. For more information, see Chapter 13, “Using Entitlements,” on page 299.
New > Job	Launches the Job Scheduler Wizard to create a job.
New > Mapping Table	Creates a Mapping Table object. A policy uses a Mapping Table object to map one set of values to another set of corresponding values.
New > Resource	Creates a Resource object. Resource objects (for example, generic, ECMAScript, mapping table, application, or repository resources) store information that drivers use. The information can be arbitrary data in any format (for example, XML or text).
New > Schema Map	Creates a schema map policy and launches the Schema Map editor. A schema map policy maps class names and attribute names between the Identity Vault namespace and the application namespace. The schema map policy is applied in both directions.
New > XSLT	Creates an XSLT policy. XSLT is a standard language for transforming XML documents. You can use the XSLT option to implement policies as XSLT style sheets.
New > From Copy	Creates a policy by copying from an existing object.
Copy > Settings	Copies data from the selected driver to a target driver. A pasted copy overwrites data in the target driver.
Copy > Server-Specific Settings	Copies data from the selected server to a target server. A pasted copy overwrites data in the target server.
Mark/Unmark as Firewall	Enables you to mark where a driver is communicating through a firewall. Used in Developer mode. If driver icons are turned off, the firewall icon doesn't appear.
Straighten Connection	Straightens a driver connection line. If a line is not within a few degrees of being horizontal or vertical, this option is dimmed.
Show Dataflow View	Displays the flow of information between the application and the driver in the Developer view. Launches the Dataflow view. For more information, see Chapter 7, “Managing the Flow of Data,” on page 167.
Dataflow	Displays the dataflow between the application and driver set. Appears only when dataflow view is activated.
Simulate	Runs the Simulate Policy Transformation program against the selected driver.

Operation	Description
Show Policy Sets	Launches the Policy Set and Policy Flow views. For more information, see Policy Set View (http://www.novell.com/documentation/designer30/designer_intro/data/projpolicysetview.html) in the *Understanding Designer for Identity Manager*.
Run Configuration Wizard	Guides you through creating a driver. After you fill out the wizard's forms, Designer automatically generates policies that configure the driver to function as described in the forms.
Edit Entitlements	Enables you to select an entitlement that is associated with the driver and edit the entitlement's settings. For more information, see Chapter 13, "Using Entitlements," on page 299.
*Password Sync*hronization	Configures and displays the flow of password synchronization. For more information, see Section 7.6, "Integrating Passwords," on page 191.
Manage Application Schema	Enables you to manage a copy of the connected system's schema. You can make changes to a copy of the application schema so that you can test the Identity Manager drivers in Designer. See Section 6.7, "Managing a Copy of an Application Schema," on page 160.
Document Selection	Launches the Document Generation Wizard, which documents the selected driver.
Export to Configuration File	Exports the driver to a `.xml` file. iManager can consume this format, and Designer can re-import it. For more information, see Section 15.10, "Exporting To a File," on page 357.
Import From Configuration File	Imports an exported `.xml` driver file.
Live > Import	Enables you to connect to a server, browse to and select a driver, and import the objects into the driver.
Live > Deploy	Prepares a deployment summary and then deploys selected objects and attributes.
Live > Compare	Compares selected drivers. Enables you to reconcile or update drivers. See Section 15.7, "Using the Compare Feature When Deploying," on page 348.
Live > Driver Configuration > Import Attributes	Imports attributes from an existing driver.
Live > Driver Configuration > Deploy Attributes	Deploys the modified or imported attributes.
Live > Driver Configuration > Compare Attributes	Allows you to compare the attributes of a policy to the attributes that are already deployed.
Live > Refresh Application Schema	Specifies the server on an eDirectory tree where the schema is refreshed after an application's schema changes. See Section 6.7.2, "Refreshing the Application Schema," on page 162.
Live > Status for All Drivers	Reports whether the driver is stopped or running.
Live > Start Driver	Starts the driver.

Operation	Description
Live > Stop Driver	Stops the driver.
Live > Set Driver Trace Level	Specifies how much information to display in a trace level log from the driver. Settings go from 0-5.
Live > Restart Driver	Restarts the drivers.
Live > Set Up Driver Security	Launches the Driver Security Equals/Exclusions dialog box. Enables you to configure the selected driver's security equivalences and to exclude selected users from administrative roles. If you select multiple drivers, this dialog box lets you add, modify, and remove common security equivalences and exclusions of the selected drivers.
Delete	Deletes the selected driver and its policies.
Properties	Launches the driver's property pages. Enables you to configure the driver.

A.5 Application Operations

The following figure illustrates Modeler operations that are available when you right-click an application.

Figure A-5 *Application Operations*

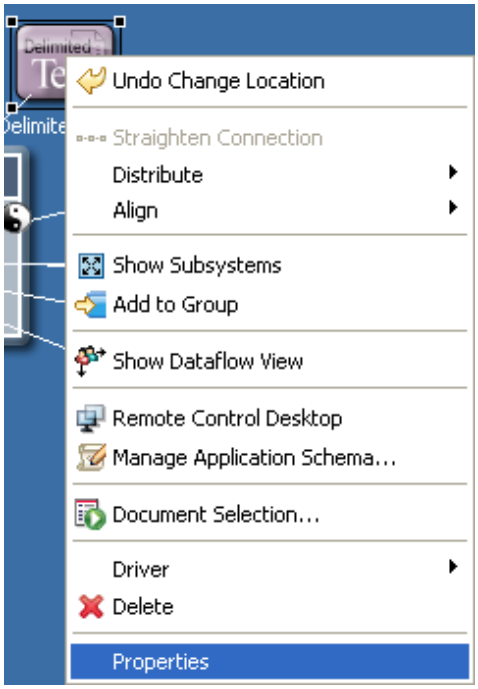


Table A-5 *Application Operations*

Operation	Description
Undo Change Location	Returns an item to its previous status.
Disconnect eDir-to-eDir	(Viewable when you select an eDir-to-eDir application) Separates the eDir-to-eDir application into two eDirectory drivers.
Straighten Connection	Straightens a driver connection line. If a line is not within a few degrees of being horizontal or vertical, this option is dimmed.
Distribute	Evenly distributes applications vertically or horizontally. Press Ctrl, select the items that you want to distribute, then select a pattern.
Align	Aligns the selected objects horizontally and vertically. For more information, see Table A-6 on page 528.
Change to eDirectory Tree	(Viewable when you select an eDirectory application.) Runs the Driver Configuration Wizard to install an eDir-to-eDir driver. Places a tree icon in the Identity Vault.
Change to Identity Vault/Meta-Directory	(Viewable when you select an eDirectory application.) Runs the Driver Configuration Wizard to install an eDir-to-eDir driver. Places a vault con in the Identity Vault.
Show/Hide Subsystems	Lets you model an application's or operating system's subsystems. For example, if you have a Linux system, you can open it and drop MySQL* inside as a subapplication that runs on Linux. This is for diagramming purposes only, but can be convenient for accurately capturing the structure of the enterprise systems around which you are building the identity solution.
Add to Group	Creates a Domain Group, and adds the selected items to it. The selected items are removed from any group that they were previously associated with.
Show Dataflow View	Displays the flow of information between the application and the driver in the Developer view. Launches the Dataflow view and lists Dataflow on the menu. For more information, see Chapter 7, "Managing the Flow of Data," on page 167.
Remote Control Desktop	Launches a remote control session for the selected application. The host server must have an existing VNC server running.
Manage Application Schema	Enables you to manage a copy of the connected system's schema. You can make changes to a copy of the application schema so that you can test the Identity Manager drivers in Designer. See Section 6.7, "Managing a Copy of an Application Schema," on page 160.
Document Selection	Launches the Document Generation Wizard, which documents the application.
Driver > Simulate	Runs the Simulate Policy Transformation program against the selected driver.

Operation	Description
Driver > Show Policy Sets	Launches the Policy Set and Policy Flow views. For more information, see Policy Set View (http://www.novell.com/documentation/designer30/designer_intro/data/projpolicysetview.html) in the *Understanding Designer for Identity Manager*.
Driver > Run Configuration Wizard	Guides you through creating a driver. After you fill out the wizard's forms, Designer automatically generates policies that configure the driver to function as described in the forms.
*Driver > Password Sync*hronization	Configures and displays the flow of password synchronization. For more information, see Section 7.6, "Integrating Passwords," on page 191.
Driver > Document Selection	Launches the Document Generation Wizard, which documents the driver.
Driver > Export to Configuration File	Exports the driver to a `.xml` file. iManager can consume this format, and Designer can re-import it. For more information, see Section 15.10, "Exporting To a File," on page 357.
Driver > Import from Configuration File	Allows you to browse to and import a driver configuration file
Driver > Import	Enables you to connect to a server, browse to and select a driver, and import the objects into the driver.
Driver > Deploy	Prepares a deployment summary and then deploys selected objects and attributes.
Driver > Compare	Allows you to compare the information structure in Designer on an object, to the object that is deployed or running on an eDirectory server.
Driver > Driver Configuration > Import Attributes	Imports attributes from an existing driver.
Driver > Driver Configuration > Deploy Attributes	Deploys the modified or imported attributes.
Driver > Driver Configuration > Compare Attributes	Allows you to compare the attributes of a policy to the attributes that are already deployed.
Driver > Status for All Drivers	Reports whether the drivers are stopped or running.
Driver > Start Driver	Starts the driver.
Driver > Stop Driver	Stops the driver.
Driver > Set Driver Trace Level	Allows you to specify how much information you want to see in a trace level log from the driver. Settings go from 0-5.
Driver > Restart Driver	Restarts the drivers.
Driver > Set Up Driver Security	Launches the Driver Security Equals/Exclusions dialog box. Enables you to configure the selected driver's security equivalences and to exclude selected users from administrative roles. If you select multiple drivers, this dialog box lets you add, modify, and remove common security equivalences and exclusions of the selected drivers.

Operation	Description
Driver > Properties	Launches the driver's property pages. Enables you to configure the driver.
Delete	Deletes the application and driver.
Properties	Enables you to configure Identity Vaults, driver sets, drivers, and applications.

A.6 Submenus

Table A-6 *Align Submenu*

Operation	Description
Align Top	Aligns the top edge of the selected objects.
Align Bottom	Aligns the bottom edge of the selected objects.
Align Left	Aligns the left edge of the selected objects.
Align Right	Aligns the right edge of the selected objects.
Align Center	Horizontally aligns the centers of the selected objects.
Align Middle	Vertically aligns the middles of the selected objects.

Table A-7 *Arrange Applications Submenu*

Operation	Description
Arrangement Off	Disables a previously selected auto-arrangement method.
Box	Arranges application icons in a square around the driver set.
Circle	Arranges application icons in a circle around the driver set.
Half Circle	Arranges application icons in a semicircle around the driver set.
Star	Arranges application icons in a star around the driver set.
Fan Out - Bottom	Arranges application icons in a fan shape below the driver set.
Fan Out - Left	Arranges application icons in a fan shape to the left of the driver set.
Fan Out - Right	Arranges application icons in a fan shape to the right of the driver set.
Fan Out - Top	Arranges application icons in a fan shape above the driver set.
Expand/Contract	Expands or contracts the layout of the application icons. Selecting this option opens a dialog box from which you drag the slide in the *Factor* field to change the layout.

Table A-8 *Dataflow Submenu*

Operation	Description
Publish	Specifies that the Publisher channel is synchronized for the selected objects (uni directional sync from selected objects.) For more information, see Chapter 7, “Managing the Flow of Data,” on page 167.
Subscribe	Specifies that the Subscriber channel is synchronized on the selected objects (uni directional sync to selected objects.) For more information, see Chapter 7, “Managing the Flow of Data,” on page 167.
Sync	Specifies that both the Publisher and Subscriber channel are synchronized for the selected objects (bidirectional sync.) For more information, see Chapter 7, “Managing the Flow of Data,” on page 167.
Ignore from	Specifies that the selected objects ignore Subscriber channel synchronization. For more information, see Chapter 7, “Managing the Flow of Data,” on page 167.
Ignore to	Specifies that the selected objects ignore both Publisher and Subscriber channel synchronization. For more information, see Chapter 7, “Managing the Flow of Data,” on page 167.
Ignore both directions	Specifies that the selected objects ignore Publisher channel synchronization. For more information, see Chapter 7, “Managing the Flow of Data,” on page 167.

Table A-9 *Distribute Operations Submenu*

Operation	Description
Horizontal	Evenly spaces the selected objects horizontally.
Vertical	Evenly spaces the selected objects vertically.

A.7 Keyboard Support

The following table describes common keyboard shortcuts available in the Modeler.

Table A-10 *Shortcut Keys*

Keystroke	Description
/	Navigates to the item's next connection.
\	Navigates to the item's previous connection.
Delete	Deletes the selected item or line.
Left-arrow	Navigates left.
Right-arrow	Navigates right.

Keystroke	Description
Up-arrow	Navigates up.
Down-arrow	Navigates down.
<Alt>+Down-arrow	Navigates into a subgroup.
<Alt>+Up-arrow	Navigates out of a subgroup.
<Ctrl> + =	Zooms in.
<Ctrl> + -	Zooms out.
<Ctrl> + A	Selects all objects in the current project.
<Ctrl> + C	Copies the selected objects to the Clipboard.
<Ctrl> + F	Opens the Find dialog box for searching the project.
<Ctrl> + V	Pastes the Clipboard contents to the selected location.

Document Generator Core Support Templates

B

- Section B.1, “dgSection.xsl,” on page 531
- Section B.2, “dgFormat.xsl,” on page 532
- Section B.3, “idmConfig.xsl,” on page 535
- Section B.4, “idmUtil.xsl,” on page 536

B.1 dgSection.xsl

Table B-1 *dgSection.xsl*

Template	Description
match “/”	Main template that invokes all sub-templates. Users can override this template to create their own template behavior; however it is recommended that you override the Section.Content, Section.Body, or Section.Main.
Section.Sequence	Main template that invokes all sub-templates. Users can override this template to create their own template behavior; however it is recommended that you override the Section.Content, Section.Body, or Section.Main.
Section.Main	This section includes Section.Content and Section.Children.
Section.Content	This section includes Section.Title and Section.Body.
Section.Body	The body content of the section.
Section.ShowStyleAttributes	Describes the default way to display attributes when no template is defined. Parameters: border - border used for tables. The default value is 0.5pt solid black.
Section.Children	Inserts the child sections that are passed as a parameter into this template.
Section.PageLayout	Formats the page layout, including paper size, headers, page numbering, and so forth. The Section.Main template is called to insert the document into this layout.
Section.staticContent	This template formats the page layout, including paper size, headers, page numbering, and so forth. The Section.Main template is called to insert the document into this layout.
Section.Title	Creates a title block containing the appropriate title text and link.
Section.TitleText	Gets the text to be displayed for this section.

B.2 dgFormat.xsl

Table B-2 *dgFormat.xsl*

Template	Description
Format.Title	This template takes care of all the details involved in formatting a title block. Parameters: ◆ text - Text to display. ◆ id - id for linking to this title (such as from the table of contents). ◆ font - Font size to use. ◆ image - Image to show as a bullet. The auto value tries to determine an image based on the current element.
Format.FigureTitle	This template takes care of all the details involved in formatting a figure title block. Parameters: ◆ title - Title text. ◆ description - Description text.
Format.OutputTextArea	Formats parameter information returned from a text area control that can contain HTML tags. If there is no HTML prefix, then line breaks are inserted. Parameters: ◆ value - Value of the textarea to output.
Format.EnabledStatus	Shows the enabled image if the value is True. The disabled image shows only when the show Disabled parameter is set to True. Parameters: ◆ value - Enabled, True/False. ◆ showDisabled - Set to True if the disabled image should show when the value is False. The default value is False.
Format.Chechbox	Shows a check box image if the value is True, or an empty check box image otherwise. Parameters: ◆ value - Checked, True/False. ◆ default - Default value if "", False, or some other value other than True exists. The default value is False.

Template	Description
Format.PropertyRow	Shows a table property row with two columns, one for the name and one for the value. Parameters: ◆ propertyName - The property name. ◆ propertyValue - The property value. ◆ border - Border used for the table. The default value is 0.5pt solid black. ◆ disable-output-escaping - Disables output escaping on the output value, so you can pass escaped FO content. The default value is False. ◆ showEmpty - Show empty values. The default value is False.
Format.ContextRow	Shows a contextual row with related attribute. Use this inside a table. Parameters: ◆ text - Text to display. ◆ level - Level or indent. The default value is 1. ◆ href - Href value to link to another portion of the document. ◆ image - Image to show as a bullet. The auto value tries to determine an image based on the current element. ◆ show-page-ref - Show page reference; True/False. The default value is True.
Format.ShowBulletImage	Show a bullet image. Parameters: ◆ image - Image to show as a bullet. The auto value will try to determine an image based on the current element.
Format.XMLFigure	This template takes care of all the details involved in formatting a figure that shows XML content. Parameters: ◆ title - Title text. ◆ description - Description text. ◆ xml - XML data to show in the figure in text. You can also use a "." to get the current node and children. ◆ simple-format - If True, this shows the XML without text selecting. This can also be preferred if name space attributes need to be included or if the XML is not well-formed. The default value is False.
match "node()" mode "xml-to-text"	XML-to-text formatting function. Parameters: ◆ attr-name-color, attr-value-color

Template	Description
match "@*" mode "xml-to-text"	XML-to-text formatting function. Parameters: ◆ attr-name-color, attr-value-color
match "text()" mode "xml-to-text"	XML-to-text formatting function.
match "comment()" mode "xml-to-text"	XML-to-text formatting function. Parameters: ◆ comment-color
Format.ImageFigure	Formats a figure that shows an image for its content. Parameters: ◆ title - Title text. ◆ description - Description text.
Format.PageBreak	Inserts a page break.
Format.BasicLink	Creates a basic link to the given href using the given text. If the href is empty, it only outputs the text value. Parameters: ◆ text - Link text ◆ href - Link href
Format.BasicLinkToReferencedItem	Creates a basic link to the XSI referenced item. This uses the @guid attribute to build the link. If no @guid is available, only the text label is rendered. Parameters: ◆ xsiHref - XSI value of referenced node.
Format.Uppercase	Used to convert a string to uppercase text. Parameters: ◆ value - The value you want to convert to uppercase.
Format.SmartSpace	Used to convert a string to smart-spaced text. Parameters: ◆ value - The value you want to smart-space.
Format.OutputDebugParameters	Outputs the debug parameters for a section when the DEBUG_PARAMS attribute is enabled.
Format.Debug	Outputs the specified text in a debug format when the DEBUG attribute is enabled. Parameters: ◆ text - Debug text.

B.3 idmConfig.xsl

Table B-3 *idmConfig.xsl*

Template	Description
match "*" mode "xmlFigure"	Build an XML Figure for any policy type. Parameters: ◆ title, description, alwaysShowPolicyXmlSource
match "xsl:stylesheet|xsl:transform"	Filter. Parameters: ◆ title, description
match "attr-name-map"	Attribute mapping.
match "policy"	Policy matching. Parameters: ◆ title, description, alwaysShowPolicyXmlSource
opConcat	
opDelim	
match "@*" mode "DirXMLScript"	
match "*" mode "DirXMLScript"	
match "arg-actions" mode "DirXMLScript"	
match "arg-dn" mode "DirXMLScript"	
match "arg-value" mode "DirXMLScript"	
match "token-text" mode "DirXMLScript"	
getLabel	Utility method used to get policy related text labels. Parameters: ◆ name - The name of the label.
match "actions"	

B.4 idmUtil.xsl

Table B-4 *idmUtil.xsl*

Template	Description
IdmUtil.ItemPropertyTable	Shows a table of values for the current Item. Depending on the item, it might filter attributes. Parameters: ◆ title - Title text. ◆ description - Description text. ◆ showEmpty - Show empty values. The default value is False.
IdmUtil.StartOptionPropertyRow	Shows the appropriate icon and text for the startup option on the current Item. (0 = Disabled, 1 = Manual, 2 = Auto) Parameters: ◆ propertyName, propertyValue, border
IdmUtil.ItemNumbering	Gets the current item numbering in context to the Designer source (such as "2.4.5.2."). This template helps centralize what should be counted in the numbering process because several places reuse this information.
IdmUtil.ItemText	Based on the XSI type, returns text for the type, followed by a colon and the name value (for example, Identity Vault: my vault 1)
IdmUtil.ItemType	Returns text representing the type of the current Item (such as Identity Vault, Domain, or Driver Set)
IdmUtil.PolicySetPropertyRow	Builds a property row with a list of the policies based on the next policy value. Parameters: ◆ policy - Root policy of the policy set, passed by attribute name (such as mappingPolicy). ◆ label - Label for the displayed value. ◆ emptyLabel - Text to show if the value is empty. The default value is (none defined).
IdmUtil.PolicySetLinks	Returns a list of policy set links, called recursively. Parameters: ◆ xsiRootPolicyHref - Root policy of the policy set.

Template	Description
IdmUtil.ConfigValuesTable	Shows a Config Value table for the given XML. Parameters: ◆ title - Title text. ◆ description - Description text. ◆ xml - XML value to use to create the table. ◆ border - Border used for the table. The default value is 0.5pt solid black. ◆ emptyLabel - Text to show if the value is empty. The default value is (none defined).
IdmUtil.FilterTable	Shows a Filter table for the given XML. Parameters: ◆ title - Title text. ◆ description - Description text. ◆ xml - XML value to use to create the table. ◆ emptyLabel - Text to show if the value is empty. The default value is (none defined).
IdmUtil.showSyncIcon	Show an Identity Manager sync icon based on input type and sub-type. Parameters: ◆ type - Type is pub, sub. ◆ sub-type - Sub-type is "", sync, ignore, notify, reset.
IdmUtil.ValueOfReferencedItem	Returns the value of the node given the XSI expression. When extracting the name of an item, you should use the Format.BasicLinkToReferencedItem method so that the text is created as a link inside the document. Parameters: ◆ xsiHref - XSI value of referenced node. ◆ suffix - Suffix to append before selecting (the default is the current node). The default value is "."
IdmUtil.ItemCustomIconFileName	Get the custom icon filename for the given GUID. Parameters: ◆ guid - The item's GUID
IdmUtil.ShowIManagerIcon	This method is for backwards compatibility. Use IdmUtil.ShowIcon instead.

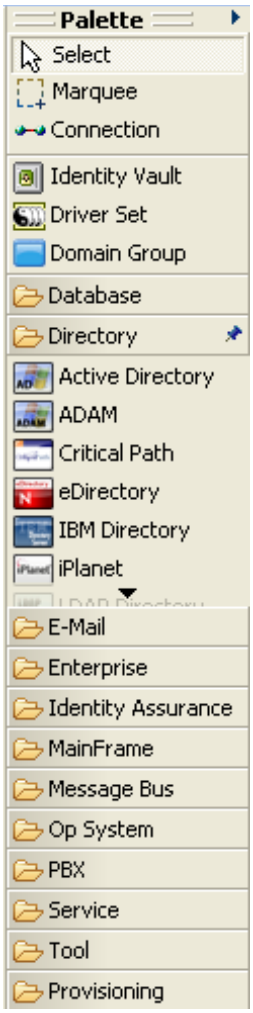
Template	Description
IdmUtil.ShowIcon	Shows the icon for the current Item. This first checks the cusomIconURI for a referenced image, then builds to a generic path based on the type attribute (for Drivers and Applications). Parameters: ◆ image-width - The image width to use. The default value is 49px.

Adding Applications and Drivers to the Palette

C

The following graphic illustrates Designer's palette. The Directory group is expanded to illustrate applications in that group.

Figure C-1 *Designer's Palette*



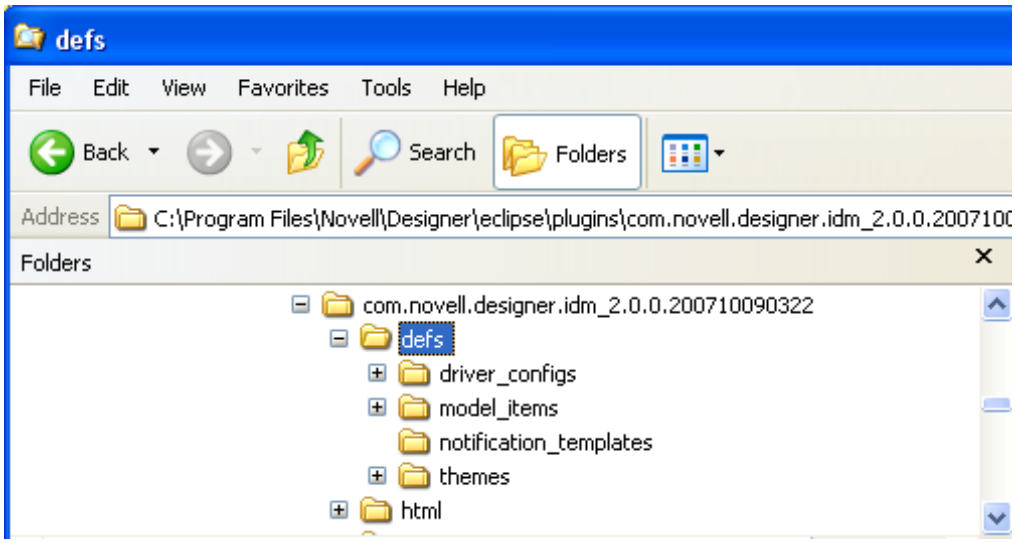
You can add a group, application, driver, or driver configuration to the palette. To do so, you must modify or create almost all the file types discussed in Section C.1, "Definition Folders and Files," on page 540. You must also exactly follow each step as explained in Section C.2, "Adding to the Palette," on page 546.

- Section C.1, "Definition Folders and Files," on page 540
- Section C.2, "Adding to the Palette," on page 546
- Section C.3, "Protecting Your Customized Files," on page 559

C.1 Definition Folders and Files

The palette definition is stored in the `com.novell.designer.idm` plug-in's `defs` folder.

Figure C-2 *The defs Folder*



The following sections provide information about subfolders and files:

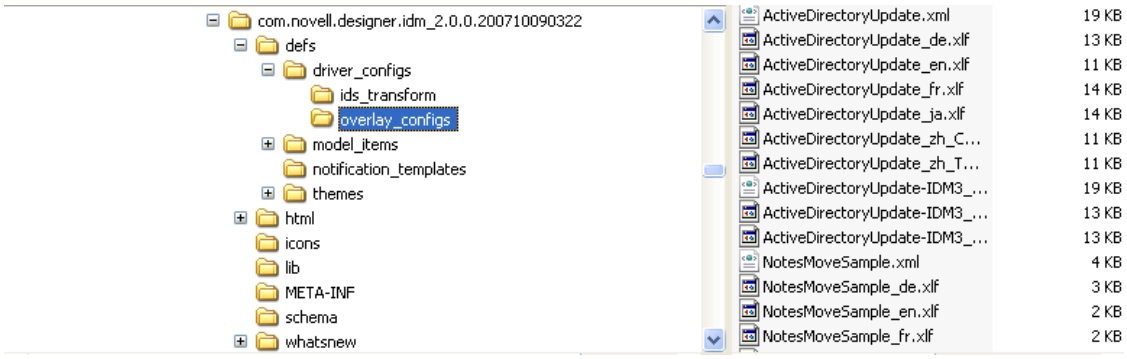
- Section C.1.1, "Driver Configuration and Localization Files," on page 540
- Section C.1.2, "Palette Folders and Files," on page 541
- Section C.1.3, "The Notification Templates Folder," on page 546
- Section C.1.4, "The Themes Folder," on page 546

C.1.1 Driver Configuration and Localization Files

The `com.novell.designer.idm/defs/driver_configs` folder contains all the driver configuration files and their localization (`.xlf`) files. These files contain Identity Manager policies. You can import them by using iManager or Designer.

The `overlay_configs` folder contains the driver overlay files.

Figure C-3 *The overlay_configs Folder*

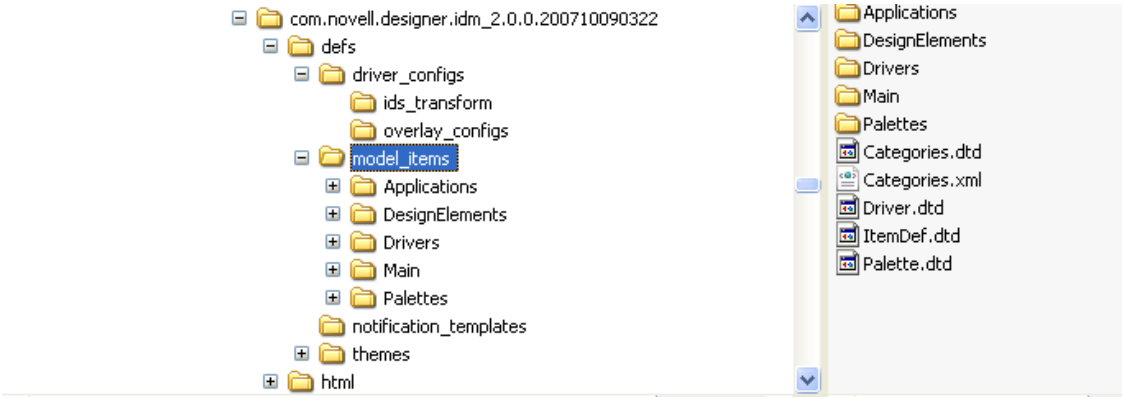


The ids_transform subdirectory should be left alone.

C.1.2 Palette Folders and Files

The com.novell.designer.idm/defs/model_items folder contains all the items that make up the palette:

Figure C-4 *The model_items Folder*



The following table lists the .xml and .dtd files found in this folder. The .dtd files contain the XML Document Type Definition for the different palette definition files.

Table C-1 *Files in the model_items Folder*

Filename	Description
Categories.dtd	Defines a category
Categories.xml	Contains all the categories that the palette can consume. Because adding or removing categories breaks existing code or has no impact at all, this file should be left alone.
Driver.dtd	Elements that make up a driver (for example, configuration files, primary and secondary applications, icons, and capabilities)
ItemDef.dtd	Defines applications and design elements
Palette.dtd	Defines the palette's name and its groups

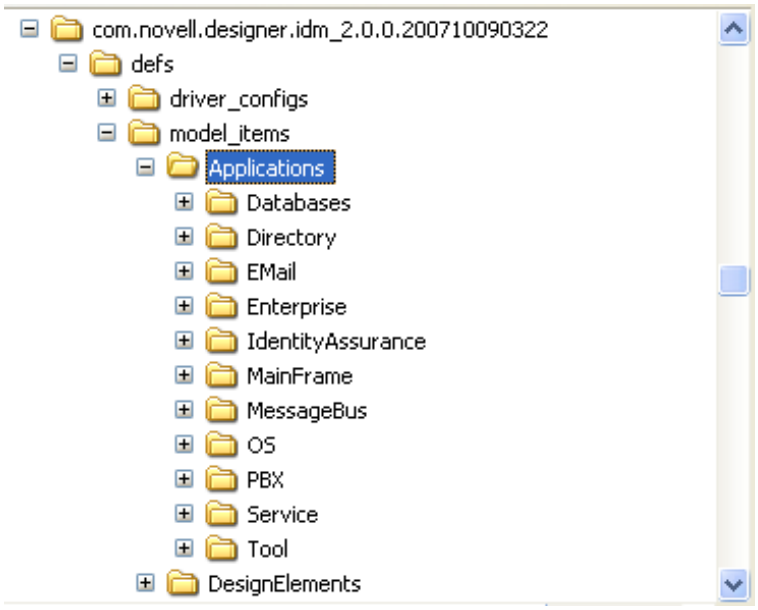
The model_items folder also contains several subfolders:

- “Definition Files for Applications” on page 541
- “Design Elements” on page 546
- “The Drivers Folder” on page 546

Definition Files for Applications

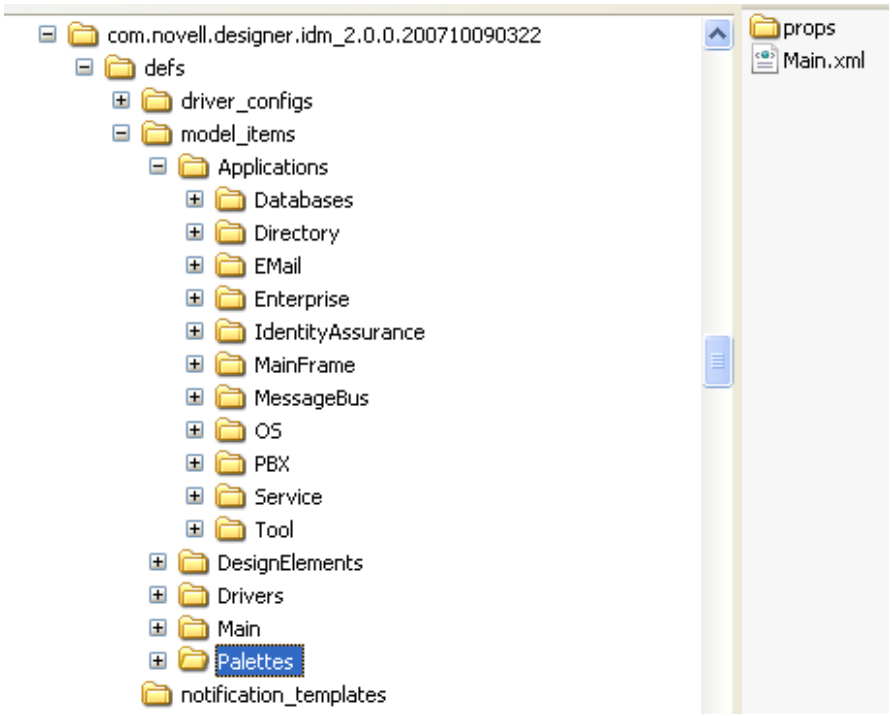
The Applications folder contains definition files for all applications that are available in the palette.

Figure C-5 *The* `Applications` *Folder*



The application definition files are grouped into folders that match the groups defined in the `Main.xml` palette definition file in the `Palettes` folder. The palette arranges applications in these same groups in Designer.

Figure C-6 *The Palettes Folder*



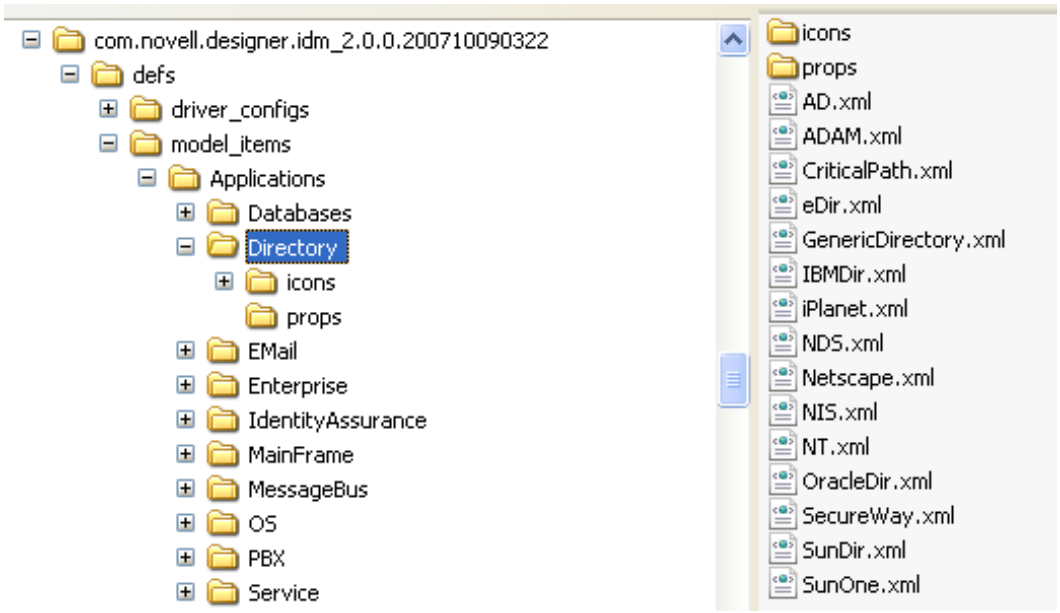
The `Applications/Directory` folder contains XML files, icons, and localization variables in properties files.

- "XML Files" on page 543
- "The Icons Folder" on page 544
- "Localization Files" on page 545

XML Files

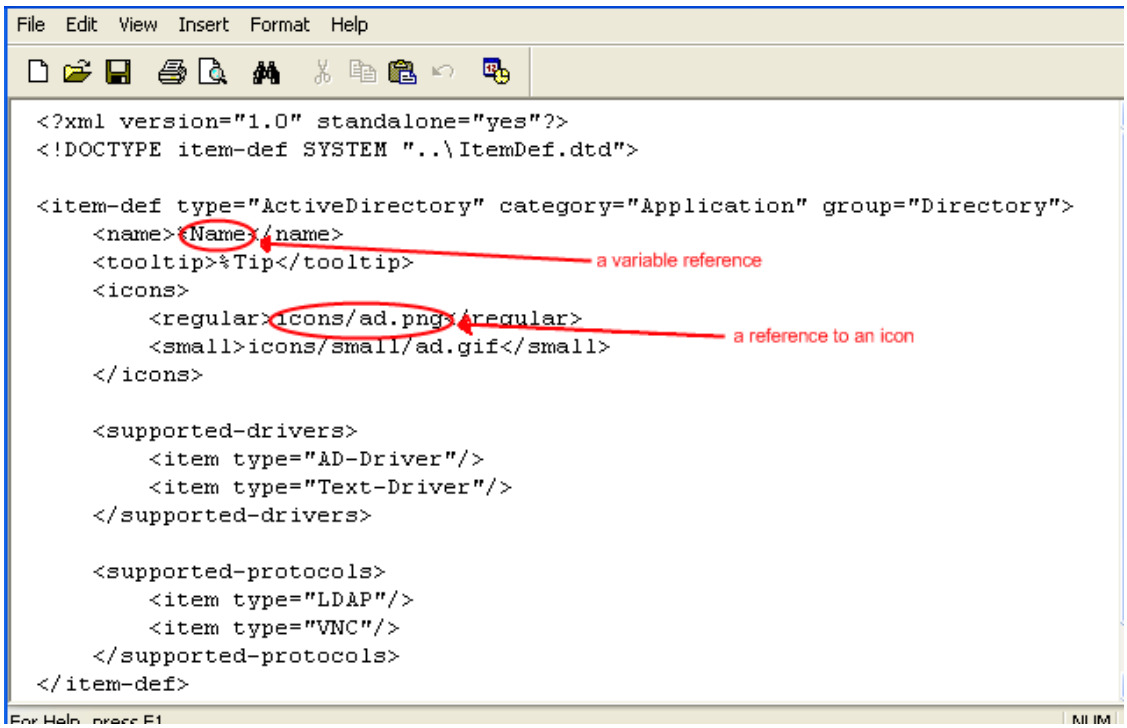
The `defs/model_items/Applications/Directory` folder contains `.xml` files. These files are the application definitions.

Figure C-7 *XML Files in the Directory Folder*



As the following figure illustrates, the definition files reference icons and localization variables:

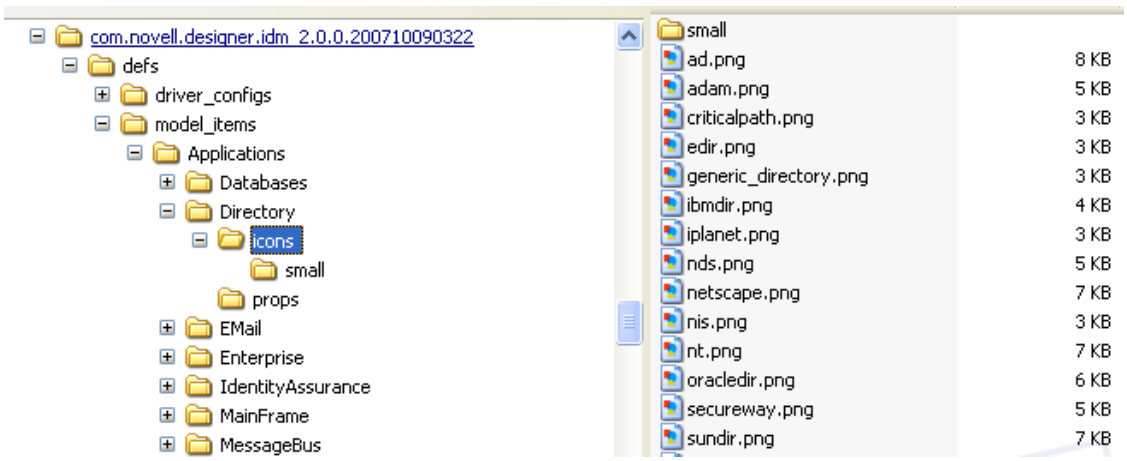
Figure C-8 *The AD.xml File*



The Icons Folder

The `defs/model_items/Applications/Directory/icons` folder contains icons in PNG format (`.png`).

Figure C-9 *The `icons` Folder*

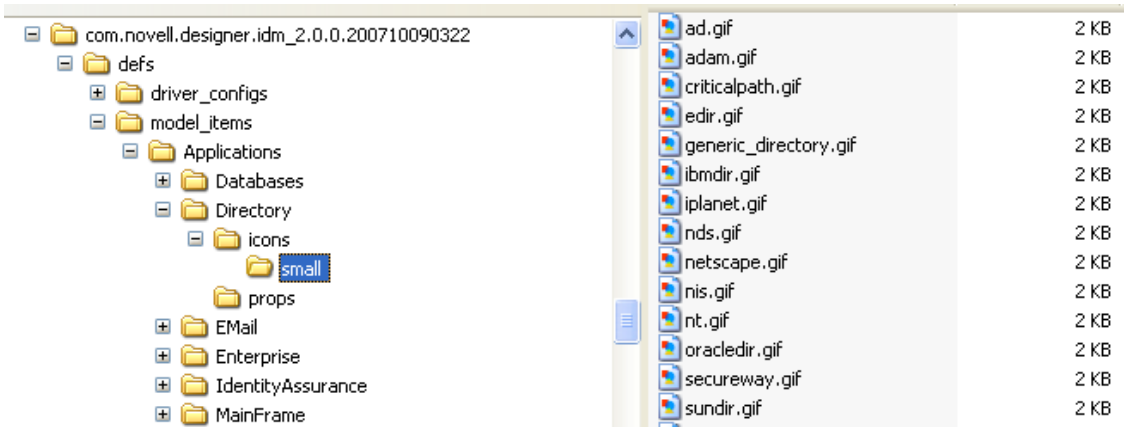


The Modeler (not the palette) uses these icons. The palette uses the small icons in the `small` subdirectory. The .png files are referenced from the application definition files.

The icons are 44x55 pixels in size and use transparency to display well in the Modeler.

The `small` folder contains smaller GIF versions of the icons in the parent directory.

Figure C-10 *The small folder*

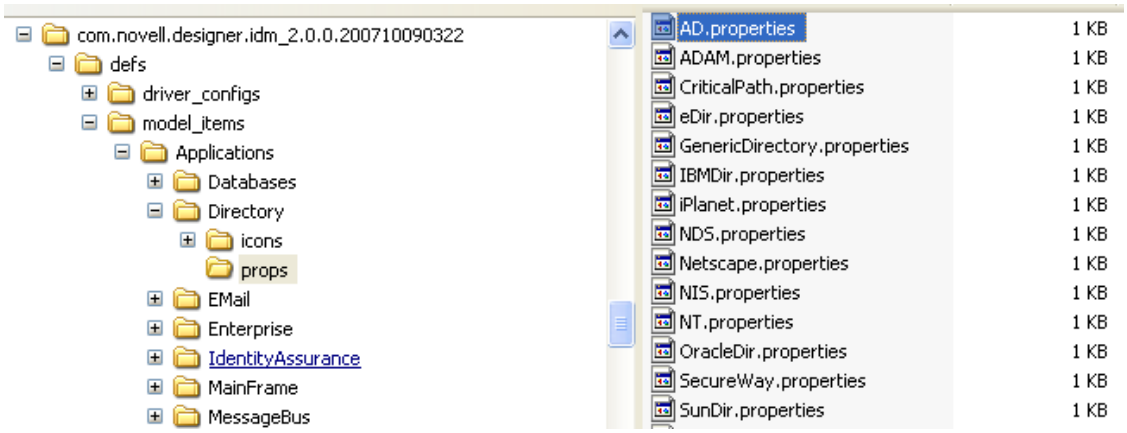


These icons are actually shown in the palette. The icons are 20x16 pixels in size and use transparency to display well in the palette.

Localization Files

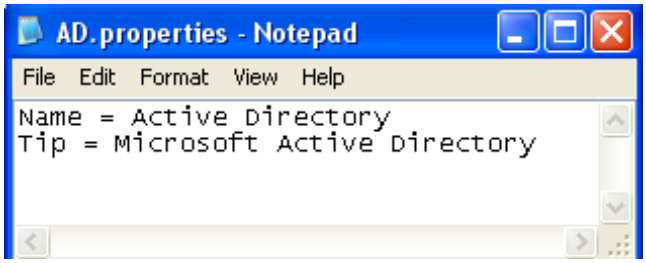
The `defs/model_items/Applications/Directory/props` folder contains localization variables that are defined in `.properties` files.

Figure C-11 *The `props` Folder*



The application definition files reference localized strings in the `.properties` files. In the following figure, *Name* can be referenced through %Name, illustrated in Figure C-8 on page 544.

Figure C-12 *A `.properties` File*



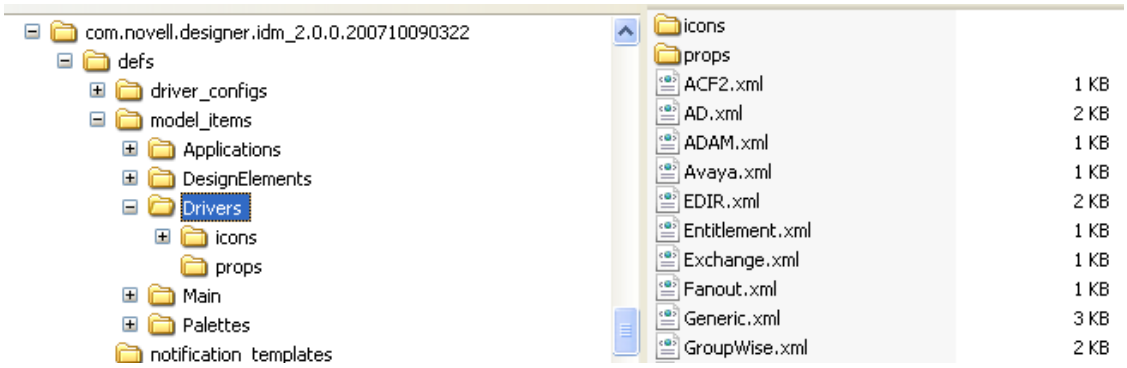
Design Elements

The `defs/model_items/DesignElements` folder mirrors the `Applications` folder, but contains design elements instead of applications. Design elements are like unknown applications to Designer. Design elements can be connected to and from anything, but Designer does not do anything with them. They have only a generic properties page, and no logic exists around them. They are basically just icons.

The Drivers Folder

The `defs/model_items/Drivers` folder contains the driver definition files (not the driver configuration files that contain Identity Manager policies and can be imported by using iManager or Designer).

Figure C-13 *The Drivers Folder*



The `icons` and `props` folders serve the same purpose as explained in "The Icons Folder" on page 544 and "Localization Files" on page 545.

C.1.3 The Notification Templates Folder

The `defs/notification_templates` folder contains the default e-mail notification templates that ship with Designer.

C.1.4 The Themes Folder

The `defs/themes` folder contains the Modeler theme definition files that ship with Designer.

C.2 Adding to the Palette

The need to extend the default palette usually arises when additional driver configuration files need to be hooked up to existing applications or to new applications or drivers.

Adding to the palette is a very delicate process and only successful if followed exactly step by step. Each step needs to be adapted to your situation.

- Section C.2.1, "Copying Configuration Files," on page 547
- Section C.2.2, "Creating the Group," on page 547
- Section C.2.3, "Adding a Key_Value Pair," on page 548

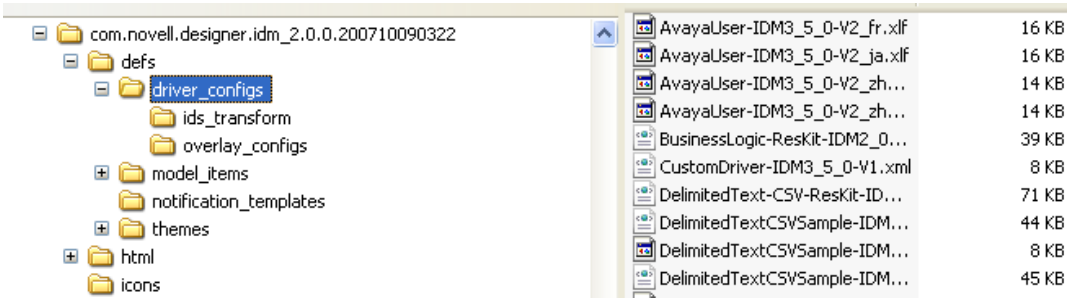
- Section C.2.4, “Creating a Driver Definition,” on page 549
- Section C.2.5, “Creating the Application,” on page 552
- Section C.2.6, “Hooking Up the Custom Application,” on page 555

C.2.1 Copying Configuration Files

1 Copy the new driver configuration file into the `driver_configs` folder so that the configuration file is accessible (but not yet hooked up) from Designer.

In this example, the new driver configuration file is `CustomDriver-IDM3_5_0-V1.xml`.

Figure C-14 *An Example New Driver Configuration File*



2 Copy into the `driver_configs` folder all corresponding `.xlf` files that belong to `CustomDriver-IDM3_5_0-V1.xml`.

The configuration file is now available and ready to be hooked up.

C.2.2 Creating the Group

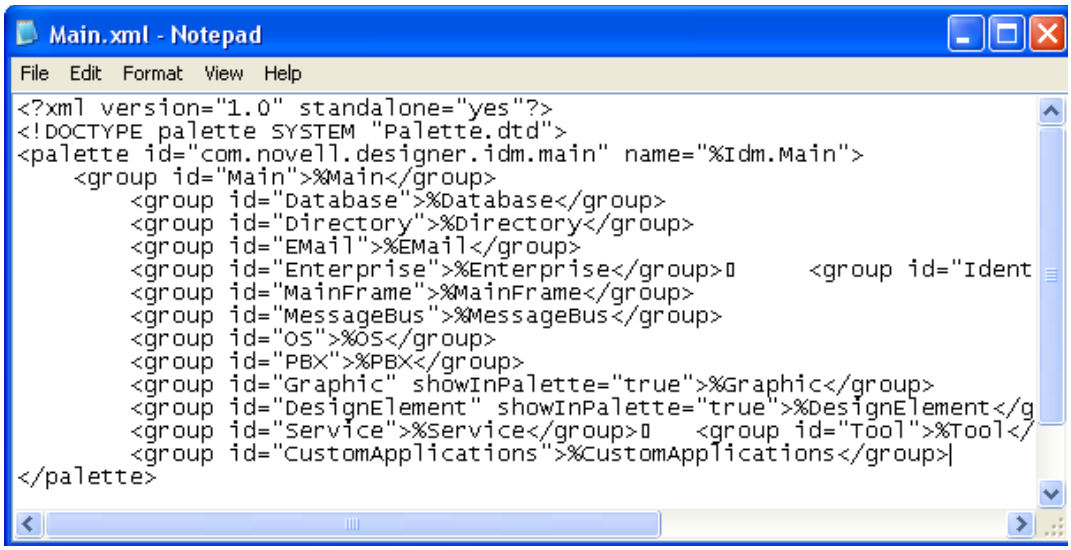
Before you place the new application Custom Application into the new Custom Applications group, you must first create the group.

1 Decide on the name of the new application that you want to create and the group that you want the new application to go into.

For this example, the following names are used:

- New application: Custom Application
- New group: Custom Applications

2 Add a group element to the `defs/model_items/Palettes/Main.xml` file.



Give the group element an ID attribute with an intuitive and unique value (for example, CustomApplications). Set the value of the element to %CustomApplications to make it localizable.

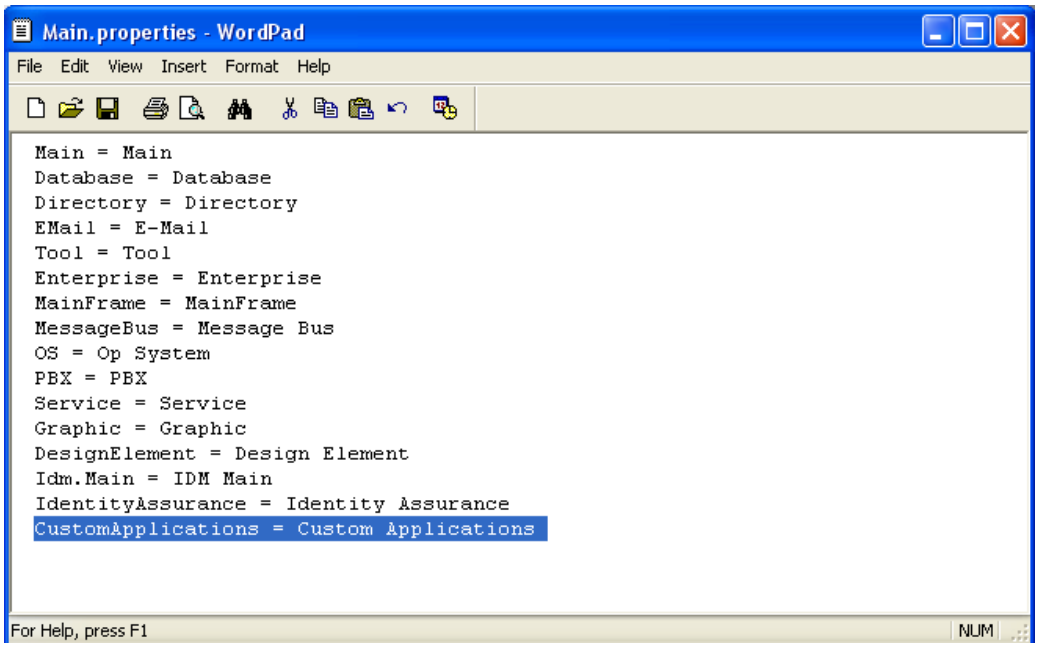
3 Save the file.

C.2.3 Adding a Key_Value Pair

1 Open `defs/model_items/Palettes/props/Main.properties`.

This is the properties file for the `Main.xml` file that you edited in Step 2 on page 547.

2 Add a key/value pair (for example, CustomApplications = Custom Applications).

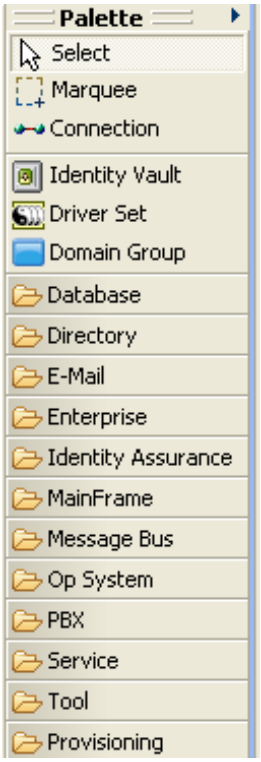


3 Save the file.

4 If you want to localize the group name into other languages, copy the properties file and rename it to `Main_language_code.properties`.

   For an example of supported languages and their codes, view the `.xlf` files in the `defs/driver_configs` folder.

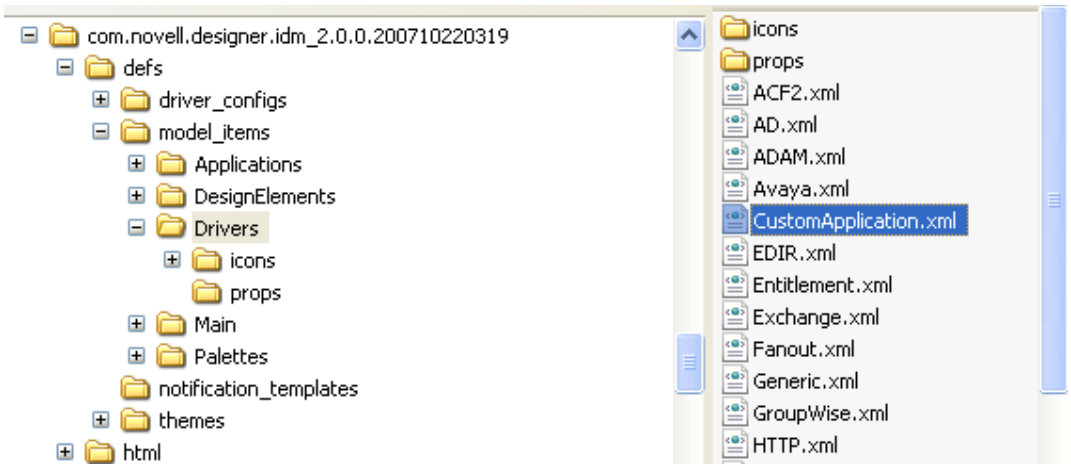
5 View the new group as an empty group in the palette by starting the copy of Designer that you are manipulating.



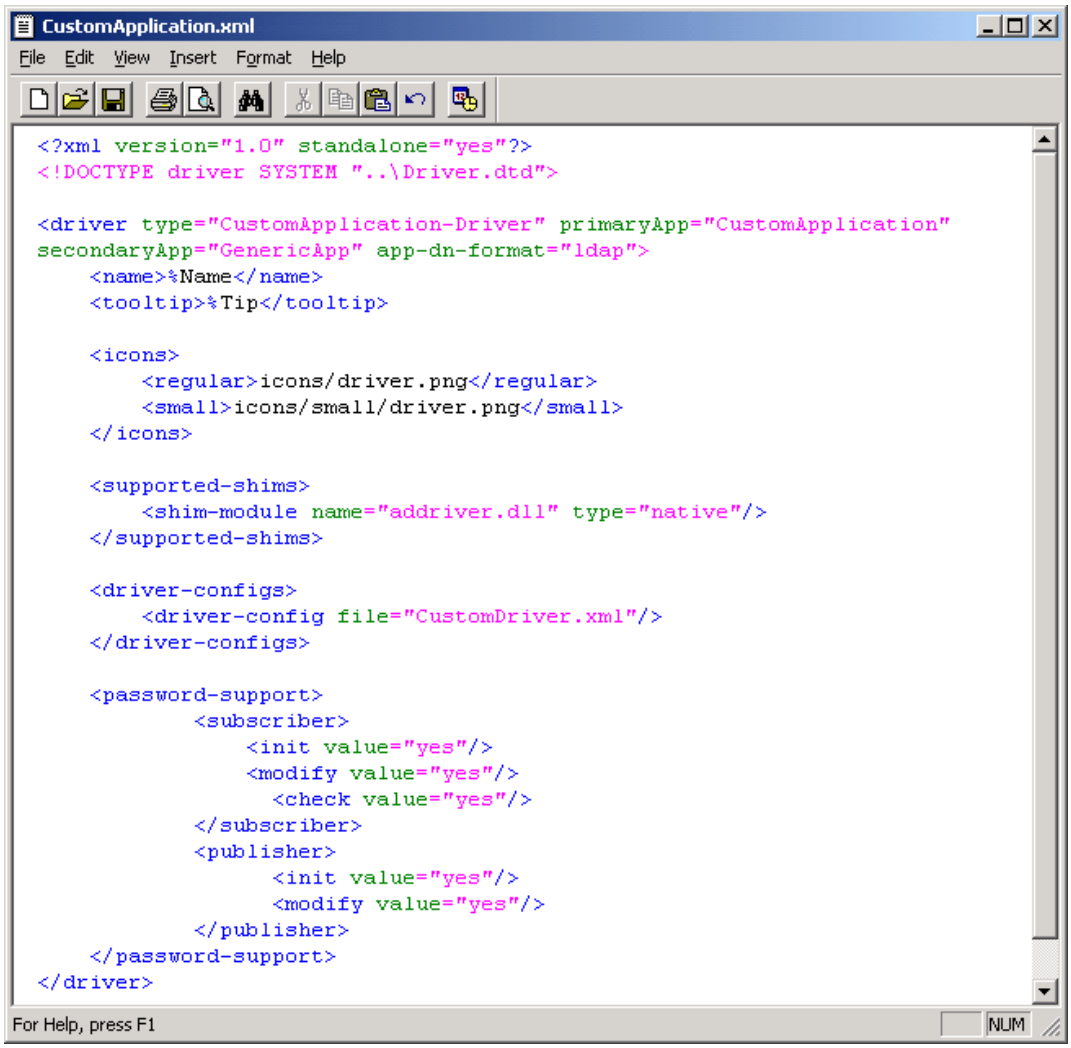
C.2.4 Creating a Driver Definition

1 Create a driver configuration file `CustomApplication.xml` in the `defs/model_items/Drivers` folder.

   The new configuration file must follow the `Driver.dtd` specifications in the folder that you just created. The easiest way to do this is to copy an existing driver definition file, rename the file, then modify it.



2 Edit the configuration file.



2a Provide an intuitive and unique type (for example, CustomApplication-Driver).

2b Set the primaryApp value to CustomApplication.

2c Set secondaryApp value to GenericApp.

2d Specify the app-dn-format that your application supports.

2e Leave the icons as they are. They are not driver-specific.

2f Specify the shims that your application supports.

2g Specify the driver configuration file to use for this driver.

Specify only the filename, without the versioning information.For example, if your driver configuration file is named `CustomDriver-IDM3_5_0-v1.xml`, you refer to it as `Custom.xml`).

Because Designer 2.0 M5 and later releases hide or display the user interface and features based on the version of the engine that you are working on, driver configuration filenames are important. You need to store the version information in the configuration filename, according to a well-defined format:

base name[-*type*]-*idm engine version-configuration file version*`.xml`

Examples:

- `ActiveDirectory-Mirror-IDM3_0_1-V9.xml`
- `ActiveDirectory-Flat-IDM3_5-V3.xml`
- `SAP-HR-IDM2_0_2-V2.xml`
- `SAP-User-IDM3_0_1-V1.xml`
- `SAP-User-IDM3_0_1-V2.xml`

In the example filenames, the IDM element identifies the engine version. The IDM elements to date are the following:

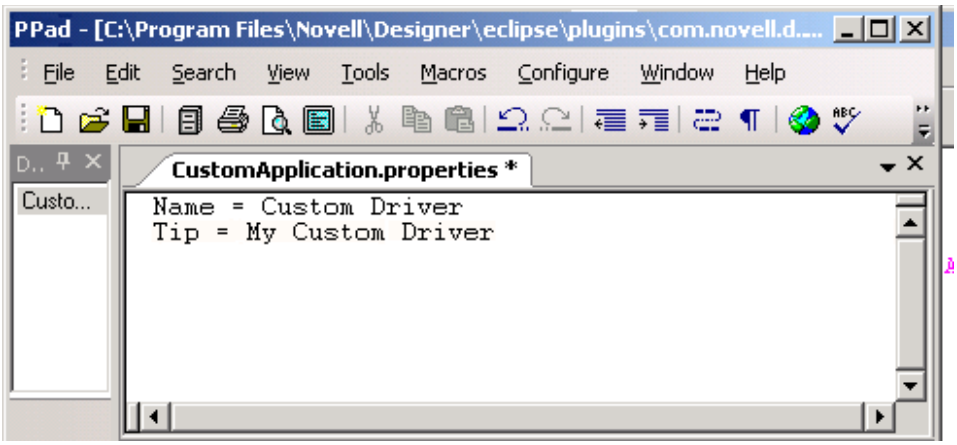
- IDM2_0
- IDM2_0_1
- IDM2_0_2
- IDM3_0
- IDM3_0_1
- IDM3_5
- IDM3_6

The V element in the example filenames specifies the version of this particular configuration file. It is a number that is incremented with each release of a new configuration file version. The following are examples:

- V1
- V9
- V11

No requirement exists for a more complex numbering schema.

3 Modify the `props/CustomApplication.properties` localization file.



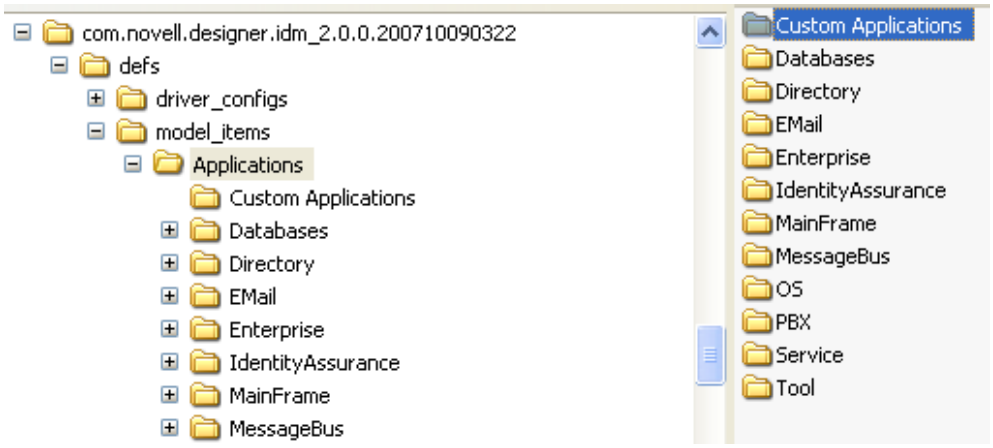
You might need to create this file. If so, the quickest way is to copy, rename, and edit the file.

C.2.5 Creating the Application

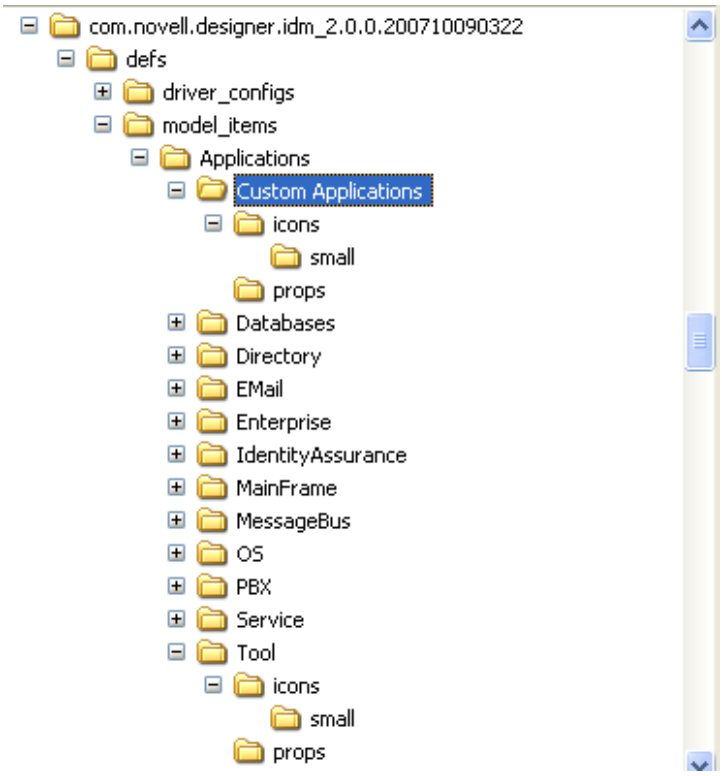
The next step is to create the Custom Application application and place it in the new Custom Applications group.

1 Create a folder in the `defs/model_items/Applications` directory.

Name the folder the same name as the group ID. In this example, the name is CustomApplications, as specified in Step 2 on page 547.



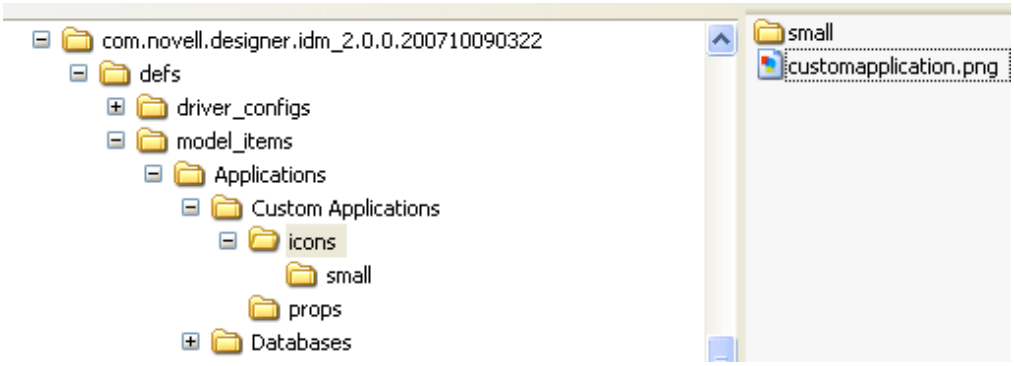
2 Create `icons`, `icons/small`, and `props` folders in the `CustomApplications` folder.



3 Create icons for the application.

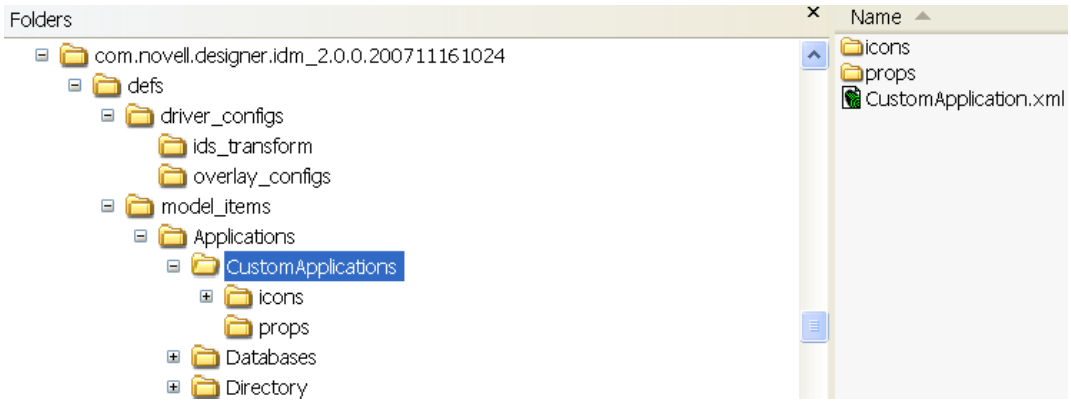
You can copy existing icons and modify them so the transparency is correct. In this example, modify the existing Generic Application icons.

3a Copy `defs/model_items/Applications/Tool/icons/generic_app.png` to `defs/model_items/Applications/CustomApplications/icons`. Rename the file as `customapplication.png`.



3b Copy `defs/model_items/Applications/Tool/icons/small/generic_app.gif` to `defs/model_items/Applications/CustomApplications/icons/small`. Rename the file as `customapplication.gif`.

4 Create an application definition file (`.xml`) in the `defs/model_items/Applications/CustomApplications` folder.

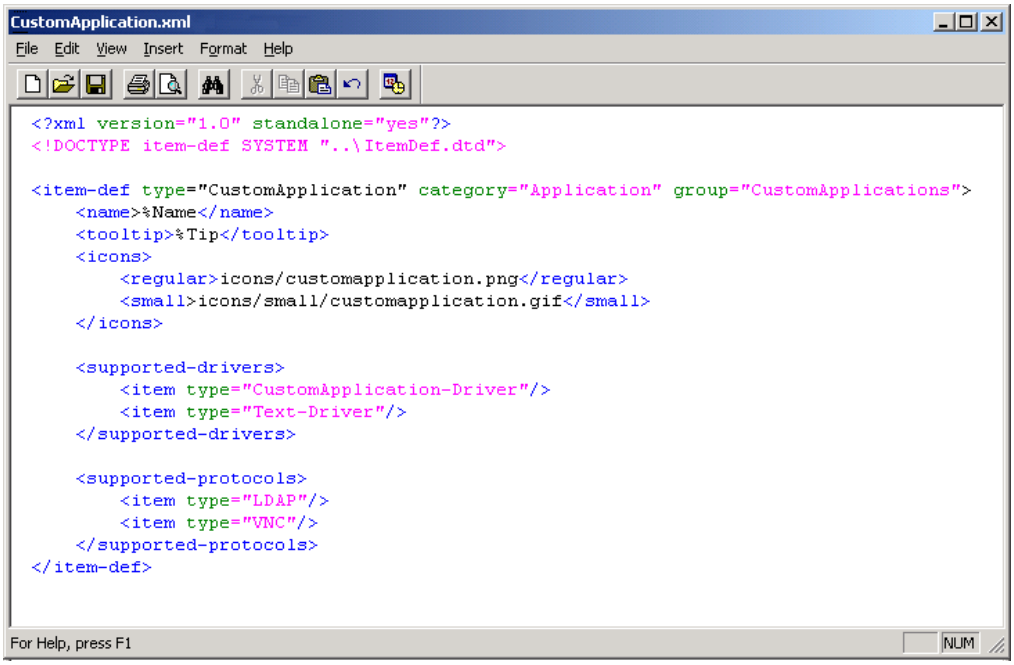


The definition file follows the `ItemDef.dtd` specifications. (See Table C-1 on page 541.)

The easiest way to create the file is to copy an existing application definition file (for example, `GenericApp.xml`), rename the file, then modify it.

NOTE: The application definition file and the `.properties` file created in Step 5 need to have the same name as the type. In this example, the files are named `CustomApplication.xml` and `CustomApplication.properties`.

4a Make sure that the type attribute of the item-def element is set to an intuitive and unique name (for example, CustomApplication).



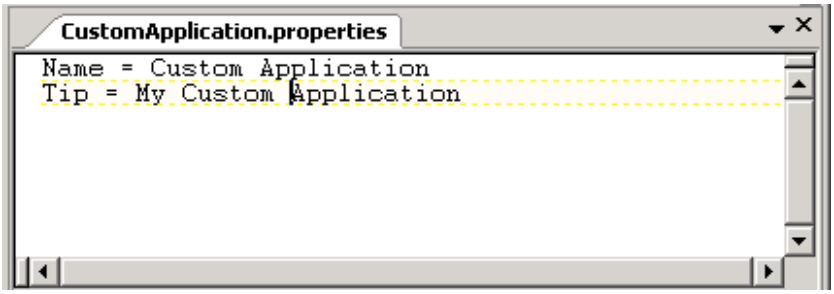
4b Leave the category attribute as *Application* and set the group attribute to the group ID, which is *CustomApplication*.

Reference the icons as you named them and do the same for the supported drivers. In this example, the Delimited Text Driver (Text-Driver) is added as an alternative to the Custom Application Driver (CustomApplication-Driver).

If the application can be connected to by using LDAP or VNC, leave these supported-protocols in. Otherwise, remove them. Usually, every application runs on a host OS that supports either one or both of the protocols. Having these protocols registered enables certain functionality in Designer for that application.

5 Modify the `props/CustomApplication.properties` localization file in the same way you modified the group.

An easy way is to copy `defs/model_items/Applications/Tool/props/GenericApp.properties` to `defs/model_items/Applications/CustomApplications/props`. Rename the file to `CustomApplication.properties`, then modify and save the file.



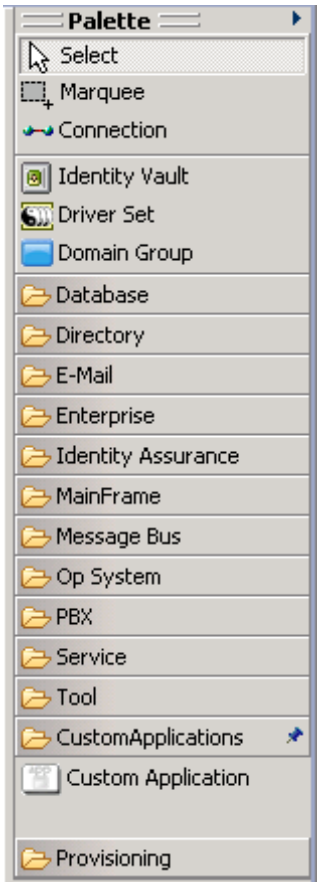
6 Copy the `.gif` icon file into the `com.novell.designer.core/icons/iManager` directory.

This icon is used in iManager after the driver is deployed into the Identity Vault.

C.2.6 Hooking Up the Custom Application

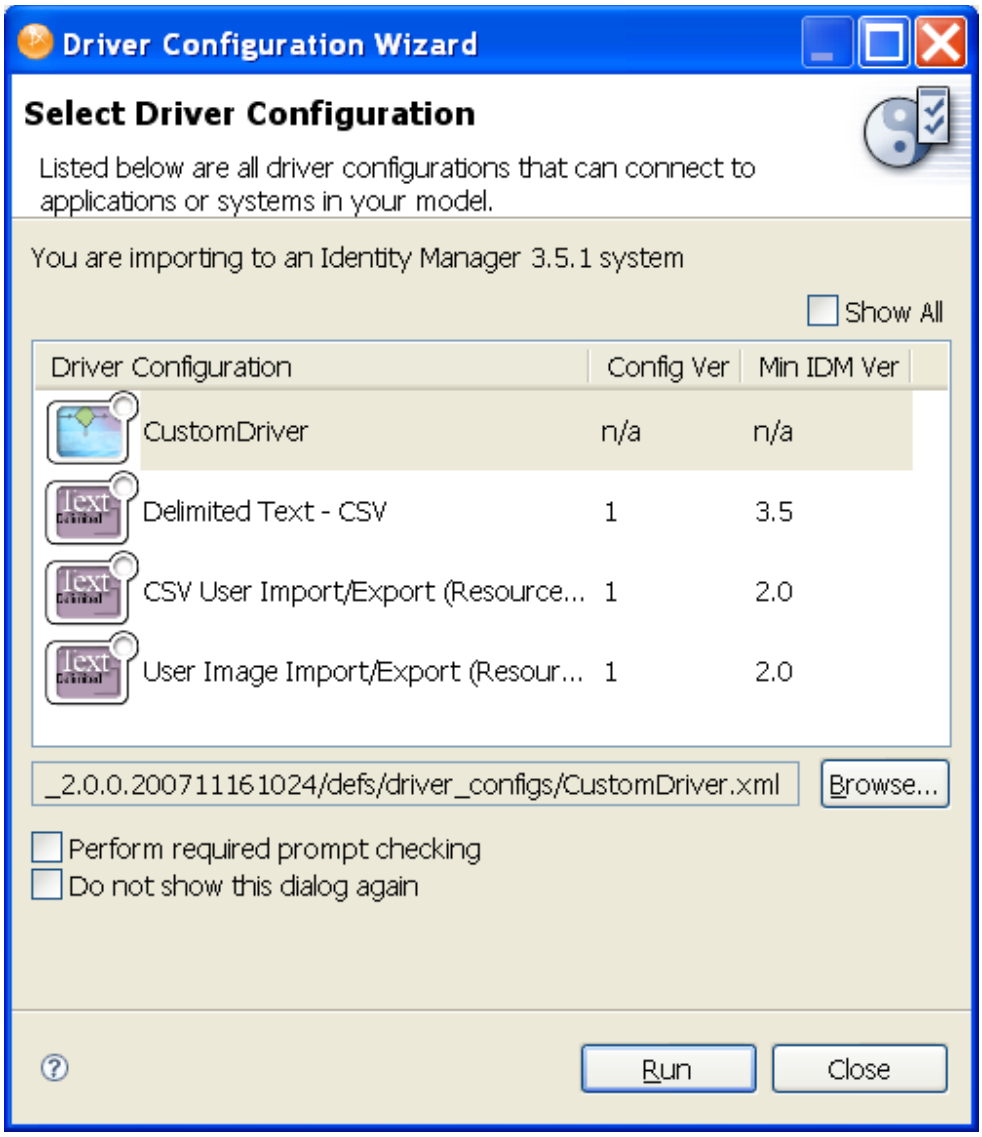
1 Run Designer.

The new Custom Application appears in the new Custom Applications group in the palette.



If you drag and drop the application to the Modeler workspace, the Driver Configuration Wizard prompts you to import the following:

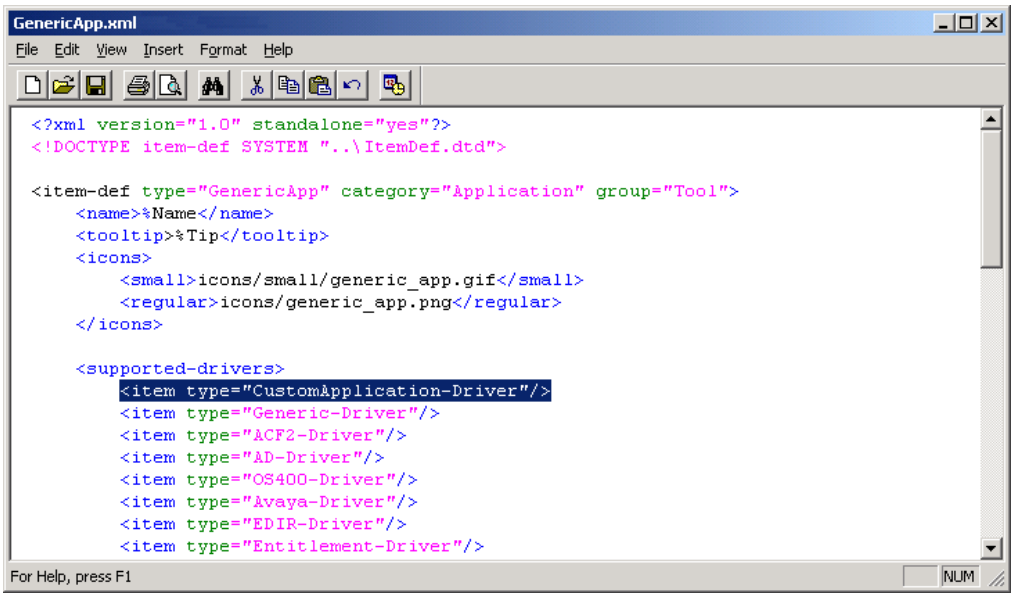
- The new Custom Driver configuration file
- All the Delimited Text driver configurations as specified in the application definition file



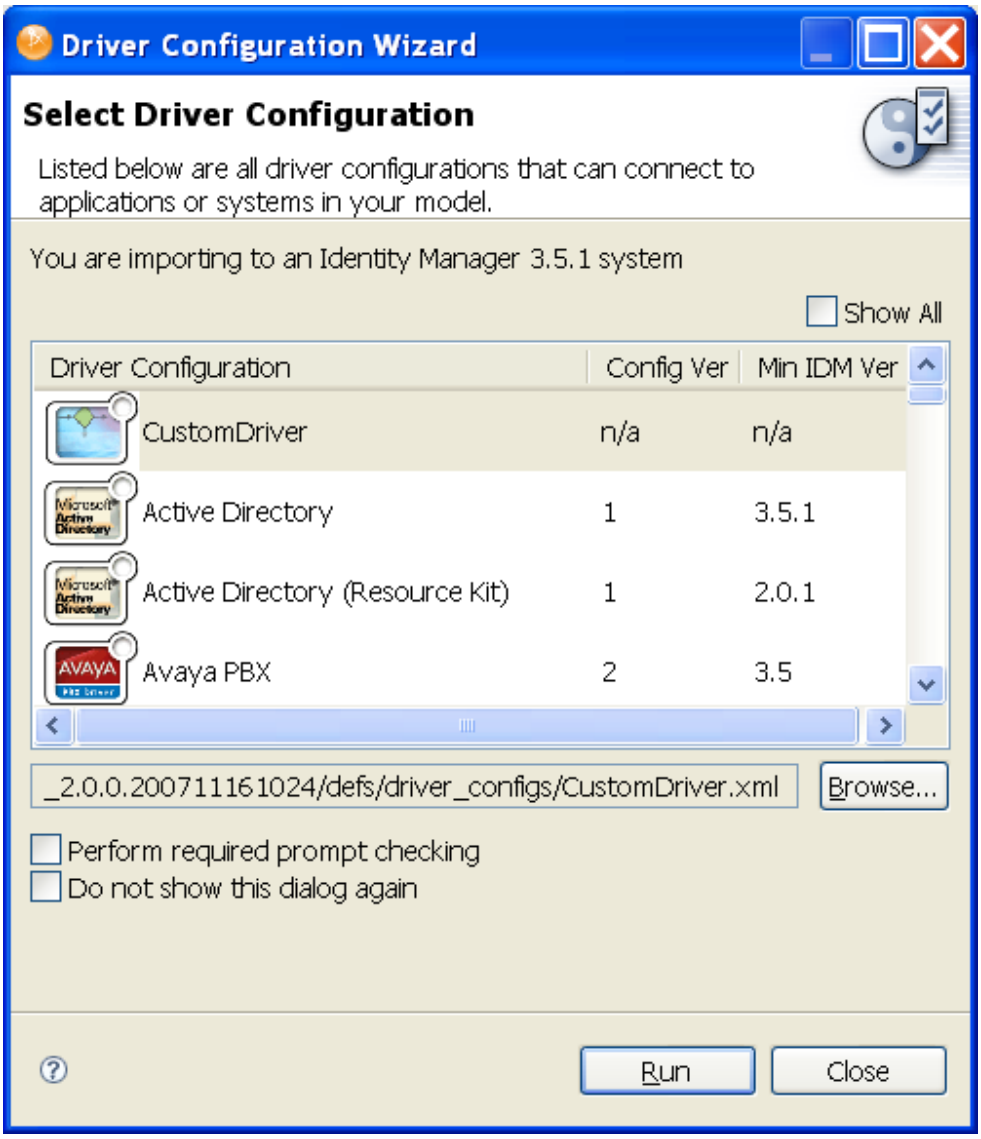
2 For full functionality in Designer, hook up your custom application to the Generic Application (GenericApp).

2a Open the application definition file `defs/model_items/Applications/Tool/GenericApp.xml`.

2b Add the new driver CustomApplication-Driver to the list of supported drivers.


GenericApp.xml
File Edit View Insert Format Help
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE item-def SYSTEM "..\ItemDef.dtd">
<item-def type="GenericApp" category="Application" group="Tool">
<name>%Name</name>
<tooltip>%Tip</tooltip>
<icons>
<small>icons/small/generic_app.gif</small>
<regular>icons/generic_app.png</regular>
</icons>
<supported-drivers>
<item type="CustomApplication-Driver"/>
<item type="Generic-Driver"/>
<item type="ACF2-Driver"/>
<item type="AD-Driver"/>
<item type="OS400-Driver"/>
<item type="Avaya-Driver"/>
<item type="EDIR-Driver"/>
<item type="Entitlement-Driver"/>
For Help, press F1
NUM

If you now drag and drop a Generic Application from the Tools group, your new custom driver appears as a selectable option in the Driver Configuration Wizard.



C.3 Protecting Your Customized Files

The files that you created are customized files. If you upgrade Designer, you lose part of the customization in these files. Therefore, before upgrading, you need to save these customized files into a protected directory. After the upgrade, copy or re-create the files.

Moving Data from Older Projects

D

Projects created in Designer 3.0 M2 are not compatible with Designer 4.0. Because there is no project converter from Milestone 2, you can use the procedures in this section to re-create Milestone 2 projects and ensure that data in projects moves over as smoothly as possible. These steps apply any time you want to manually import a project.

There are three methods for re-creating old projects:

- Section D.1, “Importing Data from a Live System,” on page 561
- Section D.2, “Exporting Data from the Old Project To Configuration Files,” on page 561
- Section D.3, “Manually Configuring Information That Is Not Imported,” on page 562

D.1 Importing Data from a Live System

If the Designer project is deployed to an Identity Vault, import a project from the Identity Vault by selecting the desired driver sets. This imports the project as it exists in the Identity Vault. This is the safest way to restore the project data; however, it does not restore passwords.

D.2 Exporting Data from the Old Project To Configuration Files

If you do not have the project deployed, you can export or import configuration files by using the following procedure:

First, perform the following steps in the non-compatible version of Designer (3.0 M2)

1. Edit the Designer preferences by selecting *Windows >Preferences*.
2. Select *Designer for IDM >Import/Deploy*.
3. Select the *Export* tab and verify that the check box for *Copy cross-driver policy references into exported configuration files* is not checked.
4. Export all libraries that are not contained by a driver set.
5. Export driver sets (if you are prompted to copy linked policies, select *no*).

After you finish the above procedure, do the following in Designer 4.0:

1. Create a new project.
2. Create Identity Vaults to match a previous project.
3. Provide Identity Vault credentials.
4. Create servers to match a previous project.
5. Delete the default driver set from the Identity Vault.
6. Import the library configurations into the appropriate Identity Vault.
7. Import the driver set configurations into the appropriate Identity Vault.

D.2.1 If Multiple Servers Are Associated with a Driver Set

During the export process, the Designer 3.0 M2 version prompts you to select which server to use. Repeat the export for each server, naming the export files something like `DriverSet-ServerX`. This process exports the server-specific information to the configuration file.

For example, if you have two servers named Stage1 and Stage2, then you would perform two exports and have two export files: `DriverSet-Stage1` and `DriverSet-Stage2`.

After you import the driver set configuration from the first server, perform the following steps:

1. Edit the properties of the driver set and add the second server to the driver set's server list.
2. Run the import again by right-clicking the Identity Vault and selecting *Import from Configuration File*.
3. Select the configuration file created for the other server.
4. You are asked which server to use for the import; select the appropriate server for the configuration being imported.
5. Finish the import.
6. Repeat Step 2 through Step 5 for each server configuration that you have exported from the Designer 3.0 M2 system.

D.2.2 Customized E-Mail Templates.

If you have customized e-mail templates, you need to export them from the Designer 3.0 M2 release and import them into Designer 4.0.

First, perform the following steps in the non-compatible version of Designer (3.0 M2):

1. Right-click the customized template and select *Save As.*.
2. Provide a name and location for the template.
3. Click *OK*.

After you finish the above procedure, do the following in Designer 4.0:

1. Right-click the Default Notification Collection in the Outline view and select *Import template from file*.
2. Select the exported template file.
3. Click *OK*.

D.3 Manually Configuring Information That Is Not Imported

After importing from the Identity Vault or from configuration files, there is information that is not migrated from the Designer 3.0 M2 project. This includes:

- Identity Vault credentials
- Server objects
- Design elements
- Administrator information

- Model layout
- Environment information
- Custom application icons
- Project properties
- Remote load documentation
- Any custom files added to the `Documents` or `Toolbox` folders
- VNC and LDAP connectivity information on applications

If you imported from configuration files, you were prompted for password information during the import process. If you imported from an Identity Vault, you need to manually enter the following information:

Driver passwords The driver shim password is prompted for during import.

Shim Auth (Application) passwords The shim authentication password is prompted for during import.

Remote Loader passwords The remote loader password is prompted for during import.

Named Password values The named password is prompted for during import.

You can copy and paste some informational data (text) from old projects to new projects:

1. Open the old project in the non-compatible version of Designer (3.0 M2).
2. Open the new project in Designer 4.0, then copy and paste informational data from the old project into the new project.

   For example, you can copy and paste administrator and environment information in this manner. However, you cannot copy and paste objects.

Version Control with Subversion and Identity Manager Designer

E

This appendix is intended for people using Identity Manager Designer and Subversion. Identity Manager Designer includes complete documentation covering how to use version control. This appendix gives more background on Subversion and indicates why you should make certain decisions. The Designer documentation tells you which protocols are supported. This paper tells you why you should choose one over the others.

There are many books available on administering a Subversion server and working with Subversion. I recommend *Version Control With Subversion*. It is available without charge at O'Reilly Media (http://svnbook.red-bean.com). This is an excellent reference book for Subversion. Many topics in the book are touched upon in this paper, and this paper references specific sections of the book.

- Section E.1, "Understanding Subversion," on page 565
- Section E.2, "Administering Your Subversion Server," on page 568
- Section E.3, "Taking Full Advantage of Version Control," on page 576
- Section E.4, "Glossary," on page 578

E.1 Understanding Subversion

Subversion is a version control system. Version control systems let you manage and create multiple revisions of your project and documents. They also allow you to share those revisions among a team of people.

- Section E.1.1, "How Revisions Work In Subversion," on page 565
- Section E.1.2, "Understanding Atomic Commits," on page 566
- Section E.1.3, "Where Subversion Stores the Project Data," on page 567
- Section E.1.4, "Moving an Existing Project," on page 567

E.1.1 How Revisions Work In Subversion

Revisions are a the heart of the Subversion functionality. A revision is a number that marks a specific set of changes made to a set of files. A single revision number can cover changes made to multiple files, but all of those files must be in the same repository.

Subversion uses a single revision number for the entire repository. This revision number is incremented every time any change is made to the Subversion server. For example, if you import a project at revision 100 and then create an Identity Vault and commit (revision 101), create a driver and commit (revision 102), and create a policy and commit, you are at revision 103. If you have multiple projects in the same repository, then every change made to any project increments the revision number for the whole server.

Although revision numbers are created for the entire server, different objects in your project can have different revision numbers. For example, suppose you start with revision 100 and create a policy and commit it; then create a mapping table resource and commit that version. The project will

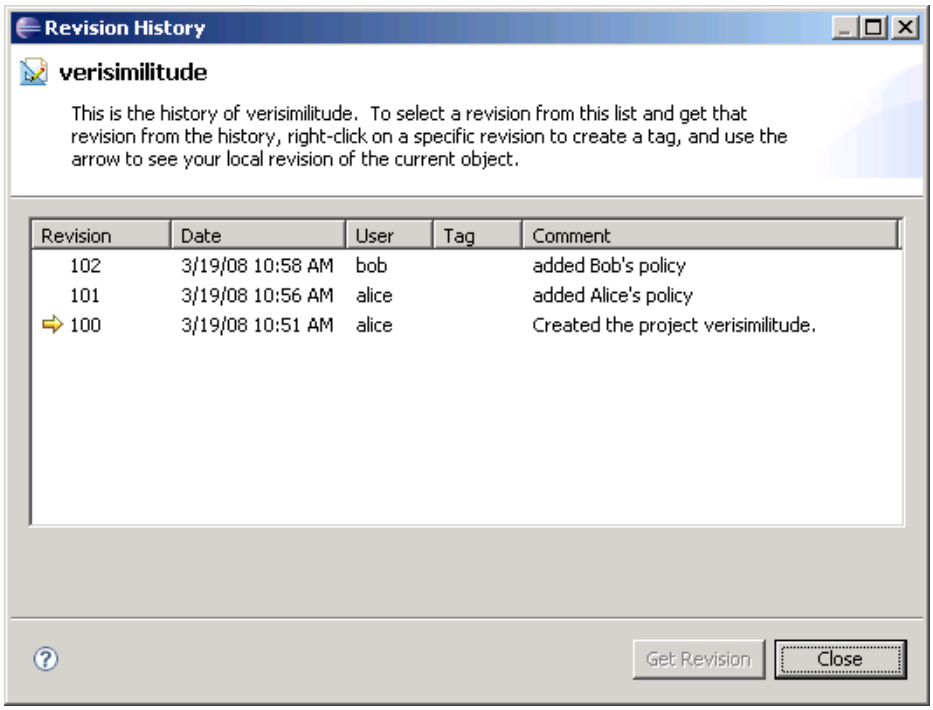
be at revision 100, the policy will be at revision 101, and the mapping table resource will be at revision 102. You can see the current revision of a specific object by using the Revision History or Properties page. The Revision History page indicates the specific object revision with a yellow arrow. In this example, the yellow arrow points to revision 100 for the project even though you see revision 101 and 102.

Subversion is meant to work in a team environment. In a team environment, there could be someone else editing the project at the same time as you. Let's look at an example:

- Alice imports a project at revision 100 to her local workspace.
- Bob imports the same project, also at revision 100, to his local workspace.
- Alice adds a new policy and commits, which creates revision 101.
- Bob adds a different new policy and commits, which creates revision 102

At this point Alice’s project is at revision 100, her policy is at revision 101, and the latest revision on the server is revision 102. If Alice wants to see Bob’s policy, she needs to update her project so she has revision 102.

Figure E-1 *Viewing Changes Through the Revision History*



Revisions are a useful way to track the versions of your project. Revisions can help you get projects back from the history and make sure that two users have the same version of a project loaded.

E.1.2 Understanding Atomic Commits

Atomic commits are a major feature of Subversion. The atomic commits treat the commit operation as a single event that either completely succeeds or fails gracefully. That means all of your changes are committed to the server or none of them are. Let's look at an example. Alice and Bob are working together on a project. Alice makes changes to multiple policies and entitlements that are all

interdependent. While Alice is in the process of committing this change, her network connection goes down. Before Alice can connect to the server again, Bob does an update. Subversion ensures that Bob does not get a partial update from Alice. Because Alice had a problem during her commit, Subversion makes sure that none of the files are changed on the server. Alice can then perform the commit after her network connection is restored.

Atomic commits are a very powerful tool and an excellent way to avoid broken projects. Atomic commits are always available within Identity Manager Designer and Subversion. You don't need to do anything special to enable them.

E.1.3 Where Subversion Stores the Project Data

When you commit a project to Subversion, the project is stored in the Subversion repository. The Subversion repository is an based on an internal database Subversion uses to store files. Subversion stores a separate file containing the specific changes made in each revision using the revision number as the filename.

These files are combined to maintain the concatenation of all of the changes made in your repository and the history of those changes. These files are iterative in nature and contain only the changes made for that specific revision. You can access these files in the `db/revs` directory of your Subversion repository.

Beyond that one requirement, there are no firm rules about setting up your projects. Here are some guidelines:

- It is a good practice to place a project in a directory of the same name. For example, the project called project1 would go into a folder such as `trunk/projects/project1`.
- Most repositories have a “sandbox” area. Users new to version control can experiment in this area without worrying about corrupting existing projects.
- It is a good practice to organize groups of projects. You can group projects by user, team, or company. The key is that having a large number of projects at the same level can be difficult to navigate.

E.1.4 Moving an Existing Project

Identity Manager Designer does not provide support for moving a committed project from one place on your Subversion server to another. However, you can do this with the Subversion command line:

- Make sure the whole team commits all of their local changes.
- Have all team members delete their local projects.
- Use the Subversion move command to move the project location.
- Have each team member import the project from the new location.

The Subversion move command is very simple. You just specify the current location of the project and the new location you want to move it to. For example, if your project is located at `trunk/project1` and you want to move it to `trunk/myprojects/project1`, use the following command:

```
 svn mv -m "<your comment for the move>" http://myserver/trunk/project1 http://myserver/trunk/myprojects/project1
```

Subversion moves the project to the new location and maintains all of the files and history.

E.2 Administering Your Subversion Server

Larger companies will most likely have a Subversion server administrator. Smaller companies might require you to install the Subversion server yourself. You can also choose to install Subversion on your own machine for easy backups. Either way, it is a good idea to know how the server should be configured and administered.

- Section E.2.1, “Server Specifications,” on page 568
- Section E.2.2, “Network Protocols,” on page 569
- Section E.2.3, “Authentication Schemes,” on page 571
- Section E.2.4, “Using Client Certificates,” on page 573
- Section E.2.5, “Configuring Subversion with Apache HTTP,” on page 574
- Section E.2.6, “Proxy Server Configuration,” on page 574
- Section E.2.7, “Subversion Server Backup,” on page 575

E.2.1 Server Specifications

The platform where you run Identity Manager Designer and the platform where you run the Subversion server are completely independent. Identity Manager Designer includes a Subversion client and is supported on any platform where Identity Manager Designer is supported.

Subversion provides official builds for the following platforms:

- Red Hat Linux
- Debian GNU/Linux
- FreeBSD
- OpenBSD
- NetBSD
- Solaris
- Mac OS X
- Windows NT, 2000, XP, and 2003
- HP-UX
- AIX
- IBM i5/OS (OS/400)

Subversion also works very well on SUSE Linux, but an official SUSE build is not available from the official Subversion site. Although Novell strongly encourages you to run on SUSE Linux, the Subversion server works well on all of the platforms. The platform you choose might depend on the IT organization you are working with, existing infrastructure, or just personal preference.

Subversion is a lightweight product and doesn't require a very powerful machine. The specific requirements depend on many factors, such as the number of users, the number of projects, and the other software running on that system. There is a discussion thread with some specific recommendations you can find at Subversion Mailing List (http://subversion.tigris.org/servlets/BrowseList?list=users&by=thread&from=330941).

E.2.2 Network Protocols

Subversion supports direct file access, and the SVN, HTTP, HTTPS, and SVN+SSH network protocols. These protocols define how Designer communicates with the Subversion server. The server must be configured to support a set of specific protocols. You specify the protocol you are using in the first part of the URL you use to connect to your version control server.

The protocol that you are using is transparent while you use Designer. Everything works basically the same whichever protocol you use. However, the choice of protocol has significant impact on the network traffic, security, and speed of your interactions with Subversion. Choosing the protocol is an important decision.

Direct File Access

Direct file access is not actually a network protocol. You can simply point Designer at a repository on your hard driver and access it directly. This is the easiest option to set up because it doesn't even require the Subversion server to be running. The version control import dialog has an option to browse for your local repository location. This is a good option for single users, experimenting with version control, and giving demonstrations.

The main drawback of direct file access is that it doesn't support network access for multiple users. Direct file access is not a network protocol; your repository cannot be accessed by other people. As a result, it does not provide good support for authentication schemes. This makes direct file access a poor choice for team environments.

You specify this protocol by connecting to your server with a URL that looks like this:

```
C:\subversion\myrepository
```

or

```
/home/<my username>/subversion/myrepository
```

SVN

SVN is a Subversion-specific protocol. This is the protocol that is used when you run the Subversion server without the Apache HTTP Server. Just follow the Subversion server setup instructions in the Identity Manager Designer documentation and you will be using this protocol. The SVN protocol supports networking and works well with small teams. It supports password file authentication as well as path-based authentication.

The SVN protocol does not support any type of encryption. This means that all information sent between Identity Manager Designer and the Subversion server is in clear text and could potentially be seen by a third party. Another concern with the SVN protocol is accessibility through firewalls. SVN is a specialized protocol and most firewalls will need specific configuration to support it. Many firewall administrators are wary of changing their configuration.

You should check with all companies involved before choosing this option. If you do need to configure a firewall to allow the SVN protocol, you must allow connections on TCP port 3690. In addition, the SVN protocol is not supported by most proxy servers.

The SVN protocol is a good choice for small teams where everyone works together in the same company. It is fast and easy to configure. You specify this protocol by connecting to your server with a URL that looks like this:

```
svn://mysubversionserver/myrepository
```

or

```
svn://localhost
```

HTTP

Subversion supports the use of HTTP by using a protocol called WebDAV. This protocol allows Designer to access Subversion by using the same protocols that Web browsers use to access the Internet. The Subversion server also requires the Apache HTTP server to support the HTTP protocol. This requires a little more server configuration, but it isn't too difficult. Using the Apache HTTP server also allows many more authentication options.

The main advantage of HTTP is that it works with existing firewalls and proxy servers. This makes HTTP a good choice when working with multiple companies, or working inside corporate networks. HTTP does not support encryption between the Subversion server and Identity Manager Designer. If you need to protect your data, then you should choose a different protocol. You specify this protocol by connecting to your server with a URL that looks like this:

```
http://subversion.mycompany.com/myrepository
```

HTTPS

HTTPS works very similarly to HTTP, with the addition of data encryption between the Subversion server and Identity Manager Designer. HTTPS uses the SSL (Secure Socket Layer) encryption protocol to make sure that third parties cannot read the communications between Identity Manager Designer and the Subversion server. HTTPS is slightly slower, but in practice the difference is negligible. HTTPS is a good choice for corporate environments concerned about securing their data. HTTPS is the protocol Novell developers use when working on the Identity Manager Designer source code.

The main drawback to HTTPS is that it can be difficult to configure. SSL requires a signing certificate that is granted by a certificate authority like Verisign.com. These certificates must be purchased, and applying for and installing them can be time-consuming. However, most server administrators are familiar with this process and should be able to guide you through it. You specify this protocol by connecting to your server with a URL that looks like this:

```
https://subversion.mycompany.com/myrepository
```

SVN+SSH

SSH (Secure Shell Protocol) is most popular on UNIX. Windows does not support SSH without additional software, and the configuration can be very difficult. SSH security is based on public key encryption using X.509 certificates. SSH is a good choice for UNIX environments looking for additional security. SSH requires a change to firewalls because it is not allowed on most corporate configurations. SSH uses TCP and UDP over port 22. You specify this protocol by connecting to your server with and URL that looks like this:

```
svn+ssh://subversion.mycompany.com/myrepository
```

Protocol Comparison

Table E-1 *Protocol Comparison*

Protocol	Pros	Cons	Port
Direct File Access	Really easy to set up, great for single developers.	Doesn't support team environments.	None
SVN	Easy setup and good network support.	Doesn't support encryption, doesn't support complex authentication, and has trouble with firewalls.	TCP 3690
HTTP	Good network support, works well with firewalls, and supports complex authentication.	Requires the Apache HTTP server and is not a good choice for running the Subversion server locally.	TCP 80
HTTPS	Good network support with good security options. A good choice for larger corporations.	Requires the Apache HTTP server, a certificate from a third party, and more complicated server configuration.	TCP 443
SVN+SSH	Good security in UNIX environments.	Doesn't support Windows well and can be difficult to configure.	TCP/UDP 22

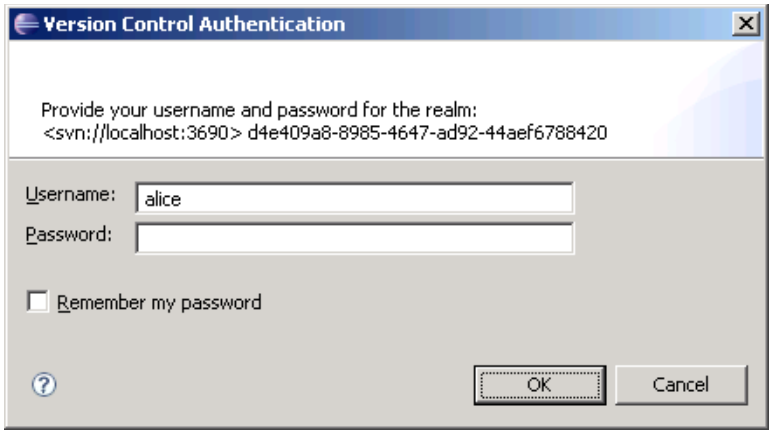
E.2.3 Authentication Schemes

In addition to deciding which protocols to use, it is important to look at authentication schemes. An authentication scheme defines the way users identify themselves to your Subversion server. This has significant impact on security as well as user management. Authentication schemes can be just a list of user names and passwords in a flat file, or a multiple-server environment requiring special certificates for each client.

Specifying a Realm

Subversion makes use of realms in order to simplify user management. A realm is a string that identifies how your server authenticates its users. This string does not need to be unique to your server. Specifying the same realm in multiple servers indicates that the same username and password can be used in any server using that realm. The realm your server is using shows up when a user is prompted for authentication information in Identity Manager Designer.

Figure E-2 *Providing Authentication for the Realm*



By default, Subversion generates a unique ID for your realm, such as:

```
de409a8-8985-4647-ad92-44aef6788420
```

You can change the realm for your server in the svnserver.conf file located in your repository's conf directory. If you are using Subversion in conjunction with the Apache HTTP server, you will need to use the Apache HTTP server configuration to specify your realm. More information about configuring this information can be found at the Appache Core Features page (http://httpd.apache.org/docs/2.2/mod/core.html#authname).

User Management

Whether you are just a single user or part of a large team, you need to manage the users who are allowed to access your Subversion server.

Flat Password File

The easiest way to manage user access is with a password file. This file specifies a list of users and their passwords. The file looks like this:

```
[users]
alice = alicepassword
bob = bobpassword
carol = carolpassword
dave = davepassword
```

This option is easy to configure and works well for small teams where security is not a major concern. However, in environments with larger teams, the management of this file quickly becomes unfeasible. In addition, this system is only as secure as the computer it is running on. If someone gains access to your Subversion server, they have access to this passwords file and every user's password.

Apache HTTP Authentication

If you configure Subversion to run with the Apache HTTP server, you can take advantage of the Apache HTTP server authentication. This mechanism also works with a flat file, but is much more flexible than the Subversion mechanism. This mechanism can manage users and groups, deny access by IP address, and much more. You can find information about this feature at Apache's Authentication for Apache HTTP Server page (http://httpd.apache.org/docs/2.0/howto/auth.html).

Apache HTTP Authentication with Third-Party Modules

Apache includes a large variety of third-party authentication modules. These modules support authentication to Windows NT domain controllers, UNIX password systems, Novell eDirectory, and many more. Novell uses a module to authenticate against an eDirectory server for its internal Subversion servers. As of this writing, there are 76 Apache HTTP modules dealing with authentication.

Creating a more complex authentication scheme might seem like a daunting task, but it can pay off in the long run. A good authentication mechanism can be mostly self-sustaining and gives users the opportunity to manage their own accounts. Combining advanced authentication with SSL or SSH provides ample security for a Subversion environment.

Specifying Project-Level Access

There are times when specifying access on a per-server basis is not sufficient. In those cases, you can use project-level access controls. There is support for this in Subversion as well as in the Apache HTTP server. When configuring this option by using the Subversion server, you can create an authorization DB file. The following sample file grants Alice the rights to read and write everything, Bob the write to read everything, Carol the right to read and write *project 1* while only reading *project 2*, and Dave only the rights to read and write *project 2*.

```
[/]
alice = rw
bob = r

[/Project 1]
carol = rw

[/Project 2]
carol = r
dave = rw
```

You must specify the location of this file by using the `authz-db` value in the `svnserve.conf` file in your Subversion repository conf directory. For more information about configuring this option with the Apache HTTP server, consult the documentation for the `mod_auth` and `mod_access` packages.

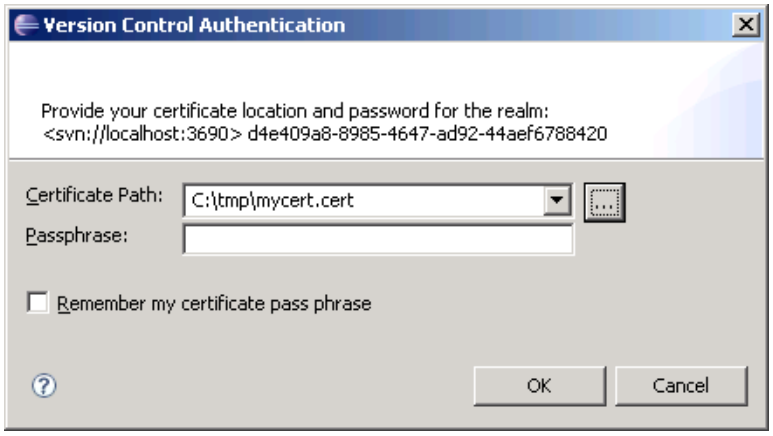
E.2.4 Using Client Certificates

Most security schemes in Subversion use a username and password to provide authentication. This is security based on something your know (your password). If you are especially concerned about security, you can use SSL client certificates. This is based on something you know (your password) and something you have (the certificate).

You can use client certificates with Identity Manager Designer and Subversion, but you must use the Apache HTTP Server. You will need to configure the Apache HTTP server to accept the client certificates. Apache can be configured to use client certificates by using the mod_ldap package. More information about that package can be found at the Apache Module mod_ldap page (http://httpd.apache.org/docs/2.2/mod/mod_ldap.html).

If your Subversion server is configured to use client certificates, you are will be prompted to provide a certificate in Identity Manager Designer. If you already have a Web browser configured to provide the client certificate, you can export the certificate for use with Identity Manager Designer. Tell your browser to export the client certificate and specify the `PKCS#12` format. You can then browse and select this certificate when you are prompted by Identity Manager Designer.

Figure E-3 *Authenticating to Version Control*



E.2.5 Configuring Subversion with Apache HTTP

The Subversion server is a set of libraries. These libraries are accessible with the custom SVN protocol by using the `svnserve` program. They are also accessible with the HTTP and HTTPS protocols by using the `mod_dav_svn` module for Apache HTTP server. This is a module that knows how to use the Apache HTTP server to support Subversion functions by using the WebDAV protocol. You can find information about installing and configuring `mod_dav_svn` at Configuring mod_dav_svn (http://svnbook.red-bean.com/en/1.1/ch09s06.html).

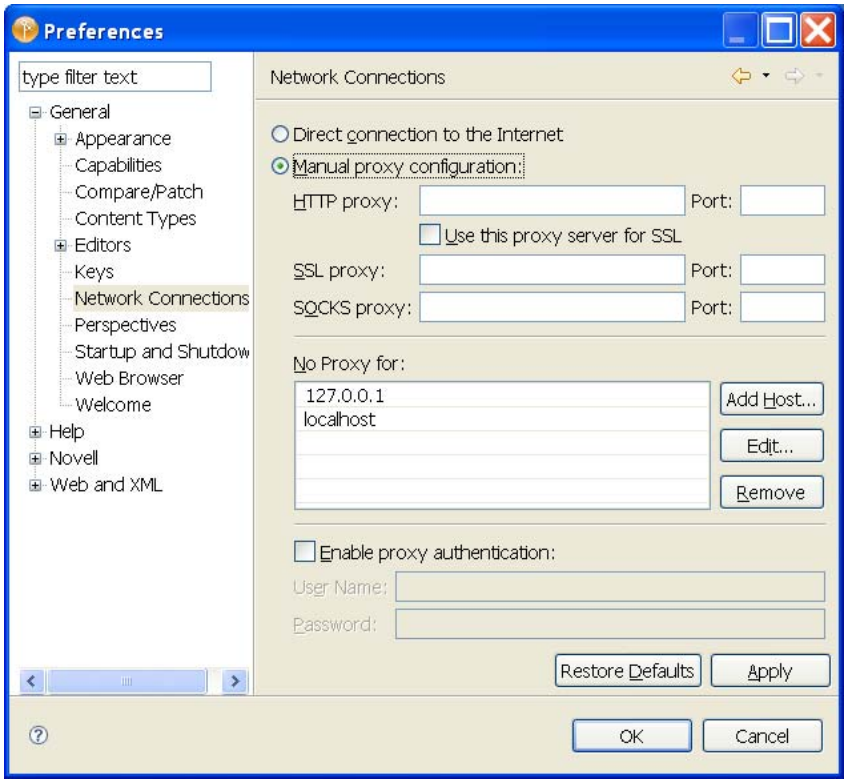
The standalone Subversion server is lightweight, easy to configure, and very stable. However, the Subversion server does not support HTTP, HTTPS, and advanced user authentication as well as other key features. The Subversion server is also not meant for large projects with many users. If you need any of the more advanced features, or if you need to support a large user base, you should use the Apache HTTP server. Both the Apache HTTP server and the Subversion server are free software.

E.2.6 Proxy Server Configuration

A proxy server is an application that takes requests and sends them on to other servers. Proxy servers are often used by companies to monitor and filter access to the Internet. Many large companies require all Internet access to be routed through the proxy server. If you are trying to access a Subversion server that is outside of such a network, then you must configure the proxy settings in Identity Manager Designer.

In the main Designer menu, go to Window and then select *Preferences*. In the Preferences page, select *General > Network Connections*. This preference page allows you to configure the proxy server settings for Identity Manager Designer. Select the *Manual proxy configuration* option and supply the proxy settings specified by your network administrator.

Figure E-4 *Setting Proxy Server Settings*



Most proxy servers support only the HTTP and HTTPS protocols. Some proxy servers support the SVN+SSH protocol and almost none support the SVN protocol.

NOTE: If you use a proxy server, errors can occur occasionally when the proxy server fails to forward a packet. When errors occur, retry the operation. If you continue to have problems, verify that the proxy server is working correctly.

E.2.7 Subversion Server Backup

When you are using version control, the Subversion server acts as a backup mechanism for all your project data. It is vital that you back up the Subversion server frequently. If you do not back up the Subversion repository and your server has a hardware failure, you will lose your project data. Daily backups are essential for active servers.

Subversion provides two tools to help create backups without interruptions of services: dump and hotcopy. The dump command takes your entire repository and sends the contents to standard out. You can also specify revisions to start and stop at. The hotcopy command creates a copy of your Subversion repository, including the database and all other configuration information. You use the output from either of these commands to restore your Subversion repository during disaster recovery.

For more information on the dump and hotcopy commands, including examples, see Dump Command (http://svnbook.red-bean.com/en/1.1/re31.html) and Hotcopy Command (http://svnbook.red-bean.com/en/1.0/re33.html) respectively.

E.3 Taking Full Advantage of Version Control

Using version control to simply commit, update, and share projects can be very useful, but if that is all you do you are missing out on some very exciting functionality. Version control can change the way you work. It can enable a truly team-oriented development methodology.

- Section E.3.1, “When to Commit and When to Update,” on page 576
- Section E.3.2, “Comments,” on page 576
- Section E.3.3, “Creating and Using Tags,” on page 577
- Section E.3.4, “Subversion Keyword Substitutions,” on page 577

E.3.1 When to Commit and When to Update

Version control is a tool for sharing and backing up your project. You should take full advantage of it. That means committing often and updating frequently. You should learn to be comfortable with committing. The project doesn’t need to be perfect, just make sure you won't impede your teammates.

You should also update frequently to get your teammate's changes. This ensures that you are working with an up-to-date project, and your changes can work with the changes your teammates are making. You also resolve conflicts in a better way. The earlier you can resolve a conflict, the easier it is be to resolve that conflict.

For example, if two individuals are editing the same policy and they work separately for a week, the two versions of the policy will be very different. This makes it very likely that there are conflicts and very likely that those conflicts are difficult to resolve. If those two users update frequently, they would avoid most of the conflicts and make them much easier to resolve.

E.3.2 Comments

Whenever you commit a change to the version control server, you are prompted for a comment. Comments are your chance to describe the change for yourself and for your teammates. Comments can explain why you did something and what you were thinking when you did it.

Good comments should take the form of sentences. They should describe what you did and why you did it. A good comment should give you a good idea of what has changed, but it does not need to describe every change in detail.

Good Comments

- Created a new project for work on the new Active Directory drivers for Unilateral Widgets Incorporated.
- Added a new AD driver to connect to the second directory and moved policy1 to a library so we can access it from the new driver.
- Changed the second rule in policy1 to avoid the potential for an infinite loop when handling more than three users.

Bad Comments

Comments should not be too brief:

- Added new policy
- New project
- Undid Joe’s change

Comments should also not be too specific:

- Changed the condition of policy add password rule operation-data to be the following:
  ```
  <and>
  <if-operation op="equal">add</if-operation>
  <if-password op="available"/>
  <if-xpath op="not-true">operation-data</if-xpath>
  </and>
  ```

E.3.3 Creating and Using Tags

A tag is a readable name given to a specific revision. For example, you could tag revision 100 as *Release 1.0*. Tagging is most useful for identifying significant revisions. If you certify that you are ready to send a project to a customer, that is probably a good time to create a tag. You can then access that tag later if you need to roll back a change. The combination of tagging and the Get from History feature gives you a powerful tool to manage releases and deployments.

E.3.4 Subversion Keyword Substitutions

You can use Subversion keyword substitution to give you more information on selected objects. For example, you can use the Description area to track the revision number, the date and time an object was last submitted to Subversion, and who submitted the last revision in the description of an object. The following example uses a policy.

1. Select a policy and bring it up in an editor.
2. In the Policy Description area, add the following keywords:
   - $Date$
   - Rev
   - $Author$
3. Save these changes to the policy.
4. Check the project into the version control server.
5. Update the project from the version control server.
6. Open the policy in an editor and you will see that Subversion has substituted the keywords to show the following:
   - $Date: 2008-05-20 11:17:51 -0500 (Tue, 20 May 2008) $
   - $Rev: 1513 $
   - $Author: tpew $

You need to perform this procedure for each object for which you want to see this type of information. You can add keywords in any place where you can add text, but the object's Description area is most accessible. The keywords are updated as you or other team members make changes to the object.

E.4 Glossary

Appache HTTP Server. The server that is used in conjunction with the Subversion server to run the HTTP and HTTPS protocols. The Apache HTTP Server is free and open source. More information, including installation instructions, is available at the Apache HTTP Server Project Homepage (http://httpd.apache.org).

Authentication Scheme. Controls how users are authenticated with your Subversion server. Can be as simple as a flat file or can support many different authentication servers.

Commit. The process of taking your local changes and sending them to the Subversion server. Also called "checking in."

Import. You import a project from version control to get a project that has already been created and put it on your local machine. After the initial import, you can use the update process to download subsequent changes.

Repository. The place where the Subversion server holds your files. Repositories can support many projects, and many repositories can be supported on the same computer.

Revision. A number identifying a specific set of changes to files in your project and across the Subversion repository. Revisions are visible in the version control history and properties dialogs.

Server Realm. A string that identifies the authentication scheme to be used for that server. Servers with the same realms can use the same set of credentials to support a variation of single sign-on.

Update. Getting the latest changes from the server and applying them to the work area on your local machine.